



University of
Zurich^{UZH}

*Burkhard Stiller, Muriel Franco, Christian Killer,
Sina Rafati, Bruno Rodrigues, Eder John Scheid (Edts).*

Communication Systems XII

TECHNICAL REPORT – No. IFI-2019.05

August 2019

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland



Introduction

The Department of Informatics (IFI) of the University of Zurich, Switzerland works on research and teaching in the area of computer networks and communication systems. Communication systems include a wide range of topics and drive many research and development activities. Therefore, during the spring term FS 2019 a new instance of the Communication Systems seminar has been prepared and students as well as supervisors worked on this topic.

The areas of communication systems include among others wired and wireless network technologies, various network protocols, network management, Quality-of-Service (QoS) provisioning, mobility, security aspects, peer-to-peer systems, multimedia communication, and manifold applications, determining important parts of future networks. Therefore, this year's seminar addressed such areas in more depth. The understanding and clear identification of problems in technical and organizational terms have been prepared and challenges as well as weaknesses of existing approaches have been addressed. All talks in this seminar provide a systematic approach to judge dedicated pieces of systems or proposals and their suitability.

Content

This new edition of the seminar entitled “Communication Systems XII” discusses a number of selected topics in the area of computer networks and communication systems.

The first talk on “An Overview of Swiss Cyber Infrastructures from a Security Perspective” evaluates the existing cyber security infrastructures of Switzerland.

Talk 2 “Approaches and Challenges in Blockchain Scalability” evaluates the current scalability methods and their differences in the blockchain realm.

Talk 3 “An Analytical Study of ERP Systems in the Supply Chain Industry” focuses on the integration of heterogeneous solutions for supply chain tracking such as blockchains, IoT and ERP systems.

Talk 4 “On the Legal Validity of Blockchain-based Smart Contracts” explains the legal aspects of developing Smart Contract based systems.

Talk 5 “Techniques and Strategies to Stimulate Cooperation in Competitive Environments” introduces the challenges of cooperation in competitive environments and the advantages of collaboration in such cases.

Talk 6 “Network Functions Virtualization (NFV) in Smart Cities” introduces some of the newest technologies and products used in the fifth generation of wireless communications.

Talk 7 “Investigating the Blockchain Technology in the Context of Cybersecurity” evaluates the proposed blockchain-based cyber security systems and their potential advantages and disadvantages with respect to the cyber security aspects.

Talk 8 “The Hyperledger Fabric” explains the IBM Hyperledger Fabric components and use cases.

Talk 9 “An Overview of Information Visualization for Data Exploring in Blockchain Universe” specifies the possibilities of leveraging visualization of blockchain based systems.

At the end, Talk 10 “Evaluation and Comparison of Blockchain Consensus Algorithms” focuses on the differences of the blockchains and the role of consensus mechanisms on their overall performance.

Seminar Operation

Based on well-developed experiences of former seminars, held in different academic environments, all interested students worked on an initially offered set of papers and book chapters. Those relate to the topic titles as presented in the Table of Content below. They prepared a written essay as a clearly focused presentation, an evaluation, and a summary of those topics. Each of these essays is included in this technical report as a separate section and allows for an overview on important areas of concern, technology architectures and functionality, sometimes business models in operation, and problems encountered.

In addition, every group of students prepared a slide presentation of approximately 45 minutes to present its findings and summaries to the audience of students attending the seminar and other interested students, research assistants, and professors. Following a general question and answer phase, a student-lead discussion debated open issues and critical statements with the audience.

Local IFI support for preparing talks, reports, and their preparation by students had been granted by Muriel Franco, Christian Killer, Sina Rafati, Bruno Rodrigues, Eder John Scheid, and Burkhard Stiller. In particular, many thanks are addressed to Sina Rafati and Bruno Rodrigues organizing the seminar and for their strong commitment on getting this technical report ready and quickly published. A larger number of pre-presentation discussions have provided valuable insights in the emerging and moving field of communication systems, both for all groups of students and supervisors. Many thanks to all people contributing to the success of this event, which has happened in a lively group of highly motivated and technically qualified students and people.

Zurich, August 2019

Contents

1	An Overview of Swiss Cyber Infrastructure from a Security Perspective	7
	<i>Louis Bienz, Getoar Gallopeni, Matej Jakovljevic, Marc Zwimpher</i>	
2	Approaches and Challenges in Blockchain Scalability	39
	<i>Mesut Ceylan, Catharina Dekker, Luka Popovic, Nathalie Torrent</i>	
3	An Analytical Study of Enterprise Resource Planning (ERP) Systems in the Supply Chain Industry	69
	<i>Hülya Hüsler, Moritz Wittwer, Ramon Huber</i>	
4	On the Legal Validity of Blockchain-based Smart Contracts	99
	<i>Fabian Kueffer, Kilian Werder, Pascal Kiechl and Lukas Mueller</i>	
5	Cooperation in Competitive Environments	127
	<i>Han-Mi Nguyen, Dominik Buenzli</i>	
6	Network Functions Virtualization (NFV) in Smart Cities	147
	<i>Lawand Muhamad, Can Inan, Atif Ghulam Nabi</i>	
7	Investigating the Blockchain Technology in the Context of Cybersecurity	173
	<i>Lenz Baumann, Roland Schlaefli, Silas Weber, Pascal Zehnder</i>	
8	The Hyperledger Fabric	205
	<i>Karim Abou el Naga, Danjuel Dordevic, Claude Muller, Lucas Thorbecke</i>	
9	An Overview of Information Visualization for Data Exploring in Blockchain Universe	239
	<i>Basil Fuchs, Jeremy Kubrak, Tim Grimm, Severin Wullschleger</i>	
10	Evaluation and Comparison of Blockchain Consensus Algorithms	277
	<i>Joel Barmettler, Özgür Acar Güler, Marc Laville, Spasen Trendafilov</i>	

Chapter 1

An Overview of Swiss Cyber Infrastructure from a Security Perspective

Louis Bienz, Getoar Gallopeni, Matej Jakovljevic, Marc Zwimpfer

In the last years, more and more systems and infrastructures have become digitized. As a result, those formerly offline systems are now more vulnerable to cyber threats. Thus, improving and upgrading the cyber security should be one of the main goals of the institutions which own or develop those system. Critical National Infrastructures (CNI) are infrastructures that are crucial for the everyday life of the population of the nation. Infrastructures from different sectors can be considered as CNI, some directly affect the health of the population others are "only" affecting the national economy. These infrastructures should receive special attention regarding reliability and security because of their importance and the dangerous impacts if they would fail. Considering current threats and Benchmarking standards regarding cyperspace as well as the National Strategy for the protection of Switzerland against cyber risks, this report gives and overview about the current Situation of Switzerland.

Contents

1.1	Critical National Infrastructures	9
1.1.1	Definition of Critical National Infrastructures	9
1.1.2	Overview Swiss CNI	9
1.1.3	Types of Security for CNI	10
1.1.4	Security problems and Challenges	11
1.1.5	Statistics on Cyber threats and financial impact	13
1.1.6	Discussion	14
1.2	Current-Threats	15
1.2.1	Overview	15
1.2.2	Malware	15
1.2.3	Phishing	17
1.2.4	(Distributed) Denial-of-Service	18
1.2.5	Man in the Middle	18
1.2.6	SQL Injection	18
1.2.7	Discussion	19
1.3	Example-Attacks	19
1.3.1	Attack on Ukrainian Power Grid	19
1.3.2	ProtonMail	20
1.3.3	Stuxnet	20
1.3.4	WannaCry	21
1.3.5	Discussion	21
1.4	Benchmarking standards: NIST, ISO	21
1.4.1	Best-practice standards: An overview	21
1.4.2	ISO 27000:2013	22
1.4.3	NIST Cybersecurity Framework	23
1.4.4	ISO vs. NIST	25
1.4.5	Discussion	26
1.5	Cybersecurity in Switzerland: Insights	26
1.5.1	Laws and regulations	26
1.5.2	Committee "ICT Switzerland"	27
1.5.3	Discussion	28
1.6	National Strategy for the protection of Switzerland against cyber risks (NCS)	28
1.6.1	Overview	28
1.6.2	Achieved Objectives	29
1.6.3	Need for Action	30
1.6.4	Spheres of Action	33
1.6.5	Discussion	35

1.1 Critical National Infrastructures

Nowadays, many systems and infrastructures are at least to some extent digitized and are no longer fully analogous as before. Contemporaneously the number and the variations of cyber-attacks is growing [8]. Hence, currently many infrastructures are potentially vulnerable to cyber-attacks. Being successfully attacked is always bad for a system per se, however the failing of certain systems or infrastructures could have an enormous influence on the every day life of the population; these infrastructures are stated as Critical National Infrastructures (CNI).

1.1.1 Definition of Critical National Infrastructures

Critical National Infrastructures are according to the Swiss government processes, systems and infrastructures which are crucial for a functioning economy or the welfare of the population of a nation [4]. Thus, these infrastructures have to function in order that the normal daily life in a nation is not interrupted in any way. The UK government even expand the infrastructures which are considered as CNI by also including the assets which are needed for full functionality of the CNI and all the people which operate and facilitate these infrastructures [7]. Hence in summary, CNI is a general term that includes all systems, assets, networks, facilities and people which are needed to keep the every day life of the population and the economy running normally.

Because of the importance of these infrastructures, it is necessary that they function all the time. Thus, their security should be one of the main concerns of every government, which is indeed the case considering that almost every government released strategies to secure these infrastructures. However, even though all systems have an effect on the daily life, the impact of the failures of some infrastructures is more immediate than others. Consider two CNI, for example a nuclear power plant and the garbage collection; both are important for the daily life. A life without electricity is not imaginable nowadays, but also the absence of a garbage collection would become very annoying (and possibly harmful) over time. Although both infrastructures are therefore critical for the population, the effects of a failure of a nuclear power plant is more immediately harmful to the population either directly for example the explosion or indirectly by a sudden power loss. Hence, critical national infrastructures can be categorized into two categories [7]:

- Infrastructures of which the failure has an immediate impact on the welfare and the security of the population of the nation
- Infrastructures of which the failure has no immediate effect on the welfare of the population or only affects the economy of the nation

Consequently, the critical national infrastructures which belong to the first category should receive more attention in regard to security and reliability due to the worse consequence if they were to fail.

1.1.2 Overview Swiss CNI

Considering the complexity of the actions which are needed for a functional nation, many systems have to interoperate perfectly. As a result of this, many system are critical for the nation. The Swiss government lists nine sectors which are treated as critical national infrastructure [4]:

As shown in table 1.1, Switzerland has many infrastructures which are considered critical for a functioning economy and nation. In this table there is no differentiation between the two categories mentioned in the previous chapter. Nevertheless, all these infrastructures should not be vulnerable to cyberattacks or any other type of attacks. The Swiss

Sectors	Subsectors
Government	Research and Education Cultural assets Judiciary, Governance, Parliament, Administration
Energy	Natural gas supply Mineral oil supply Power supply
Disposal	Waste Sewerage
Finance	Finance services Insurance services
Health	Pharmaceutical industry Hospitals Medical care
ICT	IT services Media Postal services Telecommunication
Nutrition	Food supply Water supply
Security	Military Civil defence Emergency services
Traffic	Air traffic Road traffic Rail traffic Shipping traffic

Table 1.1: Swiss Critical National Infrastructure according to Swiss government [4]

government states that all parts (IT-systems, companies, people, facilities etc.) which are used in one of the sectors or subsectors in table 1.1 are considered as Critical National Infrastructure [4]. They are considered as CNI regardless of their individual criticality. Criticality is a relative measure for the impact a failure of a Critical National Infrastructure has on the nation or population [4]. The criticality is omitted because it is dependent on the level at which it is being looked at. For example if a local sewage treatment plant breaks down, it has a big impact on the local community, but neither has it an influence on the whole nation nor affects it the welfare of the population directly. On the other hand, if a retaining dam collapses most likely many people are affected and even endangered by the following flood. Nonetheless, both infrastructures are considered as Swiss Critical National Infrastructures since a failure of any one of them would have an impact on some part of the population.

1.1.3 Types of Security for CNI

As discussed in the previous chapter, Critical National Infrastructures are essential for a nation's economy and the welfare of the population. As a result, it becomes crucial to a government to ensure that the Critical National Infrastructures function without any problem. Besides the aspect of reliability, the security against any sort of threat is perhaps the most important requirement to any Critical National Infrastructure. For a long time,

the biggest threats to any Critical National Infrastructures were environmental or posed by other humans.

A well-known disaster regarding a nuclear power plant was the meltdown of the nuclear power plant in Fukushima. In 2011, the Fukushima Daiichi Nuclear Power Plant in Japan was firstly hit by a Tsunami and later by an earth quake [13]. This combination led to the meltdown of three nuclear fusion reactors and to the release of radio-active material in the surroundings. Thus, this failure resulted from natural hazards. Furthermore, in order to create protection against natural hazards, Critical National Infrastructures are secured against threats caused by humans too. This is achieved by restricting access for example. Both of these threats will remain in the future, but in the closer past, security against a new kind of threat has gained importance. As a result of the digitization of most system, the risk of cyber attacks has increased in the last years [8]. Hence, Critical National Infrastructures must now be secured against physical as well as cyber threats. The following chapters will focus on this type of threat regarding types of threats, standards for securing systems and the Swiss strategy on how to deal those threats.

1.1.4 Security problems and Challenges

In the past, Critical National Infrastructures were operated by many staff members manually. However, over the last two decades these system became more and more digitized, resulting in the automation of some tasks and possible reduction of workers. The first digitized system were mostly offline and unconnected to public communication infrastructures [22]. This isolation saved them from many cyber threats because if no communication to the outside happens, no cyber attack can occur in this way. However, this isolation of the systems has decreased over time due to the many benefits an outside communication can bring [22]. The connection to other services could provide data which can be helpful for operating the system or allow remote control over certain parts of the system. Over the last years this trend is continuing due to new technologies, as for example Internet of things, which increase the interconnectivity even further. However, besides of all the advantages this interconnectivity brings to nowadays systems, the risk to be vulnerable to cyber attacks rises. From every new external interface originates a new possible way to attack the system thus security must increase accordingly.

Many Critical National Infrastructures are large systems with subcomponents for different tasks. Supervisory Control and Data Acquisition (SCADA) systems are Industrial Control Systems (ICS) that are widely in use in Critical National Infrastructures and they have been subject to an increasing number of cyber attacks [22]. These system are used to operate a waist number of CNI and thus must be specifically excellent in terms of reliability and security, because there failure would lead to the failure of the infrastructure itself.

SCADA system are used to collect the data of big infrastructures, for example power station, and after the analysis of this data, operate the infrastructures autonomously to some extend. In a nuclear power station this could include temperature monitoring and for example in case of overheating, shutting down the reactor. A cyber attack which alters this process could result in lethal consequences and thus this must be prevented.

Many SCADA system that are in use, evolved over time and formerly the reliability was of bigger concern than the security, but with the increasing interconnectivity the cyber risks have increased and cyber security should be the main subjective of the companies operating these system [22]. However, companies focus more on the economical aspects of these systems and since cyber security is costly, companies were rather conservative with investments in theses parts of the system. Consequently, problems regarding the cyber security were brought to some extend from the past.

Maglaras et al. [22] determine several reasons why cyber security problems persist nowadays. The reason for many of these problems is the financial decisions made by the

company operating these system. Many SCADA systems are used a lot longer than their intended lifespan since the introduction of new systems is costly. Those software and hardware are more error prone than newer system that are based on more modern technology. Older systems which were used offline in the beginning and only afterwards interconnected with other systems lack sometimes of encryption of the data and communication. When interconnected, this becomes a serious threat which was not the case before. Both of these problem could be solved by investing into systems with newer technology.

Additionally, companies which provide Critical National Infrastructure and use some sort of SCADA system to control this infrastructure, prefer to use sometimes off-the-shelf solutions rather than custom hardware and software. Normally, off-the-shelf solutions are cheaper than systems which are build from scratch for a certain infrastructure. On the other hand, off-the-shelf solutions may be well-documented and widely known, undermining obscurity which could simplify possibly cyber attacks [22].

Even when neglecting the legacy problems of SCADA system, there are other challenges that hinder keeping high security standards. Many system in use in Critical National Infrastructures have to be running all the time and hence it is not possible to interrupt them [22]. In order to secure a system the best way possible, it should be updated or patched regularly to be up-to-do against all possible threats. However, keeping in mind that SCADA system operate non-stop, upgrading the system can be quite difficult and perhaps costly, which could lead to the decision not upgrade the system. This is can be observed in many SCADA system used in Critical National Infrastructures [25]. As a result, the system could be vulnerable to threats which could have been avoided by updating it regularly. Furthermore, as the proper functioning of these systems is crucial, the reliability must be high. However, when imposing new security measures, they could have a disturbing influence on the reliability of the system. Due to this reliability frequently takes precedence over security [22].

	2015	2016	Change
Lack of skilled resources	36%	53%	+17pp
Budget constraints	56%	59%	+3pp
Lack of executive awareness or support	38%	37%	-1pp
Management and governance issues	33%	31%	-2pp
Lack of quality tools for managing information security	15%	24%	+9pp
Fragmentation of compliance/regulation	28%	24%	-4pp

Figure 1.1: Causes for cyber vulnerabilities [5]

Ernst & Young [5] released a report regarding the reason for vulnerabilities to cyber risks which summarizes the before mentioned problems and challenges. Figure 1.1 shows that almost 60% of the companies included in the survey mention that budget constraints are the reasons behind missing measures against cyber attacks. 53% of the companies stated that the lack of skilled resources, namely employees, pose danger to the cyber security of the company. This could problem could also be caused by the lack of willingness to invest into better cyber security by the company since the acquisition of skilled workers expensive in general. Ernst & Young further[5] mentions that tools and methods to recognise vulnerabilities and cyber threats are many times not advanced enough to meet the need in the current environment. All these factors together form a major peril to

the cyber security of companies and their system and may be responsible for considerable number of successful cyber attacks.

1.1.5 Statistics on Cyber threats and financial impact

As the digitization and interconnectivity of many system used in Critical National Infrastructure continues, the number of cyber threats increases. Even though many companies neglect the possibility to be a victim of a cyber attack, cyber attacks belong nowadays to the daily agenda of any company [5]. This may only be a simple and obvious spam mail or a big-sized direct cyber attack on the systems of a company. Nevertheless, many executive boards in charge have understated the threats and the necessity to take active measures against this kind of threats. Accordingly the steps they took in order to secure their systems against cyber threats were to small. However, perhaps due to all the media attention in recent times, the importance companies attach to cyber risks have risen. Figure 1.2 displays that 59% of the Swiss companies which have participated in this survey ruled that cyber risks were among the Top-5 risks they have to deal with. Although such a high percentage of companies is aware of the risks, the steps taken by these companies in order to manage the risks vary widely and may not suffice [8].



Figure 1.2: Importance of risk of cyber threat in Swiss companies [8]

Was your company a victim of a cyber attack during the last 12 months?

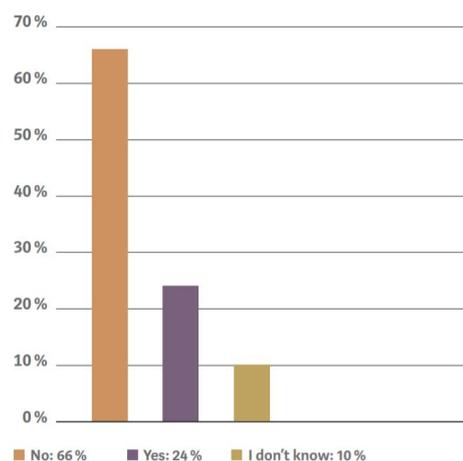


Figure 1.3: Statistic Swiss companies victims to cyber attacks [8]

Statistics on cyber attacks and threats to companies reveal the real extent of the posed dangers. Kessler & Co. Inc. [8] released a report on the actual number of cyber attacks and their impacts on Swiss companies. Figure 1.3 shows that around a quarter of all companies know that they were victim of some sort of cyber attack over the last year. One of the major problems regarding cyber security is apparent in these statistics, namely that it is very difficult to determine whether one even was victim to a cyber attack. As many as 10% of the companies interviewed for this report do not know whether they have been attacked or not. Malware often is very difficult to detected and sometimes only discovered after several weeks or month [8]. Therefore, it is likely that also some companies of the 65% group which state that they were not victim of a cyber attack in the last year, did not detect it yet. Hence, the number of actual attacks may be higher due to the possibly big number of undetected, and thus unreported, cyber attacks. As already discussed, the failure of system used in Critical National Infrastructures may have an influence on the welfare of the population. Besides this threat also the financial impacts due to cyber attacks are considerable.

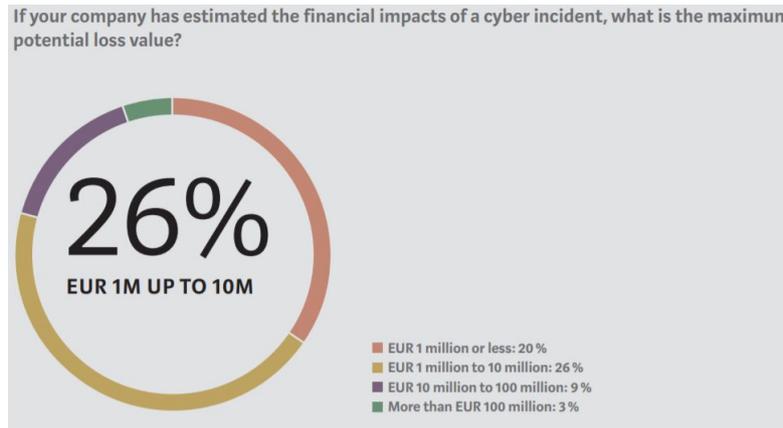


Figure 1.4: Financial Impact of cyber attack [8]

Figure 1.4 shows that more than 38% percent of the participating Swiss companies determine that they would lose more than 1 million Euro in case of a cyber attack. 3% of the companies would even suffer a loss of more than 100 million Euros. This poses the question why the investments into cyber security are in comparison still very small. Further interesting is that only slightly more than half of the involved companies estimated the potential impact of a cyber security even though they granted high risk to cyber threats [8]. In conclusion, there should be change of mind in the executive board of companies to at least further investigate the potential impacts of cyber threats and measures to prevent them.

1.1.6 Discussion

In conclusion, organizations that are in charge for any type Critical National Infrastructure facing new and difficult problems due to the digitization of their system. Whereas in the past, threats for Critical National Infrastructure were mostly of physical nature, for example natural hazards or threats introduced by humans, nowadays cyber attacks pose a non-neglectable danger to digital systems. As the failure of Critical National Infrastructure could have severe impacts on a variety of stakeholders, these infrastructures should be secured against this type of threat even more extensively than regular systems.

Furthermore the possibilities of being a victim of a cyber threats has risen over the last years and should be considered a significant risk [8]. Nevertheless, important infrastructures are still lacking an adequate security against cyber attacks. The reason for these deficits are partially originating from executive boards of companies which operate these infrastructures which are more focused on the current financial situation rather than investing in the secure infrastructure in the future. Due to this several legacy problem are present in the system which support the operation of Critical National Infrastructures, which make them potentially more vulnerable to cyber attacks [22]. Considering all the potential impacts of a failure of Critical National Infrastructures, whether it be economical or regarding the welfare of the population, the awareness of cyber risks and consequently the investment in security against them, must be raised in the future to be prepared for and successfully defend cyber attacks on these infrastructures. Without taking these measures, the threats coming from cyber attacks will rise and be even more dangerous in the future.

1.2 Current-Threats

1.2.1 Overview

When talking about cyber security it's important to know which threats exist and how they affect your infrastructure. In this section we will show the current cyber threats and explain how they can harm you and your infrastructure.

1.2.2 Malware

Malware is the abbreviation of 'malicious software'. As the name already says it is a collective of software which harms your infrastructure. There are different subtypes of malware. The infection, the kind how a malware damages your infrastructures and the propagation depends on the subtype. The maleficence of the subtypes also differs. The following subsections will explain some of the different malware types.

1.2.2.1 Computer Virus

The computer virus is the most known type of malware, the media and also the majority of endusers erroneously call every malware a computer virus. Fortunately that's not true. Computer viruses are code fragments or programs which are hidden in frequently used programs and infect your computer without your acknowledgement.[30]

The computer virus is the only type of malware, which infects other legitimate host files. Every time the infected file is executed the virus is executed to. When a virus is executed it can copy the viruses code and attack other files and make changes on your computer. The fact that viruses are executed when infected files are executed makes it very difficult to remove a them. Today's antivirus programs usually just delete the infected file or move it into quarantine. Fortunately less than 10 percent of the used malware are pure computer viruses.[30]

1.2.2.2 Computer Worm

The term computer worm has a wide span. Although there are worms which have to be activated by an user action, different than a computer virus, the typical computer worm doesn't need any user action to activate himself. Therefore it is able to spread fast through a network, what makes it very popular. Also worms do not need to attach itself to a file to damage your computer.

Different worms can differ in many ways. Weaver, Paxon, Staniford and Cunningham created a taxonomy in their work in order to classify the different computer worms. The created taxonomy is based on the following factors: target, discovery, carrier, activation, payloads and attackers[32]. The work also shows, how diverse computer worms can be.

Worms first gained widespread notice in 1988[3]. Although they differ in many ways, they often infect computers through vulnerabilities, or through email attachments. An example for a computer worm which used vulnerabilities to infect computers is the SQL slammer. The SQL slammer (or also called Sapphire) is the fastest computer worm in history, it exploited a buffer-overflow vulnerability in computers which were connected to the Microsoft SQL server. Even though a patch, which removed the vulnerability was released 6 before the worm appeared, many systems didn't apply it yet, so the worm was able to exploit the vulnerability and affect 90 percent of the connected computers, which didn't apply the patch, within 10 minutes. The worm slowed down computers all over the world[24].

Once a computer is infected with a computer worm, the worm can delete and/or modify files, which can be exploit by an attacker to take control over a computer. A worm

also steals system resources by making copies of itself, which affects the hard drive space and/or the bandwidth, by overloading a network.

To protect against computer worms, one should always keep the operating system and ones applications up to date in order to remove as much vulnerabilities as possible. Another point is to be careful by opening suspicious emails and visiting malicious websites, to recognize the latter a strong internet-security software could be helpful. Probably the biggest threat of computer worms is that they are ultimately written by humans and sooner or later something completely new will appear for what our cyber security won't be prepared. So often the easiest way to defend against computer worms is to remove the motivation for the attackers to write a computer worm[32].

1.2.2.3 Trojan Horse

Trojan horses or short trojans is a type of malware, which, as the name already says, masquerades itself as a legitimate software. The victim is usually tricked by a form of social engineering to download and execute a trojan. Trojans need a user action to be activated just like viruses do[18]. After the activation the attacker can perform actions on the victims computer. Which actions can be performed by the attacker depends on the trojans type. There exist many types of trojans, the following listing shows some types and the respectiv action:

Backdoor: This type of trojan enables the attacker to take over the control of the infected computer. Backdoors are often used to take control over a group of computers and build a botnet with them[18].

Rootkit: Rootkits prevent you to detect malicious activities or objects on your machine by concealing them[18].

Exploit: Exploits are programs which take advantage of vulnerabilities of applications or programs that run on your machine[18].

Trojan-DDoS: This type of trojans uses the infected system to perform a denial-of-Service attack, by sending traffic to a target. Denial-of-Service attacks will be described in a latter section of this report[18].

Trojan-Ransom: The trojan-ransom enables the attacker to modify the data on the infected machine[18].

This examples show that attackers can damage a computer in many different ways by using trojans, what makes them very popular under hackers.

The protection against trojans is very difficult since they trick the user by using social engineering, so it is hard for anti-malware programs to detect trojans early.

1.2.2.4 Hybrid Forms

Today attackers usually use combinations of the traditional malware types. This hybrid types usually use parts of trojans and parts of worms. Often a hybrid form appears to the victim as a trojan, thus masquerades itself as a legitimate program, but after the execution (activation) it acts like a worm, replicates itself and spreads fast over the network. This combinations of malware plays an important role for Distributed-Denial-of-Service (DDoS) attacks, which will be explained in a latter section.

Hybrid forms often use an exploit which hides it activities from the anti-malware programs on the infected computer, what makes them really hard to defend against[30].

1.2.2.5 Ransomware

Ransomware is a malicious software that encrypts data on the victims computer, what makes it unreadable/unexecutable for the victim. The attackers usually require a payment of a certain amount in form of cryptocurrency to decrypt the data again.

Ransomware became popular in the last past years and the popularity is still growing. Most ransomware programs are trojans, what means that they spread through a form of social engineering, what makes them hard to defend against. Hence the best way to be save from ransomware is to have an offline, separated, up to date backup, so that, in case of a ransomware infection, critical files can be retrieved from this backup[30].

1.2.2.6 Fileless Malware

Fileless malware isn't really a type of malware, we included it in this report because it is a growing threat. Where traditional malware usually infects a computer by using its files or file system, fileless malware doesn't directly use files or file systems. They exploit and spread in memory only or use other non-file objects in the operating system.

Fileless malware often exploits existing legitimate programs or built in tools to perform malicious instructions. the result of fileless malware is, that fileless malware and fileless attacks are harder to detect, what again makes it more attractive for cyber-criminals[30].

1.2.3 Phishing

This section is about phishing attacks. Traditionally cyber-security focuses more on encryption and authentication, but when talking about phishing attacks, the human factor has a high impact on the security. Cyber-criminals, that use phishing attacks try to obtain sensitive information, like for example passwords or bank information from a victim by using social engineering.

Usually phishing attacks are performed on many people, so that they are usually easy to rumble. As the public awareness of cyber threats increases, phishing attacks react by becoming more sophisticated[17]. Also the existence of do-it-yourself phishing kits (<http://www.sophos.com/spaminfo/articles/diyphishing.html>), which enables people without any technical knowledge to perform phishing attacks, makes the threat even bigger.

As mentioned before, phishing attempts are often performed automatically on many people. This kind of attacks is much less effective than perform phishing attacks on less people but therefor with more adapted content to the individual victim.

When talking about phishing, spoofing play an important role. Cyber criminals try to trick victims by crafting emails, faking IP packages, links and websites[21]. Often attackers send crafted emails with a sense of urgency to victims, this emails link the victim to a fake website, where the user inputs his login credentials, which can be read by the attacker.

If such phishing attacks succeed, attacker can obtain sensitive data from the victim. If the victim is a private person, cyber criminals may for example obtain sensitive bank information, if the victim is a company, the attacker may obtain important information about customers, company internals or even access to the companies network.

In order to protect against phishing attacks, it's very important to increase the public awareness of cyber threats. Since attackers use forms of social engineering, it is also very important to be careful by visiting suspect websites and reading suspect emails. Only enter in login credentials on HTTPS-protected sites. Using two-step verifications on your accounts also increases the security and makes it more difficult for attackers to successfully execute phishing attacks[11].

1.2.4 (Distributed) Denial-of-Service

In this subsection we will give an overview of Denial-of-Service (DoS) and Distributed-Denial-of-Service (DDoS) attacks. Denial-of-Service is a special kind of cyber attacks. When attackers perform Denial-of-Service attacks, they try to make the target unavailable for everybody, by sending as much traffic as possible to it.

Distributed-Denial-of-Service describes Denial-of-Service attacks from many computers simultaneously. The DDoS attack succeeds in two steps. In a first step the cyber criminal has to take over control over multiple computers. This is usually done by using trojan horses or other malware. These computers build a network which is also called botnet. In a second step, the attack on the target is performed by using the botnet to send as much traffic as possible to the chosen target. Distributed-Denial-of-Service makes it very difficult to identify the attacker, since the attacker often don't send any traffic from his own computer.

Denial-of-Service attack mostly happen on the third and fourth layer, i.e. on the infrastructure, but DoS attacks on the sixth and seventh layer, i.e. on the application layer also exist. Attacks on the infrastructure usually contain much more traffic than those on the application layer, but they are straight forward, so that they can be recognized easily. Attacks on the application layer are less enormous, but they are more sophisticated and concentrate on specific parts of the application, which wont be available for users anymore.

In order to protect against DoS and DDoS attacks, many vendors provide a DDoS defender software, which filters the incoming data and reacts if something suspicious is noticed[28].

1.2.5 Man in the Middle

Man-in-the-Middle (MITM) attacks are cyber attacks that usually don't use any kind of malware. The attacker positions himself between the victim and the resource the victim tries to access. By positioning himself between victim and resource, the attacker is able to read the transmitted data. The attack can be either active or passive, active if the transmitted data is modified or deleted and passive if the data is only read[19].

The most common variant of Man-in-the-Middle attacks uses a router to intercept the communication. To create such a router, cyber criminals can configure their computers to act like a WLAN hotspot and intercept every communication which is done in this network. Another way to perform such MITM attacks is to exploit vulnerabilities in legitimate routers to intercept the communications in a certain network. This method can be very effective since usually the MITM attack can endure over long time, so that the attacker can collect much data about the victims[19].

Another rather new type of Man-in-the-Middle attacks is the so called Man-in-the-Browser attack. Cyber criminals install malicious software on computers, which are connected to the browser. The malware saves the whole data exchange between website and victim[19]. Avoiding public WLAN routers is essential for the protection against MITM attacks. A strong WLAN encryption (WEP, WPA, WPA2 etc.) improves the protection on the server/router side[2].

1.2.6 SQL Injection

SQL injection (SQLI) describes the process where an attacker exploits vulnerabilities in websites. To perform a SQL injection attack, cyber criminals inputs SQL command which are then send from the application to its database. By performing SQL injections attackers may get unauthorized access to data, read it, modify it or also delete data from the

database.

The threat gets bigger since nowadays programs exist, which look for websites with vulnerabilities and automatically try to perform SQL injection attacks, where in the past attackers had to find their victims and perform the attacks by hand.

To protect a website from SQL injections, One should try to filter malicious inputs during the development. There also exist web application firewalls (WAF), which strengthen the protection against SQL injections[15].

1.2.7 Discussion

This section showed different cyber threats which are ubiquitous in today's world where the digitalization grows continuously. It is not enough to protect ones computer only against one of this threats, since many attacks use more then just one of these components. There are many things to consider in order to protect ones computer, The following listing shows some important measures:

- One should make sure that the computer is up to date to avoid the exploitation of vulnerabilities, which are probably removed with the next patch.
- Use reliable anti-virus programs
- Be aware of the current cyber threats and keep in mind that cyber criminals make use of social engineering to trick victims
-> Be careful when getting suspicious emails and visiting fraudulent websites, and avoid using public WLANs.
- If one runs web applications and/or provides services, one should think about a web application firewall and also about a DDoS defending software.

By following the points of this list, one can protect ones computer against known threats. Unfortunately a total protection is not possible, since cyber criminals continuously find new, previously unknown ways to perform cyber attacks and protection software can only protect against known threats.

1.3 Example-Attacks

In this section we will go through some example cyber attacks from the past, which also affected critical infrastructures. Fortunately we didn't find any examples where Switzerland's critical infrastructure was affected, but we put one small example in this section, where a cyber attack on a company with Swiss datacenters was performed.

1.3.1 Attack on Ukrainian Power Grid

One of the largest successful cyber attacks in the recent past on Critical National Infrastructures was the attack on the Ukrainian Power Grid on the 23th of December in 2015. The origin of the cyber attack is still not fully clarified since no official statement or claim for this attack was released by anyone [9]. However, it is strongly assumed that the Russian government is at least to some extent responsible for the attack due to various reasons [9]:

- The attack happened during the war and the Russian military intervention in Ukraine
- The IP-addresses of the source of the attack were located in Russia

- The complexity and size of the attack suggest that some sort of bigger "Hacker" group must be responsible with considerable financial and technical resources

The attack itself consists of several phases which all together resulted in almost 225,000 people without power for one to six hours. Prior to the actual attack the [9]. Afterwards, they switched off substation of the powergrid over the ICS and additionally destroyed parts of the firmware of the ICS [9]. Lastly, they executed an denial-of-service attack on the call center to cut of the support for the customers. Even though no major harm was caused by the attack itself, it shows the potential danger of cyber threads for Critical National Infrastructures.

1.3.2 ProtonMail

ProtonMail is a free secure email service with built-in end-to-end encryption and other state of the art security features based in Switzerland. They are developing and widely distributing their tools in order to build an secure and privacy respecting internet. Because we didn't find any cyber attack on critical infrastructures in Switzerland we decided to include the cyber attack on ProtonMail in our report[1].

ProtonMail suffered a continuous DDoS attack, which started on November 4th, 2015. They received a blackmail email, in which the attackers summon ProtonMail to pay a ransom, shortly before midnight on November 3rd, which was followed by a DDoS attack. The attackers firstly flooded ProtonMails IP addresses, which expanded to to the data-centers in Switzerland, where ProtonMails servers are located. This first attack swamped ProtonMails system for 15 minutes and was performed by two separate groups in two phases. The first group, which also demanded a ransom was a volumetric attack, which targeted only the IP addresses. The second one, which was more complex and more sophisticated, targeted vulnerabilities in the infrastructure of ProtonMails internet service provider (ISP). This DDoS attack on ProtonMail was probably the largest cyber attack in Switzerland. It affected hundreds of other companies, which took collateral damage. The attack endured and ProtonMail tried to mitigate the cyber attack. With the help of multiple companies, especially IP-Max, which is a Swiss company, and Radware, a DDoS protection company, ProtonMail managed it to gain control over the situation[1].

1.3.3 Stuxnet

Stuxnet is probably the most known cyber attack in history. Stuxnet is a computer worm, which only targeted computers with an industrial control system (ICS), and modified their programmable logic controllers (PLC) so, that they work like the attackers want. The whole process was hidden by using a Windows rootkit, so that the operator of the equipment didn't notice anything. The attack was very complex, the attackers used a vast array of components to achieve their goal. They used zero-day exploits, a Windows rootkit as mentioned before, the first ever PLC rootkit, antivirus evasion techniques, complex process injection and hooking code, network infection routines, peer-to-peer updates, and a command and control interface[27].

The worm infected over 200'000 Windows machines and physically degraded about 1'000. The fact that it was programmed for a Siemens control technology (Simatic-S7), which addresses a particular industry in Iran, leads to the unproven assumption, that the worm was intended to sabotage the nuclear power plants in Iran[10]. Through the modification of the PLC the spinning uranium enrichment centrifuges got speed up and slow down, so that some of them got destroyed (The Vulnerability of Nuclear Facilities to Cyber Attack). By current estimations the worm decreased the enrichment efficiency by 30 percent[23].

Stuxnet showed how effective cyber attacks on critical infrastructures can be and how the future warfare could change.

1.3.4 WannaCry

WannaCry is a ransomware attack, which took place in May 2017. It infected more than 230'000 computers in over 150 countries[10]. Under the victims were also hospitals, universities, companies and government organizations[31]. WannaCry was a ransomware crypto-worm and targeted Windows machines. It consists of two parts. The first was an exploit, more precise EternalBlue, which is an exploit of Windows' Server Message Block (SMB) protocol. WannaCry propagated by using this exploit[10]. The second part was an encrypter, which encrypted the data on the victims machines. The extension '.WCRY' was added to the encrypted files, so that users couldn't read them anymore. After data encryption, a 'ransom note' was displayed, which informed the victims that their files are encrypted and demanding a payment in form of cryptocurrency (Bitcoin) (300US US dollar within 3 days, 600 US dollar within 7 days). The migration could be stopped by activating a 'kill-switch'[10].

As mentioned before WannaCry also affected hospitals, especially the united kingdom was affected. Some hospitals in the UK had to stop their activities completely and thousands of appointments and operations were cancelled[12].

Even if hospitals weren't the main target of the WannaCry attack, some of them got hardly affected, what shows that shows that critical infrastructures don't even have to be the main target of an cyber attack to get damaged.

1.3.5 Discussion

In this section we showed a few examples of previous cyber attacks. These examples show, that critical infrastructure are not excluded from cyber attacks. The Stuxnet and the attack on the Ukrainian power grid example show us how cyber attacks, which are specifically targeting critical infrastructures, can affect and damage them effectively. These two examples also show us that cyber attacks could further be used in future warfare.

The fact that the WannaCry ransomware attack affected multiple hospitals and stopped their operations, shows that critical infrastructures aren't even save from cyber attacks which don't specifically target them. The DDoS attack on ProtonMail made clear that also targets in Switzerland are not safe from cyber attacks. Although we didn't find any attacks on critical infrastructures in Switzerland it can't be ruled out, that cyber criminals unsuccessfully tried to attack them or that they will try to attack them in the future.

1.4 Benchmarking standards: NIST, ISO

1.4.1 Best-practice standards: An overview

While most of corporate business actions can be measured rather easily, it is hard to measure security. Contrary to the popular opinion, never having suffered from a cyber attack is not a valid indication for a stable security organization. There are several stakeholders which have considerable interest in the security of an organization. It could either be a costumer who wants to be sure that his information are secured properly or members of the board who do not want to end up in the newspaper for a data breach. Obviously, announcing that an organization implemented a 24/7 Security Operation Center is in fact an indicator for security, but the indicator is limited to professionals only. There is need

for standards to measure and reveal the security level such that all stakeholders are able to understand and develop a common view.

Such standards are for instance ISO27000, the Cybersecurity Framework of the National Institute of Standards and Security, the Control Objectives for Information and Related Technology (COBIT) of the Information Systems Audit and Control Association (ISACA) or IEC6244 of the International Electrotechnical Commission. Even though they work differently, they are following common goals. With help of the mentioned standards, security responsables have the ability to compare themselves to other market players or even to whole industries. Committing to be certified against ISO27000 or improving one's NIST tier level makes investments into security justifiable for the security responsible in an organization. What this means in detail, will be discussed in the following sections.

1.4.2 ISO 27000:2013

The ISO 27000-series is the series of standards which is published by the International Standards Organization (ISO) and is revised in a five years cycle. While most of the companies are certified against the version of 2013, noted as ISO27000:2013, the most recent publication is ISO27000:2018. Part of the series are the standards 27001 - 27021 and 27799, which all either specify requirements, describe general or even define sector-specific guidelines. In Figure 1.5 you can see the exact allocation of the standards to the topics.[16]

The standard foresees 14 Control Objectives, each covering a topic which should be taken up by every security organization in order to have comprehensive security measures in place. The control objectives are:

- Information Security Policies
- Organization of Information Security
- Human resource security
- Asset management
- Access control
- Cryptography
- Physical and environmental security
- Operations security
- Communications security
- System acquisition, development and maintenance
- Supplier relationships
- Information security incident management
- Information security aspects of business continuity management
- Compliance

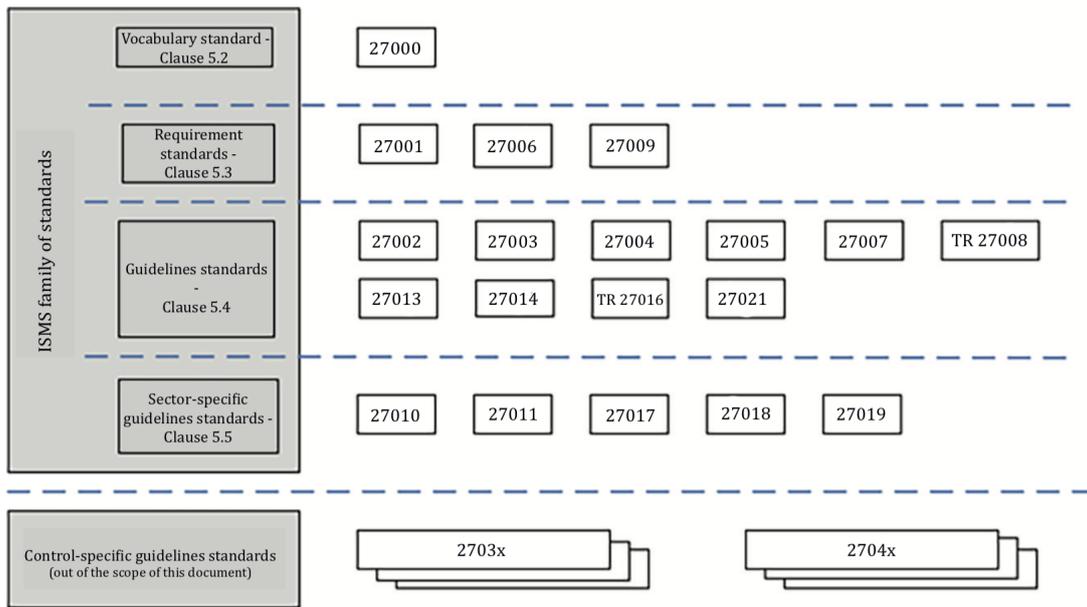


Figure 1.5: Overview ISO2700 series groups [16]

Each control objective comes with a certain amount of pre-defined controls, which can differ significantly between the objectives. While ISO foresees 15 controls for physical and environmental security, there are only two for information security policies. In order to receive the certification all controls have to be functionally in place, as specified in the control specification and best-practice guidelines in ISO27002.[16]

As all certifications of ISO, the Information Security Management certification is well-known and a clear signal for a certain security level. It is a respected measure which shows clear commitment to the security organization of a company. Although the International Standards Organization renews its publication all five years and therefore keeps up to current challenges which are relevant to the market, many dynamic components are not taken into consideration. Aspects as the fit of the corporate strategy to the security strategy and the adaption to the risk appetite are left behind, which equals the rating of a static snapshot rather than a future-oriented assessment.

1.4.3 NIST Cybersecurity Framework

The Cybersecurity Framework (CSF) of the National Institute for Standards and Technology, also known as the Framework for Improving Critical Infrastructure Cybersecurity (FICIC) is a voluntary framework that utilizes existing standards and "informative references" to create a repeatable and standardized process for measuring the implementation of a firm's cybersecurity program and controls. The framework serves as a basis for organizations to determine their current cybersecurity capabilities, to set goals for target future state and establish a plan for maintaining and improving their cybersecurity program. The framework is comprised of three primary components: framework profile, framework core and framework implementation tiers. The framework was not primarily intended to be a maturity assessment or for benchmarking purposes. Nor was it intended to be a one-size-fits-all methodology, due to its very varying rating content. The results of the implementation tiers for NIST are intended to provide organizations with insights into the current state of the controls they have placed into operation.[26]

Function	Category	ID
Identify	Asset Management	ID.AM
	Business Environment	ID.BE
	Governance	ID.GV
	Risk Assessment	ID.RA
	Risk Management Strategy	ID.RM
	Supply Chain Risk Management	ID.SC
Protect	Identity Management & Access Control	PR.AC
	Awareness and Training	PR.AT
	Data Security	PR.DS
	Information Protection Processes & Procedures	PR.IP
	Maintenance	PR.MA
	Protective Technology	PR.PT
Detect	Anomalies and Events	DE.AE
	Security Continuous Monitoring	DE.CM
	Detection Processes	DE.DP
Respond	Response Planning	RS.RP
	Communications	RS.CO
	Analysis	RS.AN
	Mitigation	RS.MI
	Improvements	RS.IM
Recover	Recovery Planning	RC.RP
	Improvements	RC.IM
	Communications	RC.CO

Figure 1.6: NIST Functions, Categories and IDs

The NIST Framework Profiles are used to describe the current state or the desired target state of specific cybersecurity activities. The current profile indicates the cybersecurity outcomes that are currently being achieved. The target profile indicates the outcomes needed to achieve the desired cybersecurity risk management goals.[26]

The Framework core, which demonstrates the methodology of the CSF, consists of five functions each containing several categories within. In Table 1.6 we can see which categories belong to which functions including the ID's of the categories. The lifecycle methodology of the NIST CSF begins with *identifying* the firm's most important information, taking necessary measures to *protect* them, establishing mechanisms to *detect* potential incidents, react appropriately and finally *recover* from it. While the function from identifying onwards to the detection are strategies to prepare for disruption, the following functions are strategies for responding effectively to it.[26]

Tier level descriptions	
1	<p>Partial</p> <ul style="list-style-type: none"> Organizational cybersecurity risk management practices are not formalized, and risk is managed in an ad-hoc and sometimes reactive manner. There is limited awareness of cybersecurity risk at the organizational level, and an organization-wide approach to managing cybersecurity risk has not been established. An organization may not have the processes in place to participate in coordination or collaboration with other entities.
2	<p>Risk informed</p> <ul style="list-style-type: none"> Risk management practices are approved by management but may not be established as organization-wide policy. There is an awareness of cybersecurity risk at the organizational level, but an organization-wide approach to managing cybersecurity risk has not been established. The organization knows its role in the larger ecosystem but has not formalized its capabilities to interact and share information externally.
3	<p>Repeatable</p> <ul style="list-style-type: none"> The organization's risk management practices are formally approved and expressed as policy. There is an organization-wide approach to manage cybersecurity risk. The organization understands its dependencies and partners, and receives information from these partners that enables collaboration and risk-based management decisions within the organization in response to events.
4	<p>Advanced</p> <ul style="list-style-type: none"> The organization adapts its cybersecurity practices based on lessons learned and predictive indicators derived from previous and current cybersecurity activities. There is an organization-wide approach to managing cybersecurity risk that uses risk-informed policies, processes and procedures to address potential cybersecurity events. The organization manages risk and actively shares information with partners so that accurate, current information is being distributed and consumed to improve cybersecurity before a cybersecurity event occurs.

Figure 1.7: NIST CSF Implementation Tier Levels

According to NIST, the implementation tier scale in Figure 1.7 is used to assess the current controls implemented within each function and category. Mapping the tier scale to the numbers 1-4, one gets an overall rating along the NIST lifecycle. Such a rating is mostly revealed in form of a spider, which also can be adapted to a companies needs as desired. In the Figures 1.8 and 1.9 we can see two exemplary mock-up spiders, on one side the overall rating of the lifecycle, on the other side the rating of the identifying function with its categories.

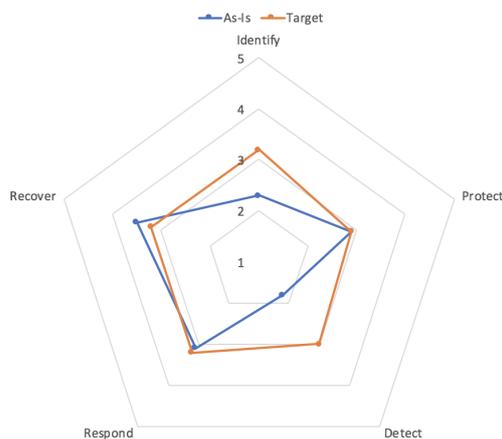


Figure 1.8: Exemplary Spider

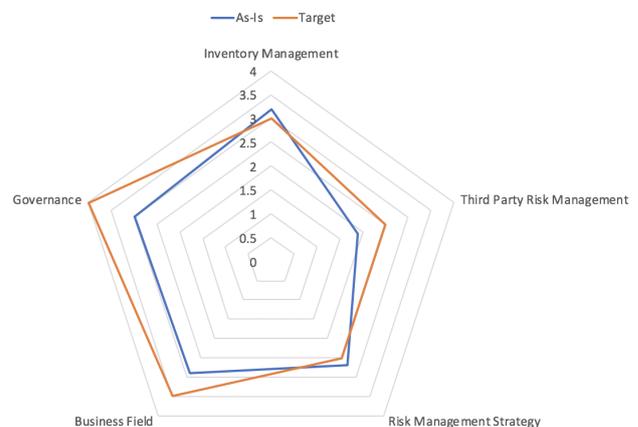


Figure 1.9: Exemplary Spider with some categories of the function "Identify"

1.4.4 ISO vs. NIST

Although the use of the two standards is designed very differently, there are some very relevant similarities. The most obvious ones are their technology neutrality which makes them applicable to every organization and that they both are risk management frameworks which both require some sort of safeguards to be implemented. Realistically, not only one of them but the optimal combination of both, the NIST CSF and ISO's 27000 series, build an efficient and effective system for information security management.[20]

For instance, they both are technology neutral, applicable to all organizations and provide a methodology to implement *information security* and *Cybersecurity* in an organization as such. They both are so called risk management frameworks which require safeguards to be implemented for the case that security risks are detected. And lastly, they both hold important references to other frameworks, including COBIT 5, SANS Critical Security Controls, ISA 62443 and NIST's Special Publication (SP) 800-53.[20]

On the other hand, there are some differences which need to be taken into consideration as well. For instance, while ISO27001 is a recognized and accepted standard which organizations can be certified against, NIST CSF is a common but still voluntary framework for measuring the implementation of cybersecurity controls. Furthermore, ISO considers a wider definition of information and guides generally for handling information in an organization while NIST focuses solely on information stored and processed by (critical) IT systems. Also, the implementation methodology differs a lot: On one side we have a functionally based approach with main focus on Identify, Protect, Detect, Respond and Recover, whereas working with ISO follows a Plan-Do-Check-Act (PDCA). Finally, we know that ISO defines requirements for organizations to build an information security management system, but NIST helps organizations, based on their risk and security profiles, structure areas of security to be implemented (current state versus desired state).[20]

1.4.5 Discussion

It turns out that each of the two discussed standards have their own strengths and weaknesses. Optimally, a purposefully rationing of both comes in place and ensures an ideal security set-up.

One main benefit of the ISO27000 series is clearly the fact that companies can be certified against it. Its international prominence makes it a very strong signal for internal as well as external purposes. And unlike the cybersecurity framework, it focuses on protecting information in a much wider sense. Although non-digitally stored information can be neglected in most companies, it is indispensable in terms of completeness of information security. For many companies these guidelines and controls still both make sense and are helpful. What brings additional simplicity is the detailed specification of the required documents and records need to be in place and what the minimum implementation is for the certification.

On the other hand, the structure of NIST's Cybersecurity Framework is unreachable. Five functions are divided into 22 categories which all together contain 98 subcategories covering the entire IT landscape. This is in fact similar to the controls defined in ISO27000 Annex A, with the remarkable difference that each subcategory contains multiple references to the known benchmarking standards like ISO27000, COBIT, ISA62443 and NIST's own Special Publication (SP) 800-53. Having those included, the requirement elicitation effort for the implementing firm is distinctly lower. Furthermore, the tier concept supports the people responsible when it comes questions of implementation depth. Which degrees of implementation depth are to be considered and which target makes sense to aim at can be answered easier with the tier system of NIST CSF. Also, as part of the Third-Party Risk Management for example, the Framework is very convenient to set minimum standards other organizations as suppliers or other partners.

1.5 Cybersecurity in Switzerland: Insights

1.5.1 Laws and regulations

In May 25th 2018, the General Data Protection Regulation (GDPR) came into force. This regulation, adopted by the European Parliament and issued in first instance for banks and

other financial institutions, was a first step into taking cyber risks seriously and facing up to the topic in general. Although it was a small start into the right direction, impact on the market was huge as many companies, inside and outside the European Union, were affected. For the first time, politicians had to deal with the "ilities" of the security principles. What might be overdue already, is now reality for all firms operating within or offering services to the European Union, many of them in Switzerland too. [29]

GDPR focuses on data privacy and data protection for individuals, including the exportation of data outside the European area. Main goal is that individuals gain the control over their personal data and are aware how their data is used by an organization. Even though EU GDPR has no touch points with the security of critical national infrastructure, it is the first regulation on legislative level in the area of IT security and could point to further development in this direction, preferably in security of CNI too.[29]

Considering the cyber security framework of NIST in the adoption of this regulation, there are also points to be criticized. NIST stands for structuring areas of security based on a risk- and security profile. Acknowledging the fact that there is and will be no complete security, this includes also the consideration of a firm's risk appetite. The combination of these define the implementation of the NIST functions. *Protection* is undeniably important but must not be considered the most important and superior. Unfortunately, the main focus in this regulation is put into the protection. There are some kind of dynamic components contained with topics as the Data Protection Impact Assessment (DPIA), it only regulates the change of already assessed systems or the change of data flow of a such.

1.5.2 Committee "ICT Switzerland"

In 1980 the committee "ICTSwitzerland" was founded in order to represent their interests to the public, the government and third parties. Meanwhile it aims to promote digital technologies, digital education and the training of ICT specialists in Switzerland. Last but not least "ICTswitzerland is also committed to the identification and prevention of cyber-risks", it says on the home page.[14]

The committee started as a industry union representing its interest but grew with the importance of the industry. Meanwhile, there are 30 medium and large member firms and 21 associations including representatives from major players like Google, Microsoft and Cisco as well as minor players like OpenSystems, Zuehlke and the Swiss Post. They have set themselves high goals as "to make Switzerland the global leader in cybersecurity" and "promoting the ICT sector as a key pillar of the economy". It is questionable if these goals are reachable or should be their goals in first place since being a key pillar of the Swiss economy would require an enormous demand. They also retained their initial interest to advocate for the ICT industry, promote digital technologies across the country and making sure that there are enough ICT specialists to meet the market demand.[14]

The measures that ICTswitzerland takes in order to reach the goals vary a lot. It reaches from organizing conferences, publications, support work for public authorities up to launching platforms like *digital.swiss*. Digital.swiss is a monitoring service in the web for the digital adoption within Switzerland, indexing various parameters like energy, mobility, health and research & innovation. To take the *legal norms & legislation* as an example, which are currently rated at 28%, are criticized because of digital limits in existing laws and that the legislation should be acting as an "enabler" rather than a "preventer". In the figures 1.10 and 1.11 we can see the progress of two of such topics, on the left side the progress of digitalization in the area of basic infrastructure and on the right side the progress in the area of legal norms & regulations. Although the rating for the latter might not seem well, we can see an upwards trend.[14]

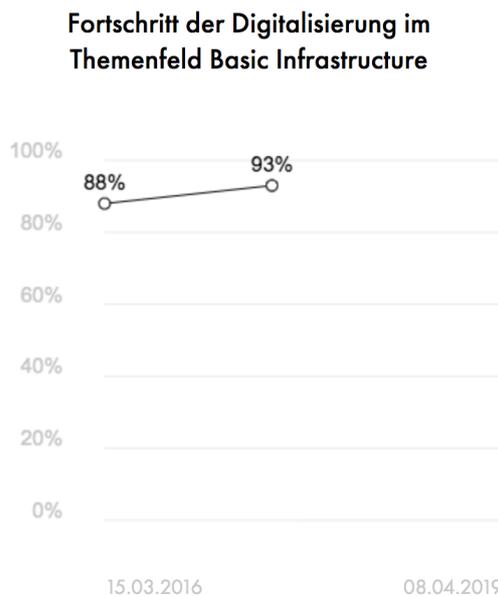


Figure 1.10: Progress of digitalization in the area of Basic Infrastructure [14]

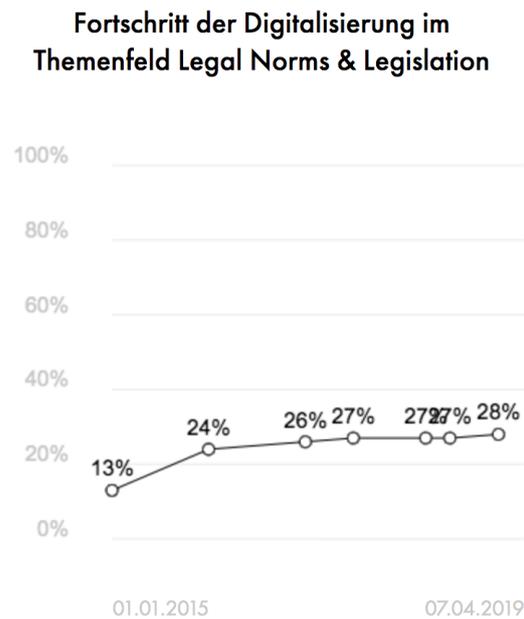


Figure 1.11: Progress of digitalization in the area of Legal Norms & Regulations [14]

1.5.3 Discussion

The growing interest and relevance in the industry is reflected in the creation of committees like ICT Switzerland. While some of their goals seem to sophisticated, they clearly fulfill their purpose. In our opinion, the ICT industry is getting too little attention though. The initiatives taken by ICT Switzerland like digital.swiss and with economisuisse can be beneficial for people within and outside the IT sector. Why would organizations or firms become members in first place? "ICTswitzerland represents your interests", "[..] offers a strong network" and "[..] ensures that young people can qualify as ICT specialists and that existing employees maintain their specialist competences" are the reasons mentioned by the committee on their website. Anyhow, these benefits have to legitimate fees of 50 CHF per ICT employee with a minimum of 25'000 CHF and bring value to the member companies.

1.6 National Strategy for the protection of Switzerland against cyber risks (NCS)

1.6.1 Overview

Along with the process of digitalisation Switzerland touches upon great opportunities and risks. Switzerland wants to benefit from the technological progress to secure and expand welfare for the society, economy and state. Nevertheless, the dependency on information and communication systems raises the risk of breakdowns, disruptions and misuse of such technologies. In order to maintain Switzerland's locational advantage from a social, economical, political and security point of view, measures must be taken to protect Switzerland against cyber crime in all areas. However, the risks coming along with the process of digitalisation cannot be captured completely with those measures. Thus, one achievement of Switerlands is to increase resilience against cyber incidents. [33]

The national strategy for the protection of Switzerland against cyber risks (NCS) sets out how these goals are to be achieved by 2022. Building up on the first NCS from 2012 to

2017, the current NCS further develops the measures of the old NCS to be able to handle changing and intensified threats. Additionally, it's supposed to improve prevention and identification, response and resilience in all areas relevant to cyber risks. Therefore, a strategic framework will be provided. [33]

Switzerland as a Federation, Switzerland's society and the private sectors are all responsible for their protection against cyber risks. However, the NCS raises the need for cooperation between all actors. The NCS states that it becomes a joint responsibility and suggests shared implementation of the measures of the NCS. The federal government, cantons, private sectors and society should implement those measures in close cooperation. [33]

Furthermore, dealing with cyber risks is a big challenge which will continue to grow. Thus, it becomes more and more important to cooperate not only with all actors in Switzerland, but also in international bodies. This is crucial for the process of digitalisation of society and economy. To conclude, the NCS serves as a manual or guideline regarding measures against cyber risks as well as it states to cooperate in an efficient manner on a national and international level. [33]

1.6.2 Achieved Objectives

The most important measures achieved by the NCS 2012 - 2017, which had 16 measures in total, building the basis for the NCS 2018 - 2022 are explained in detail in the following subsections. [33]

1.6.2.1 Building capacities, capabilities and knowledge

The goal of the NCS 2012 - 2017 was to get an overview about the current situation regarding the competencies in the area of cyber risk. It had been shown that there was a lack of necessary resources and expertise. Therefore, the government addressed in a concept how to foster competence building, especially in the areas of education such as universities or other educational institutions. With the implementation of the concept a new educational attainment ICT Security Expert got launched in cooperation with the ICT-Berufsbildung. Further, in the educational sector, the research in universities gets more support from the federal government. Thus, the situation has improved. [6]

1.6.2.2 Building processes, structures and foundations

The NCS emphasizes that cyber risks affects different actors and thus the protection against it becomes a joint responsibility. Therefore it is important to organise the cooperation between different bodies. Further, to create structures and processes. [33] One of the goals was achieved by strengthening the Reporting and Analysis Center for Information Assurance (MELANI) as a platform. Responsible actors from the economy, society and politics can use MELANI as a source of information about cyber incidents. It also supplies an information radar in order to assess the current threat situation, especially regarding critical infrastructures. Nevertheless, MELANI and the radar need continuous improvement as well as adaptation to new cyber risks. [6]

1.6.2.3 Focus on the protection of critical infrastructures

The NCS measures are mainly related to the protection of critical infrastructures. Risk and vulnerability analyses were carried out for the critical sub-sectors, measures were identified, support in the event of incidents was expanded, and a picture of the situation of cyber threats was developed. This work formed the core of the NCS and can now be deepened and expanded. [6]

1.6.2.4 Strengthening cooperation with third parties

A further measure to protect Switzerland was not only to strengthen the cooperation in the federal government, but also with the cantons, private sectors, society and even international partners. Regarding the international cooperation, a concept stating the role, activity and initiatives of Switzerland's Federal Department of Foreign Affairs (EDA) has been written. Further, the Division for Security Policy (ASP) provides the EDA once a year with a status report about the most important activities, processes and initiatives in the area of cyber risks. Also, several boards got created which exchange their knowledge. An example is the group ch@world which serves also as a platform where its members can upload their knowledge or insights in form of reports. All those entities provide a good basis for future development of national as well as international cooperation.[6]

1.6.2.5 Strategic context

The strategic context is defined by several strategies from the federal government. They define the guidelines relevant to the topic of cyber risks. The basic strategies are explained below:[33]

Federal Council report on Swiss security policy: The Security Policy Report 2016 from the federal government defines the basic strategic orientation. It states that the NCS is to be used as the basis for protecting Switzerland against cyber risks. Also that the protection and increasment of the resilience of critical national infrastructures should play an even greater role in the future.

Federal Council strategy for a digital Switzerland: The strategy states that Switzerland intends to make use of the process of digitalisation in order expand welfare in Switzerland and make use of the benefits of technological improvement. Further, transparency and security should be established such that Switzerland's society is able to rettain necessary information regarding cyber risks. Also, it states how Switzerland should act on an international level in the field of digitalisation. This is to be achieved or further developed by the NCS 2018 - 2022. [33]

National strategy for critical infrastrucutre protection: Critical infrastrucutres and it's subsectors and how to protect them is defined in the CIP strategy from the Federal Office for Civil Protection (BABS). The NCS is supposed to cover all the measures mentioned in the CIP. [33]

1.6.3 Need for Action

1.6.3.1 Necessary further development of the NCS

At various levels, there is a need for action in Switzerland. The NCS 2012 - 2017 froms the basis for the further work. It states that it is important that maintaining the status quo is not sufficient. To protect Switzerland adequately against cyber risks, continuous analysis of current threats and strengthening the resilience of critical infrastructures remains an ongoing process. [33]

One one hand the expansion of existing capacities and capabilities as well as making use of the implemented processes, structures and foundations remain goals in the NCS 2018-2022. However, the NCS should serve as a national strategy not only for the ferderal government and critical infrastrucutres, but also for the economy, society and the political sphere, since cyber threats can have deep impacts in all areas. Therefore, the target group of the NCS needs to be extended accordingly and existing cooperation must be expanded. The goal is to create a network in Switzerland to protect all areas against cyber risks. Finally,

the beforehand decentralised organisation, where almost all actors implemented their own measures, should become a central point of contact for the public with a strong strategic management. The society, economy and politics should be informed about the threats and have the availability to stay updated. To conclude, the second NCS continues the work from the first NCS, expands it where necessary and creates new measures. The image extracted from the NCS 2018 - 2022 shows a summary about the need for action. [33]

Level	NCS 2012 - 2017	Need for action
Capacities, capabilities and knowledge	Increased capacities and better knowledge compared to 2012.	Further expansion of capacities and knowledge is necessary to do justice to the intensified threat situation.
Objectives of the NCS measures	Creation of processes, structures and foundations.	Productive use of processes, structures and foundations to reduce cyber risks. The measures and products conceived must be implemented, further developed and, where necessary, supplemented.
Organisational structure	Implementation is carried out in a decentralised manner by the competent authorities.	The increased political, economic and social relevance and rapid development of cyber risks make stronger strategic management of the NCS necessary. The decentralised organisational structure must be supplemented to that effect.
Target groups	Focus on protecting critical infrastructures against cyber risks.	Cyber threats affect the whole of Switzerland, which is why the NCS target group needs to be expanded.
Cooperation	Establishment of cooperation with cantons, the private sector and international partners.	The increase in interconnectivity is strengthening the importance of cooperation at all levels. Existing cooperation arrangements and public-private partnerships must be strengthened and linked in order to create a network to protect Switzerland from cyber risks.

1.6.3.2 Vision and strategic objectives

The Vision and strategic objectives of the NCS 2018 - 2022 determine what is to be achieved by the end of the period. Since many areas of Switzerland are related to the NCS, it is important to have a coherent vision. The Vision of the NCS 2018 - 2022 states: "In exploiting the opportunities of digitalisation, Switzerland is adequately protected against cyber risks and is resilient to them. The capacity to act and the integrity of its population, economy and the state against cyber threats is safeguarded." If the following strategic objectives, extracted from the NCS 2018 - 2022 are achieved, the vision can be realised: [33]

- Switzerland has the competencies, the knowledge, and the capabilities to identify and assess cyber risks at an early stage.
- Switzerland is developing effective measures to reduce cyber risks and is implementing them within the framework of prevention.
- Switzerland has the necessary capacities and organisational structures in all situations to identify cyber incidents quickly and to deal with them even if they persist over an extended period of time and affect different areas simultaneously.
- Switzerland is resilient to cyber risks. The ability of critical infrastructures to provide services and goods remains safeguarded even in the event of major cyber incidents.
- The protection of Switzerland against cyber risks is the joint responsibility of society, the private sector and the state, with responsibilities and competencies clearly defined and put into practice by all those involved.

- Switzerland is committed to international cooperation to increase cyber security. It promotes dialogue in cyber foreign and security policy, participates actively in international expert bodies, and maintains exchanges with other states and international organisations.
- Switzerland learns from cyber incidents at home and abroad. Cyber incidents are carefully analysed, and appropriate measures are taken on the basis of the findings.

1.6.3.3 Principles

The principles define how the goals of the NCS 2018 - 2022 are to be achieved. The approaches to be followed are listed below, stated in the current NCS. [33]

Risk-based, comprehensive approach: This approach aims at improving Switzerland's resilience to cyber risk. A full protection against cyber risk is probably not possible. This approach assumes that. However, the risks can be dealt with to an extent such that the remaining risk is acceptable. The comprehensive part takes all relevant vulnerabilities and threats into account. [33]

Decentralised Implementation: Considering cyber risks and its effects is relevant for almost all actors in Switzerland. Thus, everybody is responsible for their protection. However, the NCS aims to strengthen this shared responsibility by holding actors of the relevant areas to account and by using the existing structures. Therefore, this leads to a decentralised implementation, but with the NCS the implementation is supposed to be controlled in a centralised manner by the strategic management of the NCS. [33]

Subsidiary role of the state: The state only intervenes when the welfare of our society is affected and the private actors are unable or unwilling to solve the problem independently. Therefore, the state provides support, creates incentives or intervenes through regulations. [33]

Public-private partnership and international cooperation: In order to follow the cooperative approach, at national level the government promotes public-private cooperation and further expands cooperation between the federal government, cantons and communes. Also the promotion of the international level with international partners needs to be extended. [33]

Active communication regarding the NCS: With an active communication about the measures of the NCS, the implementation of the NCS becomes transparent. This communication is supposed to be held with the society, the private sector and the policymakers. [33]

1.6.3.4 Target Groups

The NCS and its implementation needs to be considered by the whole of Switzerland. Therefore, the NCS explicitly addresses the following target groups. [33]

Critical Infrastructures: The availability of essential goods and services for Switzerland provided by the critical infrastructures needs to be ensured all the time. They are indispensable for the population and economy of Switzerland. Thus, it becomes the top priority of the NCS to protect the critical infrastructures. [33]

Public Authorities: Some of the critical infrastructures include services of administrations and public authorities. The federal government, cantons and communes are

direct actors of those and thus responsible for it. With the strengthened cooperation the current NCS tries to achieve, also the public authorities are supposed to be protected adequately against cyber risks. [33]

Population: The population as a target group represents the most important one. Besides the protection of the critical infrastructures, which aims towards the protection of the population, the NCS wants to further protect Switzerland's population by focusing on cybercrime in particular. The NCS aim is to sensitize Switzerland's population regarding the safe use of ICT, in an informed and confident manner. [33]

Private Sectors: Switzerland's strong location factor from an economical point of view needs to be ensured or even strengthened. A safe and trustworthy environment is therefore essential. The NCS tries to achieve the safest possible conditions for Swiss companies and supports them in dealing with cyber risks. [33]

1.6.4 Spheres of Action

The strategic objectives and measures of the NCS get distinguished into ten spheres of action, which address different aspects of cyber risks. Thereof, the NCS captures a total of 29 measures. The list of all spheres and measures is shown below. [33]

Sphere of action	Measures
Building competencies and knowledge	<ol style="list-style-type: none"> 1. Early identification of trends and technologies and knowledge building 2. Expansion and promotion of research and educational competence 3. Creation of a favourable framework for an innovative ICT security economy in Switzerland
Threat situation	<ol style="list-style-type: none"> 4. Expansion of capabilities for assessing and presenting the cyber threat situation
Resilience management	<ol style="list-style-type: none"> 5. Improving ICT resilience of critical infrastructures 6. Improving ICT resilience in the Federal Administration 7. Exchange of experience and creation of foundations for improving ICT resilience in the cantons
Standardisation / Regulation	<ol style="list-style-type: none"> 8. Evaluation and introduction of minimum standards 9. Examination of a reporting obligation for cyber incidents and decision on introduction 10. Global internet governance 11. Building expertise on standardisation questions relating to cyber security
Incident management	<ol style="list-style-type: none"> 12. Expansion of MELANI as a public-private partnership for operators of critical infrastructures 13. Development of services for all enterprises 14. Cooperation between the federal government and relevant agencies and competence centres 15. Processes and foundations for incident management of the federal government
Crisis management	<ol style="list-style-type: none"> 16. Integration of the responsible cyber security offices into the federal crisis teams 17. Joint crisis management exercises
Prosecution	<ol style="list-style-type: none"> 18. Picture of the cybercrime situation 19. Investigation Support Network for Digital Law Enforcement 20. Training 21. Central Office for Cybercrime
Cyber defence	<ol style="list-style-type: none"> 22. Expansion of capabilities for information gathering and attribution 23. Capability for implementing active measures in cyberspace under the IntelSA and ArMA 24. Ensuring the Armed Forces' operational readiness across all situations in cyberspace and regulating their subsidiary role in support of the civilian authorities
Active positioning of Switzerland in international cyber security policy	<ol style="list-style-type: none"> 25. Active shaping of and participation in processes of foreign cyber security policy 26. International cooperation to build and expand cyber security capacities 27. Bilateral political consultations and multilateral dialogues on foreign cyber security policy
Public impact and awareness raising	<ol style="list-style-type: none"> 28. Creation and implementation of a communication concept for the NCS 29. Raising public awareness of cyber risks

Below is a selection of three measures explained in more detail.

3. Creation of a favourable framework for an innovative ICT security economy:

The goal is to increase ICT security solutions produced in Switzerland. Therefore, the cooperation between the government, the private sector and research needs to be strengthened. The basis is provided by research in the field of cyber security. It is important that specialist knowledge is available such that an environment for the exchange between research and the private sectors, respectively innovative companies in the area of cyber risks is provided. Therefore, Switzerland aims towards becoming an attractive location for companies in the field of ICT security. The basis is built up to promote innovative start-ups, in cooperation with universities. [33]

10. Global Internet Governance: Within the scope of regulation falls the internet governance processes created by the UN World Summit on the Information Society (WSIS). These processes deal with the development of principles, norms, rules and decision-making mechanisms for the development and use of the internet at interna-

tional level. Therefore, the NCS states the Switzerland should actively participate in such bodies in order to represent Switzerland's ideals of freedom, democracy and personal responsibility, equal opportunities, security, human rights and the rule of law. The goal is to at least create a compatible environment regarding international regulations and Switzerland's standards and ideals. [33]

- 17. Joint crisis management exercises:** It is important that crisis teams are supported by specialist knowledge and intensive cooperation by all actors such as the federal government, the cantons, communes and even the private sectors. It needs to be ensured that all relevant information is available in order to handle the crisis adequately. Therefore, the NCS sets out the goal to test its crisis management against cyber risks. This will be held in joint exercises between all relevant actors including the representatives of critical infrastructures. [33]

1.6.5 Discussion

Having the NCS 2018 - 2022 at hand, using the NCS 2012 - 2017 as its basis, Switzerland has put lots of effort in the protection of Switzerland against cyber risks. Important goals from the first NCS such as identifying necessary capacities and knowledge in the area of cyber risks as well as the identification of the need for cooperation of all the actors in Switzerland as a federation have been achieved in the first NCS. This builds the needed basis in order to be resilient against cyber incidents. In the context of Switzerland as a federation the approaches suggested in the NCS to implement the measures in a decentralised manner but with a centralised strategic management seems to be the correct way to encounter cyber risks. Further, the creation of the ICT Security Expert education, as well as the aim to strengthen Switzerland as a location for innovative ICT security companies in cooperation with universities and the private sectors help to further improve the protection against cyber risks which continuously encounters changed threats. However, the bottleneck might be the bureaucracy regarding the cooperation between all actors as well as the pace in which Switzerland is able to introduce the necessary regulations of laws regarding cyber risks.

Bibliography

- [1] <https://www.protonmail.com>.
- [2] Was ist ein man-in-the-middle-angriff und wie kann man sich schützen? <http://www.was-ist-malware.de/it-sicherheit/man-in-the-middle-angriff/>.
- [3] *Proceedings. 1989 IEEE Symposium on Security and Privacy*. IEEE Comput. Soc. Press, 1-3 May 1989.
- [4] Nationale strategie zum schutz kritischer infrastrukturen 2018–2022, 2017. <https://www.babs.admin.ch/de/aufgabenbabs/ski/nationalestrategie.html>.
- [5] Swiss organization better prepared to predict and resist cyber attacks - but still a long way to go: Ey global information security survey, 2017. <https://www.ey.com/ch/en/newsroom/news-releases/news-release-ey-swiss-organizations-better-prepared-to-predict-and-resist-cyber-attacks>.
- [6] Wirksamkeitsüberprüfung ncs, 2018. https://www.isb.admin.ch/isb/de/home/themen/cyber_risiken_ncs/ncs_strategie-2012/wirksamkeitsueberpruefung.html.
- [7] Critical national infrastructure, <https://www.cpni.gov.uk/critical-national-infrastructure-0>, Last visited: 2019-04-17.
- [8] Cyber risk survey report 2018: Cyber risk from a swiss perspective, https://www.kessler.ch/fileadmin/09_PDFs/KS_Cyber_Report_2018_EN.pdf, Last visited: 2019-04-17.
- [9] Analysis of the cyber attack on the ukrainian power grid, March, 2016. https://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf.
- [10] Burkhard Stiller. Cecn, slides, 2017.
- [11] DAVID BISSON. 6 common phishing attacks and how to protect against them, 2016. <https://www.tripwire.com/state-of-security/security-awareness/6-common-phishing-attacks-and-how-to-protect-against-them/>.
- [12] Department of Health, editor. *Investigation: WannaCry cyber attack and the NHS*. 2018.
- [13] Eliza Strickland. Explainer: What went wrong in japan’s nuclear reactors, 2011. <https://spectrum.ieee.org/tech-talk/energy/nuclear/explainer-what-went-wrong-in-japans-nuclear-reactors>.
- [14] ICT Switzerland. Website of the committee ICT Switzerland - ictswitzerland.ch, 2019. <https://ictswitzerland.ch/>.
- [15] Imperva. What is sql injection. <https://www.imperva.com/learn/application-security/sql-injection-sqli/>.

- [16] International Standards Organisation (ISO). SO/IEC 27000:2018, <https://www.iso.org/standard/73906.html>.
- [17] Markus Jakobsson. Modeling and preventing phishing attacks. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Andrew S. Patrick, and Moti Yung, editors, *Financial Cryptography and Data Security*, volume 3570 of *Lecture Notes in Computer Science*, page 89. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [18] Kaspersky. What is a trojan virus. <https://www.kaspersky.com/resource-center/threats/trojans>.
- [19] Kaspersky. Was ist ein man-in-the-middle-angriff?, 2013. <https://www.kaspersky.de/blog/was-ist-eine-man-in-the-middle-attacke/905/>.
- [20] Dejan Kosutic. Which one to go with - cybersecurity framework or iso 27001?, 2014. <https://advisera.com/27001academy/blog/2014/02/24/which-one-to-go-with-cybersecurity-framework-or-iso-27001/>.
- [21] Lorenz Hilty. Wirtschaftsinformatik1, slides, 2018.
- [22] Leandros A. Maglaras, Ki-Hyung Kim, Helge Janicke, Mohamed Amine Ferrag, Stylianos Rallis, Pavlina Fragkou, Athanasios Maglaras, and Tiago J. Cruz. Cyber security of critical infrastructures. *ICT Express*, 4(1):42–45, 2018.
- [23] Michael Holloway. Stuxnet worm attack on iranian nuclear facilities, 2015. <http://large.stanford.edu/courses/2015/ph241/holloway1/>.
- [24] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security & Privacy*, 1(4):33–39, 2003.
- [25] Igor Nai Fovino, Luca Guidi, Marcelo Masera, and Alberto Stefanini. Cyber security assessment of a power plant. *Electric Power Systems Research*, 81(2):518–526, 2011.
- [26] National Institute of Standards and Technology. Cybersecurity Framework of NIST. <https://www.nist.gov/cyberframework>.
- [27] Nicolas Falliere, Liam O Murchu. W32.stuxnet dossier, 2011. <https://nsarchive2.gwu.edu//NSAEBB/NSAEBB424/docs/Cyber-044.pdf>.
- [28] Norton. What is a distributed denial of service attack (ddos) and what can you do about them? <https://us.norton.com/internetsecurity-emerging-threats-what-is-a-ddos-attack-30sectech-by-norton.html>.
- [29] Official website of the EU GDPR. www.eugdpr.org. <https://eugdpr.org/>.
- [30] Roger A. Grimes. 8 types of malware and how to recognize them, 2018. <https://www.csoonline.com/article/2615925/security-your-quick-guide-to-malware-types.html>.
- [31] Savita Mohurle and Manisha Patil, editors. *A brief study of Wannacry Threat: Ransomware Attack 2017*. 2017.
- [32] Stuart Staniford and Stefan Savage, editors. *Proceedings of the 2003 ACM workshop on Rapid Malcode - WORM'03*, New York, New York, USA, 2003. ACM Press.

- [33] SN002 - Nationale Strategie zum Schutz der Schweiz vor Cyber-Risiken (NCS), 2018, https://www.isb.admin.ch/isb/de/home/ikt-vorgaben/strategien-teilstrategien/sn002-nationale_strategie_schutz_schweiz_cyber-risiken_ncs.html, Last visited: 11-04-2019.

Chapter 2

Approaches and Challenges in Blockchain Scalability

Mesut Ceylan, Catharina Dekker, Luka Popovic, Nathalie Torrent

Most blockchain-based networks have not yet addressed the issue of scalability. This paper introduces various solutions in the context of blockchains that attempt to solve the scalability issues at hand. The main obstacles of blockchain technologies revolve around the available throughput, the cost of transactions and the capacity of such operations on the blockchain. Possible scalability solutions (e.g. on chain, off chain, child chain, side chain and scalable consensus mechanism) will be described, and a brief survey of the different types is provided. Furthermore, detailed examples provide a comparison and analysis of the different approaches that are mainly concerned with trying to improve the scalability of popular blockchain platforms such as Bitcoin and Ethereum.

Contents

2.1	Introduction	41
2.2	Scalability in the Context of Blockchain	41
2.3	On Chain	43
2.3.1	Sharding	43
2.3.2	Rapid Chain	44
2.3.3	BitCoin NG	46
2.3.4	Segregated Witness	49
2.4	Off Chain	50
2.4.1	The Bitcoin Lightning Network	51
2.4.2	Raiden Network	52
2.5	Side Chain	53
2.5.1	Rootstock	53
2.5.2	Blockstream	55
2.6	Child Chain	57
2.6.1	Ethereum Plasma	57
2.7	Scalable Consensus Mechanisms	59
2.7.1	Delegated Proof-of-Stake	59
2.7.2	Byzantine Fault Tolerance	60
2.8	Conclusion	63

2.1 Introduction

In the last years, cryptocurrencies have gained a lot of interest and have especially attracted the interest of non-IT people, due to the popularity of trading and speculating with cryptocurrencies. The blockchain, which is the underlying technology of most cryptocurrencies, has two main advantages: decentralization and immutability. All the nodes of a blockchain are running the same consensus protocol and form together a peer-to-peer (P2P) network. Economical incentives included in the consensus protocol makes the network decentralized as nodes are responsible for validating newly added information. The second important feature, that comes from the structure of the blockchain, is the immutability: once a transaction is stored in a block it is not possible to modify it since it would require to modify all the previous blocks. However, blockchain networks used to deal with cryptocurrencies are becoming increasingly slower, due to the increasing amount of transactions. The transactions are stored in blocks, which are of limited size and created at a regular time interval. For example, in Bitcoin, the block size is limited to 1 Megabyte (MB) and a new block is mined every ten minutes. This implies that Bitcoin can only handle 7 transactions per second. Having such a hard block mining restriction causes the pool of unvalidated transactions to increase, since it can not handle the growing need of transactions. [1].

In the context of networks, scalability is defined as the capability to handle a growing amount of work and the ability to accommodate that growth. A system would be considered scalable if it is able to handle an increased load and increase its output in that situation [3]. Currently the most common blockchain technologies, which are Bitcoin and Ethereum, are not considered as scalable as they are only able to handle a very limited amount of transactions that bottlenecks the growth and adoption of this technology. At first, this paper will introduce the concept of scalability regarding blockchains in details. Then some solutions regarding how to address these issues, and the advantages and limitations of the solutions will be presented.

2.2 Scalability in the Context of Blockchain

Currently, a scalability trilemma can be observed in public blockchains (Figure 2.1). There are three properties that participants wish to see in a blockchain-based network: speed, security and decentralization. Bitcoin is able to handle 7 transactions per second, however it is unacceptably slow. Ripple, which consists of a payment infrastructure RippleNet, which runs on blockchain technology, and a cryptocurrency called XRP, can handle around 1500 transactions per second [2]. Clearly Ripple is very fast in comparison to Bitcoin (1500 transactions per second versus 7) but is not decentralized. Evidently, a blockchain protocol so far has not been able to have the desired three out of three properties. The existing protocols and solutions out there are not completely scalable because they can not cater to the three properties that consumers want to see [4].

Currently blockchain technologies have very limited scalability. Bitcoin is able to handle 3 to 7 transactions per second, Ethereum 23-25 transactions per seconds while Visa is able to process more than 20'000 transactions per seconds [5]. During Christmas holidays 2013, Visa could face more than 47'000 transactions per seconds. Clearly achieving the same capacity as Visa using blockchain technology is not feasible today. In order for Bitcoin to proceed with approximately the same transactions per seconds as Visa, it would require blocks of 8 MB, compared to the current 1 MB, being generated every 10 minutes. With this modification, the blockchain would become extremely centralized as mining blocks would become more intensive [10]. At present, the two common blockchain technologies are nowhere close to replacing Visa in the payment processing platform.

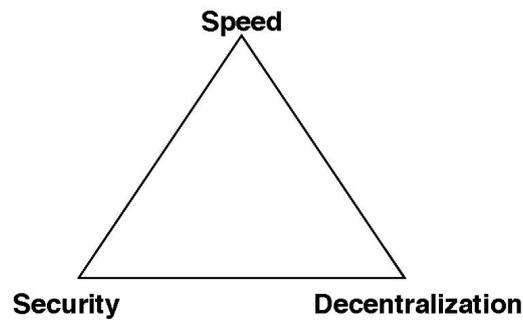


Figure 2.1: Scalability Trilemma

In the last years, the Bitcoin community was engaged in a debate regarding the block size. Some of them thought that increasing the block size would be a good way of allowing more transactions to be done on the blockchain. For the others, increasing the block size wasn't a viable solution, as it was only a short term fix and would lead to a more centralized blockchain: it would require even more computing power and electricity to mine blocks, therefore ensuring that only the biggest, most powerful miners could participate in the network. Furthermore, by increasing the weight of the blockchain, a new miner would take weeks to download the whole blockchain, and would detrimentally limit new users from joining the network. Therefore increasing the block size heavily influences the decentralization characteristic of blockchain technologies [4].

As explained, blockchain solutions have serious advantages such as the decentralization and immutability. Often, one of these characteristic is given up in order to achieve some scalability. Some of the scalability limitations that blockchain technologies are facing are the low transactions per second, the block size issue, the increased chain size and the electronic signatures size [6]. These four scalability issues can be grouped into three categories of scalability problems: the throughput, the cost and the capacity.

The throughput or processing speed is a function of block size and inter-block interval. Transactions need to be validated by being included into a block. The block size limit only allows to store a limited number of transactions into a block, therefore transactions are in a latent stage while waiting to be included into a block. This influences the throughput because by having a bigger block size it would be possible to store more transactions in a block, and therefore the throughput would be higher. However, at the same time the blockchain would become more centralized, since the blocks are bigger and would be more difficult to mine. The throughput is also influenced by the generation time of a block, which is the time between the creation of two blocks. By generating blocks more frequently, it would be possible to validate transactions faster while also increasing the throughput. However, this would lead to a larger number of forks.

A fee must be paid with every transaction, much like a transaction fee with a credit card. The block size limits the number of transactions that can be stored into one block, this limitation makes it harder to carry out transactions and causes fees to increase. Miners prefer to have large transactions in blocks because a larger fee will be paid. Micro transactions, which are considered as a light transaction, will always be delayed since miners do not receive enough incentives to include those transactions in the next block.

The capacity in this context, is the capacity needed to maintain the blockchain, this is directly related to the size of the whole blockchain. Currently, Bitcoin and Ethereum were weighing 248 Gigabytes (GB) and 208 GB [7], [8]. Due to the size of the blockchain, a new miner who wants to join the network would take a few weeks to download the whole blockchain. The larger the blockchain is, the slower the propagation into the network will be. As a result the increasing size of the blockchain produces a centralized network where only the users with high resources are able to participate in the network.

Scalability especially matters in the context of processing payment transactions or handling with smart contracts. For some other blockchain applications, scalability is less important. For example with Everledger, a blockchain solution that aims to guarantee the provenance of assets, scalability plays a less important role. They propose solutions to keep track of the provenance of diamonds or art pieces in a transparent way. In this context, the scalability level matters less as transactions aren't occurring every second [9]. This paper will focus on the solution to address the scalability issue in the context of payments and smart contracts.

There are several possible viewpoints available to approach the scalability issue, which will be discussed in this paper. All of the solutions provide modifications that either affect the main chain, or are techniques that happen in the background and later on get pushed to the main chain. The techniques that will be discussed are on chain, off chain, side chain, child chain and scalable consensus algorithms. All of these techniques aim to modify the factors discussed that affect scalability: throughput, costs and capacity.

2.3 On Chain

On chain is a scalability approach in which modifications are only made to the blockchain. The possible modifications could be: changes of the block size or changes in the protocol of the blockchain [6].

By changing the block size of the blockchain, the transaction limit of the chain increases, achieving an increased throughput. Since the throughput is higher, the transaction costs will be lower, due to less competition to get a transaction included in the main chain. However, this comes at a cost, as the block size increases, the propagation speed becomes slower. Eventually resulting in multiple forks and a higher probability of orphan blocks appearing during this process. A slow propagation speed due to a large block size will ultimately lead to centralization. This effect happens because the block size increases the cost of mining and maintaining the ledger. Meaning that the size of the ledger will be too big for a participant with a conventional computer with a home broadband connection to download. If this is the case, the blockchain will not be attainable to participants with standard consumer hardware, thereby leading to centralization because of the prerequisites needed to join the network [6].

2.3.1 Sharding

Another method to implement the on chain technique is to alter the protocol of the blockchain, instead of changing underlying parameters, for example changing the block size. Sharding is an example of an on chain solution which changes the protocol to increase scalability. It is an approach that Ethereum is currently looking into for implementation to address their scalability issues. With Sharding, a database is split up into several pieces, called shards, where each shard has its own history of transactions and states. Groups of participants are formed called committees, over which the shards will be distributed. Each committee verifies a shard, side-by-side, instead of one after the other. This speeds up the verification process exponentially, since committees can verify transaction in parallel, while being relieved of the size burden. However, a negative aspect of Sharding is that a malicious adversary could achieve control over a shard, by not truthfully disclosing information, and thereby break the data integrity [6].

2.3.1.1 Elements of a shard

Each shard is verified by a group of nodes called committees. The shard header will contain the information that the committee will verify, which includes the current state

of the shard, the state of the shard after all the transactions are processed and the digital signatures of at least $2/3$ of the other shards in order to verify that the shard is not corrupted.

Once the committees have verified the shards, they pass that information on to super nodes. They will continue to process the transactions in the shard and maintain a full record of everything that's happened. They then create a single block that can be added onto the blockchain. This block will only be valid if all the transactions are valid in the shard headers, the current state of the collation is the same as the state in the header, and if the shards are digitally signed by $2/3$ of the committees.

In order to keep track of transactions between shards, receipts are needed. If shard 1 wants to pay an amount to shard 2, the following procedure will happen. Shard 1 will receive a transaction and generate a receipt which reflects the changes in its balance. This receipt will be stored in the merkle root, where it can be easily verified. After that, shard 2 will receive a transaction with receipt data along with a way to verify whether the receipt has been spent by shard 1 or not. When the transaction is finalized the balance is changed in shard 2, and the receipt is marked as spent [12].

2.3.1.2 Advantages and Limitations

Through these 3 components, the committees, super nodes and receipts, transactions can be verified in parallel, thereby increasing the throughput because it requires less communication between participants, less computation and less storage per node, which in theory, would allow the system to scale to large networks. However, simple Sharding protocols still allow for a bottleneck to occur because they need a linear amount of communication per transaction, therefore only benefiting partially of the benefits that Sharding could bring. The more basic Sharding protocols also have a weaker security protocol. They have a byzantine fault resiliency of $1/8$, or they rely on a trusted set-up between participants, which limits their applicability to large open networks, such as payment systems [6].

2.3.2 Rapid Chain

In [12] a new sharding-based protocol called RapidChain, which comes with the benefits of sharding, while avoiding a bottleneck communication issue, and not assuming a trusted set up. RapidChain has an empirically evaluated throughput of 7,300 transactions per second (tps) in a network of 4000 nodes, this is a thousandfold increase in throughput compared to Bitcoin. It has a Byzantine fault tolerance of 33% for its participants. RapidChain's impressive features are achieved by avoiding gossip transactions to the entire network by using a cross-shard verification technique [12].

RapidChain creates different committees than traditional sharding protocols. Raykova et al. describe the parameters as: *"Let n denote the number of participants in the protocol at any given time, and $m < n$ denote the size of each committee. Rapid-Chain creates $k = n/m$ committees each of size $m = c \cdot \log(n)$ nodes, where c is a constant depending only on the security parameter (in practice, c is roughly 20)"* [12]. So by allowing nodes to only communicate with a logarithmic amount of other nodes, compared to a gossip-to-all network, RapidChain applies its cross-shard verification technique, which achieves better sublinear communication, security and throughput than other sharding protocols.

2.3.2.1 Design Components

RapidChain's protocol starts with a Bootstrap, then continues into epochs, where each epoch has multiple rounds of Consensus, and finishes with a Reconfiguration stage. The

bootstrap, which occurs once at the beginning, allows all nodes to agree on a root node, which is a group of $O(\sqrt{n})$ nodes. The root node's responsibility is to establish a reference committee, of size $O(\log n)$. The reference committee then continues by creating k committees each of size $O(\log n)$. Here is where one of the key-factors of RapidChain comes in, since we create committees of size $O(\log n)$, which limits our gossiping-radius, we prevent gossiping to the entire network.

After the bootstrapping phase, RapidChain continues into the consensus part of the protocol. This occurs when all members of a committee have finished their epoch reconfiguration. They then wait for the external users to submit transactions, through a P2P discovery protocol, to a subset of nodes. Those nodes then forward the transaction to a committee that will be responsible for processing them. That committee will proceed with an intra-committee consensus protocol to approve the transaction and add it to its ledger.

RapidChain's protocol concludes with the Reconfiguration stage, which allows new nodes to establish identities and join the existing committees. This is done while ensuring that the committees maintain their 1/3 resiliency, thereby making it a secure system for open networks.

2.3.2.2 Cross Shards

While sharding achieves a reduction in communication, computation and storage requirements for each node, it makes the verification of each transaction more difficult because the input and output of each transaction could reside in multiple committees. Each transaction has a unique ID, a list of inputs and a list of outputs, as seen in Figure 2.2. All inputs to the transaction must be unused coins from previous transaction, called unspent transaction outputs (UTXO's). The nodes have to verify the validity of a transaction by checking two things: if the input is unspent and whether the sum of the inputs is more than the sum of the outputs. If the transaction is deemed as valid, the nodes will add the transaction to the next block that they are accepting. The transactions are distributed among the committees, which will then store the transaction outputs in their UTXO databases, based on the transaction ID. The transactions are only stored if they have the committee ID as their prefix in their ID's [12].

During the verification process, multiple committees could be involved with ensuring that all the input UTXO's for the transaction sent by the user, \mathbf{tx} , are valid. The committee that stores \mathbf{tx} and the possible UTXO's is called the output committee, while the committee that stores the input UTXO's for the \mathbf{tx} is called the input committee. In the RapidChain protocol, the user communicates with any committee who routes \mathbf{tx} to the output committee via the inter-committee routing protocol. By constructing the cross sharding protocol in this way, the user does not have to attach any proof to \mathbf{tx} , and therefore gets rid of an extra burden to ensure that the user nodes retain their "lightness". RapidChain's protocol for obtaining the proof-of-acceptance is very different from another popular blockchain platform OmniLedger's, where each input committee provides a proof-of-acceptance if the UTXO is valid, the user then has to collect it from every input committee and submit that proof to the output committee for validation. This protocol creates a large communication overhead and it requires the user to collect the validation proof, thereby making OmniLedger's network heavier than RapidChain's.

Limiting the amount of inter-committee communication to verify transactions is one of the most important features of RapidChain. A crucial characteristic is the number of shards. In RapidChain's target network of 4000 nodes and 16 committees, it is expected that 99.98% of all transactions are cross-shards, therefore at least one of every transaction's UTXO's is expected to be located in a different shard than itself. By not having to gossip to the entire network, a serious reduction is achieved in RapidChain's protocol [12].

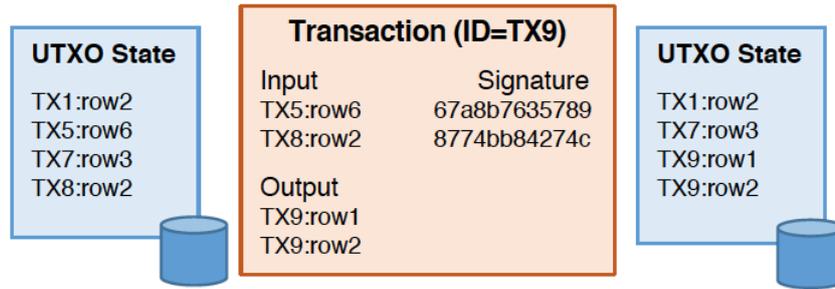


Figure 2.2: Rapid Chain, source: [12]

2.3.2.3 Byzantine Resiliency

Separating the nodes into committees to achieve scalability gives rise to a new problem, where corrupt nodes could strategically leave and rejoin the network in order to eventually take control over one of the committees. One way to deal with this security issue is to re-elect all committees periodically faster than the corrupt nodes' ability to generate churn. However, this is not a very effective solution since it requires a lot of overhead storage and a need to maintain a separate ledger for each committee. Therefore, RapidChain has built upon the Cuckoo rule in order to re-organize only a subset of committee members [13]. This is done in such a way that committees are balanced with respect to their size as nodes join or leave the network [12].

2.3.2.4 Advantages and Limitations

RapidChain is a promising sharding-based protocol, requiring only sublinear communication, thereby reducing communication overhead by 3 times for gossiping of transactions to other committees. This is achieved by introducing the cross-shard verification technique, which allows committees to discover each other while only knowing about a logarithmic amount of other committees, which effectively reduces the latency and storage. Compared to commonly having several "gossip-to-all" invocations for committees to find each other. While reducing the communication, it is also the first sharding-based protocol that does not assume a trusted set-up, with a byzantine fault tolerance of $1/3$. RapidChain has been tested in experiments with up to 4000 nodes.

RapidChain introduced a novel sharing-based protocol to mitigate the scalability issues that public blockchains are currently facing. Even though RapidChain has some impressive speed improvements and security benefits, it has only been tested for up to 4000 nodes. Thereby not making it an applicable protocol, so far, to be implemented in a network along the size of Bitcoin, which has been estimated by Bitnodes to have a size of around 9500 nodes [14]. RapidChain is able to increase speed, and security, however only to a certain extent. It could also be argued that RapidChain's Byzantine Fault Tolerance of $1/3$ is also not secure enough for an open network and leaves space for attacks.

2.3.3 BitCoin NG

Bitcoin NG [15] is a Byzantine fault-tolerant protocol that shares the same trust model as Bitcoin [16]: both imply trusting the peers of the network to validate transactions. The Bitcoin NG protocol has been developed, with the goal in mind, to scale better than Bitcoin. Improving scalability can be accomplished by splitting the current Bitcoin operations into two types: leader election and transaction serialization. In brief, the

Bitcoin NG protocol divides time into epochs where leaders are chosen. The leader who gets elected for one epoch can serialize transactions up, until the new leader is chosen.

2.3.3.1 Leader Election

Randomly and at infrequent time intervals, a leader election finds place. All the miners try to solve the crypto puzzle at hand, the miner that solves it first becomes the leader. A new key block is generated and added onto the blockchain. The key blocks are the new types of blocks introduced in this protocol. Once the construction of the key block has happened, the leader election is done and the new epoch starts. At this point, the leader can start to serialize transactions.

2.3.3.2 Transaction Serialization

While the leader has the ability to serialize transactions, he is able to append multiple micro blocks on the blockchain. The micro blocks are the second type of blocks introduced by the Bitcoin NG protocol. The micro blocks are aimed to store transactions in a cheap and effortless way. The transactions in the micro blocks are stored and are signed with just the private key of the leader. The core idea behind the micro blocks is that the leader can generate many micro blocks during its epoch so that more transactions can be validated in a short period of time. As there is only one leader, he can generate the micro blocks unilaterally.

2.3.3.3 End of an epoch

An epoch ends when a new leader election takes place. Often short forks are required to make sure that all the micro blocks are added on the blockchain before a new key block is added. It is necessary to consider the propagation time into the network to make sure that all the nodes are aware of the previously generated micro blocks.

2.3.3.4 Remuneration

Leaders are rewarded for their work. Their reward (named remuneration in the context of the Bitcoin NG protocol) consists of two parts. On the one hand, they get a certain amount of cryptocurrency as a reward for the creation of key blocks, which is resource-intensive as Proof of Work (PoW) is required. On the other hand, leaders receive a part of the transaction fee. 40% of a transaction fee is given to the current leader, while 60% will be given to the following leader.

The global functioning of the Bitcoin NG protocol can be seen on the Figure 2.3. The blue squares represent key blocks which contain the public key of the leader. Approximately each 10 minutes, a new leader election takes place. Once a miner has become the leader, he can generate micro blocks, the black circles on the figure, every 10 seconds. Generating micro blocks is light because the leader just needs to sign it with his private key (sig_A). Then the fees for each micro blocks are separated between the current and the previous leader.

2.3.3.5 Types of block

The main difference with the current Bitcoin protocol ([16]) is the distinction made between the key blocks and the micro blocks.

Key Blocks

The key blocks are extremely similar to the blocks in the Bitcoin blockchain. They contain a reference to the predecessor header, a coinbase transaction to reward the miner that

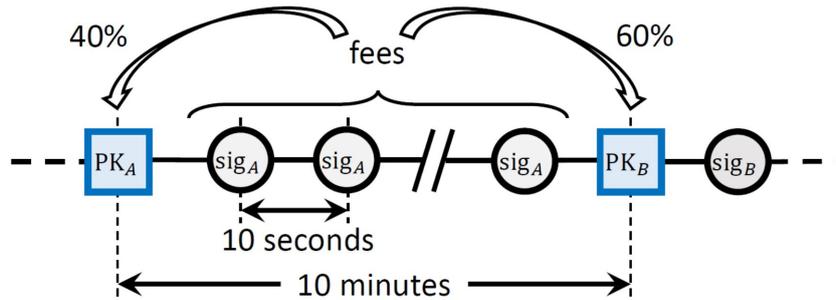


Figure 2.3: Bitcoin NG blockchain, source: [15]

solved the crypto puzzle, the PoW which is the solution of the crypto puzzle. The main distinction between Bitcoin NG and Bitcoin is that Bitcoin NG introduced key blocks which contain the public key of the leader. The key blocks are considered as heavy, since mining is required to create them.

Micro Blocks

The micro blocks also have a typical block structure except for the fact that they weren't mined under the PoW consensus algorithm. Essentially they contain a header containing the reference to the previous block, the private signature of the leader, the hash of the ledger entries and the ledger entries. The main difference with the key blocks is that the micro blocks do not contain any PoW and that they contain transactions. Moreover, the micro blocks are digitally signed with the miners' private key. This allows to guarantee the origin of the micro blocks.

As micro blocks don't need PoW to be generated, they are considered as weightless by the authors of the Bitcoin NG protocol [15]. The figure 2.4 represents two chains that are considered as equally weighted, as each contain one key block (square). No statement is made about the exact size of the micro blocks. To sum up, the biggest advantages of micro blocks is the quick generation time as PoW is not required and the fact that they don't add weight on the blockchain.

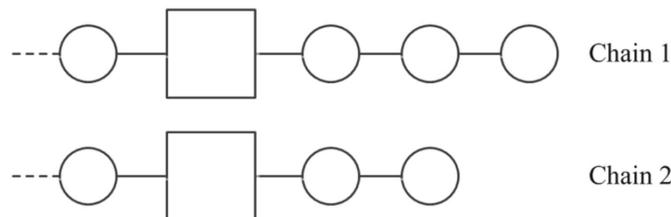


Figure 2.4: Bitcoin NG micro blocks, source: [17]

2.3.3.6 Advantages and Limitations

The developers of the Bitcoin NG protocol ran an experiment comparing their newly developed protocol with Bitcoin [16]. The experiment consisted out of a chain of 1000 nodes which was approximately 15% of the current Bitcoin blockchain at that time.

The Bitcoin NG developer achieved a better performance than Bitcoin because of the decoupling between the leader election and the transaction serialization. The paradigm is drastically different between the two protocols: in Bitcoin there is a system freeze between two leader elections and the miners serialize the history of transactions whereas in Bitcoin NG transactions are continually being processed. During the tests, Bitcoin NG achieved a bandwidth that was 3.5 times bigger than Bitcoin's bandwidth (with 1 MB blocks being

mined every 10 minutes). Clearly Bitcoin NG outperforms Bitcoin as it can achieve a higher throughput without a security deterioration. Furthermore we see that in Bitcoin increasing the bandwidth leads to decreasing the block interval and increasing the block size. This implies a lower security as mining power is lost, which makes the system more vulnerable to attacks. In their experiment, the developers found a deterioration of security related metric in Bitcoin when increasing the bandwidth. It is also worth to mention that hard forks remain a rare event in Bitcoin NG, even if micro blocks are generated faster than normal blocks in Bitcoin. This fast generation of micro blocks also allows Bitcoin NG to process more transactions in a shorter period of time, which leads to a better throughput for the network. According to the authors, *"Bitcoin NG scales optimally with bandwidth limited to the capacity of individual nodes and latency limited only by the propagation time of the network"* [15]. They achieved a higher throughput and a lower latency than Bitcoin, while achieving the same security level. Moreover as micro blocks do not contain PoW, they will not affect the weight in a significant way on the current blockchain. Therefore the capacity of the blockchain will remain the same. However, there are some limitations to the scalability while using Bitcoin NG, the bandwidth is limited to the processing capacity of the network. Furthermore, the propagation delay of the network could also slow down the protocol. One of the major drawbacks, which is slowing down the realistic usage of this protocol, is that a hard fork would be required in order to use it on the Bitcoin blockchain. It would be complicated to achieve a consensus among all the members of the network. On the other hand, there are still some security issues about the double spending attacks. As the micro blocks are not secured by a PoW, it would theoretically be possible for a leader to generate loads of micro blocks at the same time with contradicting information and therefore allow double spending. A fraudulent leader could do some forks while generating micro blocks in order to double spend coins. Nonetheless, the developers considered that issue and provided an incentive for the leader to remain honest. If a double spending is detected by the other nodes of the network, a unique poison transaction is created. This transaction is placed by the subsequent leader and contains a Proof of Fault which consists of the header of the first block in the pruned branch. This poison transaction, that must be placed before the fraudulent leader spent his revenue, will destroy the revenue (reward and fees) of the double spending leader. The subsequent leader placing the poison transaction will get a fraction of the cancelled revenue of the fraudulent leader. Despite that poison transaction, Bitcoin NG remains more vulnerable than Bitcoin to double spending attacks [15]. Despite this major security issue, Bitcoin NG remains a robust and scalable protocol that allows a higher throughput and a lower latency, which could be considered as a possible protocol to solve the current scalability problem that Bitcoin is facing. The authors of the Bitcoin NG paper think that their solution, that trades off the double spending security in order to increase the throughput, would make the global usage of Bitcoin possible for various applications such as payments, digital asset transactions and smart contracts.

2.3.4 Segregated Witness

Segregated Witness or SegWit is an implemented solution that has been integrated into the Bitcoin blockchain by a soft fork in 2017. The idea is to develop a new transaction format in order to go behind the maximal block size of 1 MB. Based on the fact that signatures count for up to 70% of the block size in a transaction makes it clear that blocks mainly consist out of signature information. So the idea of SegWit was to define a new transaction format so that more transaction could be stored in a block, by altering the signature in such a way that blocks can be used to their full potential. The idea was to remove the signature from the transaction and store the signature in a special structure called Witness. This Witness structure will count only for 1/4 of the block's actual size

and will be appended to blocks separately. This removal of the signatures which is an on chain solution allows us to store more transaction into the 1 MB blocks [18].

2.3.4.1 Advantages and Limitations

With this new structure the block size could be extended to 4 MB, which implies a higher transaction speed. Segwit allows to process 1.7 to 3 times more transactions per second, thereby obtaining a reduction of the transaction fees. Clearly, SegWit increases the throughput and reduces the cost. However, the biggest advantage of SegWit is that it solves transaction malleability. Surprisingly, Bitcoin transactions are signed but the signature does not protect the whole data. It remains possible to modify the transaction identifiers transaction id (txid) without modifying the current effect of the transaction, meaning that the amount is still transferred to the original owner (third-party malleability). At present, the transactions respecting the SegWit protocol are resistant against transaction malleability. By moving the signatures into the Witness, the txid can be calculated independently (without being based on the Witness content). With this resistance against transaction malleability, it would be easier to develop protocols to allow Lightning Network or Side Chain solutions on Bitcoin.

On the other hand, those modifications increase the code complexity of Bitcoin blockchain. It can lead to some problems as SegWit transactions are separated from non-Segwit transactions [6]. SegWit does not address the capacity issue of the scalability problem, the capacity remains the same.

The most important step forward that this protocol proposes is the solution regarding the transaction malleability, which will allow users to use other solutions to improve the scalability of the Bitcoin solution.

2.4 Off Chain

Off Chain is a scalability approach in which transactions are only added to the main chain after already being processed outside of the main chain [10].

Bidirectional payment channels - micropayment channels

As explained in the [10] the main idea of the off chain solution is to use separate channels for processing transaction between participants in the network. Those channels are called micropayment channels. When two participants want to conduct a transaction, they have to open a new micropayment channel between them, if one does not yet exist. Every channel possesses its own wallet in which the channel owners can deposit assets. The inserted amount of assets, and only that amount, will be used in the transactions between participants while using that channel. The wallet keeps track of how many assets each of the participants has deposited. After participants have added their assets, the initial state of the shared wallet is added to the main chain. This transaction will then be broadcasted to the main chain under the name Funding Transaction, which basically tells the main chain that there's a channel with deposited amount of assets between two participants. This is one of the two times that publishing of information to the main chain must occur in the implementation of an off chain solution. The second time is when the users want to terminate their channel. In this way, the last state of the wallet is added to the main chain and that state keeps the precise information regarding how many assets each of the channel owners have after conducting numerous transactions between them. In summary, the first publishing to the main chain occurs when the channel is created and this state tells how much each of the channel owners has deposited in the channel wallet. The second publishing to the main chain occurs when channel is terminated and this state tells how much assets each of the channel owners can withdraw from the channel wallet. After the

creation and before the termination of the channel, owners can perform unlimited number of transaction between them and they do not have to be published to the main chain.

Security mechanisms

Since transactions are not recorded on the main chain, one can think that they are susceptible to frauds. In order to prevent fraudulent transactions, the off chain mechanism has several solutions. Firstly, all transactions that occur in the micropayment channel are signed by both transaction parties, this method is called the 2-of-2 multisignature [10]. In this way, when participants want to close the channel and publish the last state of the wallet to the main chain, where the latest transaction depicts last state of the wallet, it is possible to prove which participant published the transaction. Some participants are prone to publish transactions that do not depict the real state of the wallet, but instead some other transaction that favors a higher amount of assets for them. Therefore this is very important, since participants often do not know each other so there is no mutual trust between them. The significance of the 2-of-2 multisignature is connected with the next mechanism called the penalty mechanism.

The penalty mechanism should turn away all participants who would like to be unfair during a transaction process. This would be of use in a case where one of the channel owners wants to terminate the channel and add a fraudulent transaction to the main chain. For example, in a case of transaction that claims channel owner possesses more assets than what is actually the case, and if the other channel owner finds that out, then the unfair channel owner will be penalized by losing all his assets in the wallet and they will be rewarded to the other micropayment channel participant [10].

Transactions do not have to be conducted directly between participants. It would be overwhelming for the off chain network to possess direct channel between every pair of participants in the network. Let participant A wants to transfer assets to participant C but A and C do not possess a common channel. On the other hand A has an open channel with B and B has an open channel with C, as a result A can transfer assets to C over the existing channel with participant B. In order to have fair transactions over several participants in the network, also called nodes or hops, another security mechanism is used. This mechanism is called Hash Time Lock Contract (HTLC). The main purpose of the Hash Time Lock Contract is that if the sender of assets receives a confirmation, in the form of a pre-image of a hash, that assets have been successfully delivered to the receiver in the agreed time-based period then the transaction becomes valid, otherwise the sender has a right to revoke the transaction and revoke his assets [10]. This mechanism is introduced because nodes can be unfair and try to alter transactions or simply stop participating in the network, thereby not delivering transactions to the receiver. Two networks that implement an off chain solution are the Bitcoin Lightning Network and the Raiden Network.

2.4.1 The Bitcoin Lightning Network

The Bitcoin Lightning Network is a second layer payment protocol that implements an off chain solution. It is called second layer because new concepts, like micropayment channels, are added to the existing Bitcoin network, which represents the first layer. It applies all concepts mentioned in the previous part, opening channels, terminating channels, processing transactions, 2-of-2 multisignature, penalty and HTLC security mechanisms. It supports transactions over several nodes. The transaction fees are very low and they are paid directly to the participants that hold the communication channel [10]. Since an unlimited amount of transactions with low fees can be made between two users in a channel, the idea to use the Bitcoin for paying everyday services, like groceries or a coffee in a restaurant is finally feasible. Those small transactions are also called microtransactions.

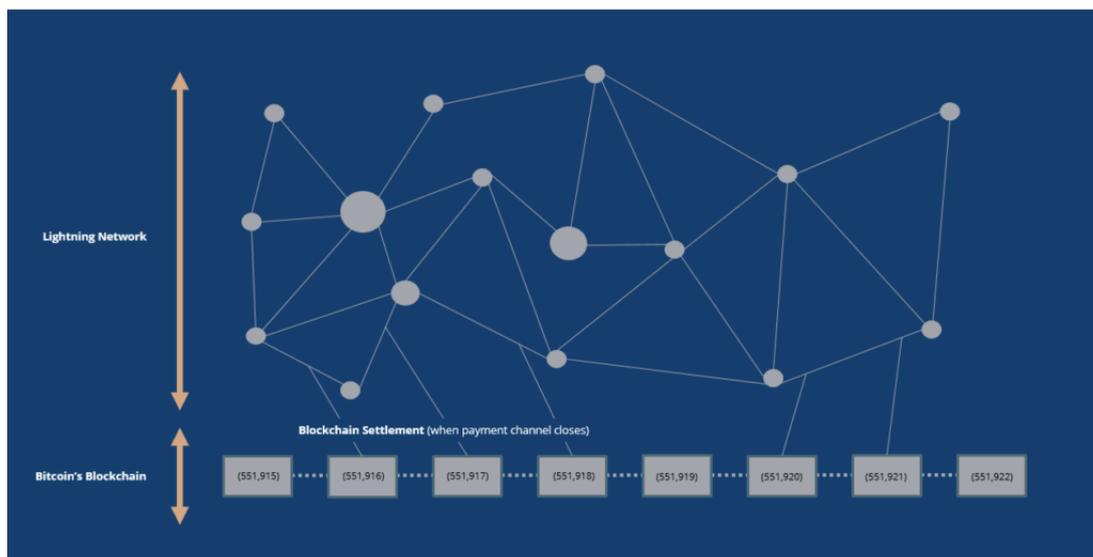


Figure 2.5: The Bitcoin Lightning Network, source: [19]

The feature that is not included in the Bitcoin Lightning Network but is currently in a testing phase is the cross-chain atomic swaps [20]. This feature would allow transferring of assets between different blockchain networks. Having high transaction speed, low transaction fees, scalability and possibly a cross-chain atomic swaps feature, makes the Bitcoin Lightning Network to appear as an in-time solution for blockchain scalability problem. Since it is in operation now, with more than 4 200 nodes and 37 000 channels (data of May 2019), time will tell whether the blockchain scalability problem is successfully solved using the off chain solution.

2.4.2 Raiden Network

As stated in [21] the Raiden Network is another example of network that uses an off chain solution for solving the scalability problem. In comparison to the Bitcoin Lightning Network, Raiden Network is Ethereum based. This means that Raiden Network supports smart contracts, which allows for a broader area of use cases than the Bitcoin Lightning Network, since the Lightning Network is only used for assets transactions. The Smart Contracts can be used restricted to Ethereum.

2.4.2.1 Advantages and Limitations

Obviously, the advantage of an off chain solution is faster transactions, since they occur off the main chain and in the micropayment channels. Transactions are not added and processed on the main chain, which allows diminishing transaction fees. The most important improvement compared to the standard Bitcoin network is that micropayment channels allow for unlimited transactions per second. It is also microtransaction oriented making it more applicable for wide scale implementations.

On the other hand, this solution is not perfect, it deals with big security risks. One of them is an attack called force expiration spam [10]. An adversary performs force expiration spam by opening many micropayment channels with different participants in the network and then closing them all at the same time. When the micropayment channel is terminated, publishing on the main chain must occur. Having a great number of transactions published on the main chain at the same time, which do not provide any useful information since no transactions occurred in those channels and their balance is the same as when the channel was created, results in the blocking of the real transactions to get included in a block on the main chain. This type of attack represents one of the greatest threats

for the Lightning Network. Furthermore, both the Bitcoin Lightning Network and the Raiden Network allow transactions over several nodes in the network and if the nodes are uncooperative, transactions are delayed and not processed. Because of this, off chain solution is not suitable for large transactions, since large amount of assets could be locked up in a channel for a period of time that is agreed in the HTLC [10].

Despite these problems, off chain approach is a promising solution to tackle the scalability problem in public blockchains. Microtransaction orientation of the off chain solution facilitates the use of Bitcoin for day-to-day payments. Other promising use-cases of an off chain solution are instant transactions, exchange arbitrage, cross-chain payment and financial smart contracts and escrow. Both the Bitcoin Lightning Network and the Raiden Network are live now. Thus, time will show if the proponents of the off chain solution were right.

2.5 Side Chain

The main concept of side chain solutions is to exchange data between different blockchains and enable interoperability of different blockchains. As an example, it would be interesting to be able to use Bitcoin in Ethereum smart contracts. The functionalities of Bitcoin are limited to a few operations, making the Bitcoin scripting language a non-turing complete primitive [11]. By using the Ethereum blockchain, transactions can be done in a faster way as Ethereum's throughput is a little bit higher than Bitcoin's. The side chain solutions is an attempt to take the best from both blockchains. Suppose that cryptocurrency A has interesting properties but it is slow, while on the contrary blockchain B is much faster but the related cryptocurrency is less interesting. Then an implementation of a side chain solution, where we combine the benefits of both blockchains, would work like this: On blockchain BC_A , a certain amount of assets is blocked and on blockchain SC_B a transaction with the corresponding value of the blocked asset is created. The blocked assets in BC_A will remain frozen while all the transactions are processed on the side chain, namely SC_B . The transactions on the side chain must respect the condition of SC_B which could be different from the conditions of the main chain. These transactions could be executed in a faster way as SC_B is more scalable than BC_A . Once all the transactions are done, the balance of the transactions is calculated, then this amount will be de-frozen on BC_A , implying that the assets move back from the side chain SC_B to the main chain BC_A . Two side chains solutions, namely Rootstock and Blockstream will be presented in the following sections.

2.5.1 Rootstock

Rootstock is an open-source side chain project, that was proposed in 2014 by Sergio Demain Lerner. It was developed to enable a Bitcoin user to build decentralized applications on top of Bitcoin, to have smart contract functionality with the Bitcoin platform, and allow for increased scalability and instant transactions by using a few novel approaches. Rootstock is a sidechain that is pegged to Bitcoin through a two-way peg and rewards Bitcoin miners through merge-mining, which allows them to actively participate in the "smart-contract revolution" [22]. The two-peg method describes a mechanism in which the coins on the primary chain are locked up, and the corresponding tokens on the side chain are released for use. Rootstock does not compete with Bitcoin as a cryptocurrency, since it does not have its own tokens. Instead, Rootstock has its own pegged coin, called SmartBitcoin (SBTC), that is linked to Bitcoin[23]. Meaning that the value of SBTC is linked to the value of Bitcoin.

A transaction on Rootstock will work as follows: BTC will first be locked on the Bitcoin blockchain (primary chain), after that the corresponding amount will be unlocked on the Rootstock chain (side chain). When transferring from Rootstock to Bitcoin, the SBTC is locked on the side chain and the BTC is unlocked on the primary chain. Since it is impossible to verify the transactions occurring on the side chain, a temporary storing mechanism locks and unlocks BTC on the primary chain to allow for cross-chain transfers. Rootstock uses a hybrid Federated side chain model to ensure a degree of trust in the two-way peg (2WP) architecture. The Federation model consists out of semi-trusted third parties (TTP) with high-security standards, that are reputable companies within the blockchain sector. These semi-trusted TTPs are notaries who act as BTC custodians and obey the network rules [24]. Even though it requires TTPs, the process is highly automated, where the Federation's task is to audit the behavior of the software governing process. Rootstock has the same blockchain immutability as Bitcoin due to the important part that the Federation plays in the two-peg system [23].

The transfer to and from the Bitcoin blockchain to a Secondary blockchain, relies on a 2WP. RSK settled on a "Multi-sig Federation" 2WP that allows bitcoin miners to mine both chains at once without loss in efficiency, known as a merge-mining. The 2WP is implemented by having the Federation in control of a multi-signature, where funds are unlocked with approval from the majority of the Federation. By having a group of TTPs the design works better than having a centralized controller, however there is still some degree of centralization due to the small number of TTPs in the Federation.

2.5.1.1 Rootstock Functionalities

Rootstock introduces smart contract functionality to Bitcoin and positions itself as a scaling solution for the issues Bitcoin is facing. The smart contracts are using the Rootstock Virtual Machine (RVM). Rootstock uses the same programming language as Ethereum, Solidity, thereby making it easy to run Ethereum Virtual Machines on RVM.

Currently Rootstock can handle 100 transactions per second with a block confirmation time of approximately 10 seconds. However, even though Rootstock is a side-chain approach, the project aims to improve scalability through the on-chain implementation of a protocol that they have named the Lumino Transaction Compression Protocol (LTCP). Rootstock will use LTCP as part of their off-chain payment, the choice to implement LTCP is left to the individual users. It will be a trade-off choice for the users since the transaction compression that happens with LTCP, will lead to data reuse and reduced privacy. In combination with LTCP, Rootstock will also increase scalability by using fraud proofs and probabilistic verification of transactions [23].

2.5.1.2 Advantages and Limitations

By using merged mining, Rootstock is able to leverage on the large and distributed public blockchain of Bitcoin, and thereby be able to reach 2000 transactions per second. Rootstock claims that a soft-fork upgrade of the Bitcoin-core protocol would improve scalability of Bitcoin itself up to 100 transactions per second. By using this approach Rootstock simplifies the trilemma by introducing the federated structure, which allows for some centralization.

One of the properties of the scalability trilemma is decentralization. Rootstock plans to grow towards a decentralized solution, first by starting out with a federation of reputable blockchain companies. Once the merge-mining gets a miner acceptance of 95%, then the federation role in voting can and will be disabled. It could be very difficult to achieve decentralization, due to a few things, one of them being that the companies in the federation will get a fee for notarizing the incoming and outgoing payments using Rootstock. There-

fore it could be very likely that Rootstock's path to decentralization can be undermined by the companies in the Federation in order to not lose their payment fees.

Rootstock's smart contract platform, which can also execute Solidity bytecode, allows the seamless migration from the Ethereum blockchain. However, whether or not the costs of change from Ethereum to Rootstock and could opt to stay there, seeing as Ethereum is researching and implementing solutions for its scalability issue.

Rootstock's main implementation would be smart contracts, coupled with scalability opportunities and security provided by the federation model, which are desirable elements to have for a blockchain network. Even though Rootstock has some problems to deal with and might not be able to become fully decentralized, it has an impressive financial backing from blockchain companies such as Coinsilium, Bitmain, Digital Currency Group and more [24]. Furthermore, Rootstock's advancements to Bitcoin would be very promising by allowing the network to scale up to 100 transactions, and allowing Rootstock to scale up to 1000 transactions per second, all while retaining safety through the Federation.

2.5.2 Blockstream

Blockstream is a solution that issues new digital assets by adding new functionality to Bitcoin. A side chain that runs parallel to Bitcoin was implemented. This side chain solution, called Liquid Network, contains new features and capabilities and is especially made to process micropayments. The goal of Liquid Network was to facilitate rapid and secure transfers between accounts. This was motivated by the fact that Bitcoin users experience delays when transferring Bitcoin between different accounts. Therefore, they decided to keep the useful properties of Bitcoin which are the decentralization, innovation and security while improving the scalability. In Liquid, the funds can be moved instantaneously, which would allow to actually begin trading cryptocurrencies. They decided to extend Bitcoin blockchain by adding some private components on the side chain in order to improve upon existing protocols and allow the rapid transfer between accounts, a high transaction volume, and low fees as all the transactions are executed on the side chain [25].

Due to the private components, the Liquid side chain, which is a federated side chain, is considered as centralized. The Liquid Network therefore trades the decentralization of blockchains in order to process transactions faster. On the Liquid blockchain, transactions are not validated using a PoW mechanism, as PoW is the reason causing the current, slow latency. Instead Liquid used a federated mechanism called "Federated Pegs". In particular, Liquid Network implemented a Strong Federation, which is a way to implement a federated side chain, which can be viewed on the Figure 2.6. The main blockchain is included in a federation of other side chains that could be connected with each other.

Two different types of functionaries, responsible to handle transactions are included in federations: the Blocksigners and the Watchmen. The functionaries are hosted by multiple independent companies that decided to be part of Blockstream's project. In this sense, the decentralization is traded as the companies could be considered as trusted parties. A Strong Federation means that the functionaries must be geographically and juridically distributed in order to make the network compromise resistant. This also allows to retain some of the properties of a decentralized model. The distinction between the types of functionaries was made in order to limit the danger if an attack occurs [26]. The distinction between the roles is the primary way to achieve security.

2.5.2.1 Blocksigners

On the side chain, the Blocksigners define the consensus history as they are responsible for signing the blocks that contain the transactions. They are responsible for the transfer

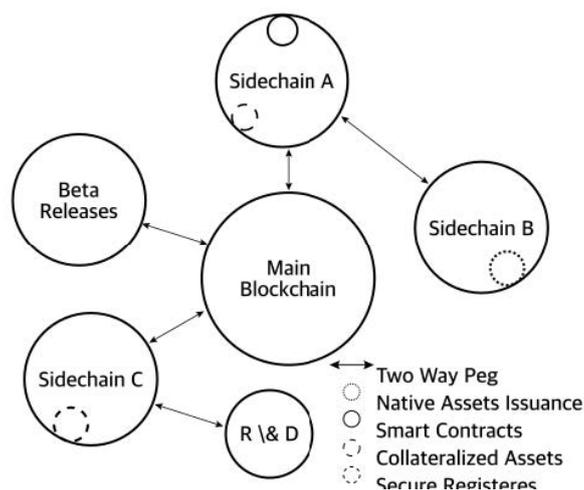


Figure 2.6: Side chain: Strong Federations, source: [26]

of assets from the Bitcoin to the Liquid chain, and for including transactions occurring on the Liquid chain in blocks. The blocks on the Liquid side-chain must be signed by a certain threshold of signers: k of n multisignature scheme, which makes the system byzantine secure. In a Strong Federation, the consensus must only be obtained among the Blocksigners. This is the main difference with the main chain: the side chain does not follow the same rules as the main Bitcoin chain. In Bitcoin PoW is required to validate blocks while in Liquid side chain blocks are validated using multisignatures. In Bitcoin, consensus is needed among all the peers while Liquid only requires consensus among k out of n peers.

2.5.2.2 Watchmen

The Watchmen are responsible for moving the assets from the Liquid side chain to the Bitcoin main chain while the Blocksigners are responsible for the transfer from the main chain to the side chain. So the Watchmen are responsible for signing transactions on the main chain.

2.5.2.3 Advantages and Limitations

Liquid was developed to make transactions more efficient by using Bitcoin as a main chain. Liquid was the first implementation of a Strong Federation, which is a concept that allows to solve the transaction latency. Liquid offers less latency as it has a novel security set-up and different trust assumptions. According to the authors, "because a Strong Federation's block generation is not probabilistic and is based on a fixed set of signers, it can be made to never reorganize. This allows for a significant reduction in the wait time associated with confirming transactions" [26]. Therefore Liquid allows blocks being generated every 1 minute.

As the group of functionaries is small and well defined, this means that the network can be "significantly faster than Bitcoin" [26]. Therefore a confirmation of a transaction can occur as "quickly as information can be broadcast between the federation members and processed into a block" [26]. This distinction between the types of functionaries allows to make the network more secure: in order to fraud the network, it would require to corrupt the Watchmen and the Blocksigners, which are geographically and juridically distributed. In general, this solution does not modify the scalability issues of Bitcoin, but rather by processing the transactions on a side-chain that scales better. Thereby achieving an increased throughput, since transactions could be made instantaneously without latency, and the costs are reduced. For now, the Liquid Network is only compatible with Bitcoin,

but Liquid Network plans to develop their concept for other blockchain solutions. However, a drawback for this solution would be that it contains private components, thereby not making it fully decentralized.

2.6 Child Chain

One of the other developed solutions to blockchain scalability problem is called child chain solution. "The child chain solution has a parent-child structure where the transactions are processed in the child chain and the results are recorded in the parent chain" [27]. Child chain solution explicitly consists of a parent chain that is linked to child chains which are interconnected with each other. Each child chain acts as an ordinary blockchain and may contain other sub child chains. This structure, which can be seen on figure 2.7, shows that each sub chain is connected to the main chain either in a direct or in an indirect way.

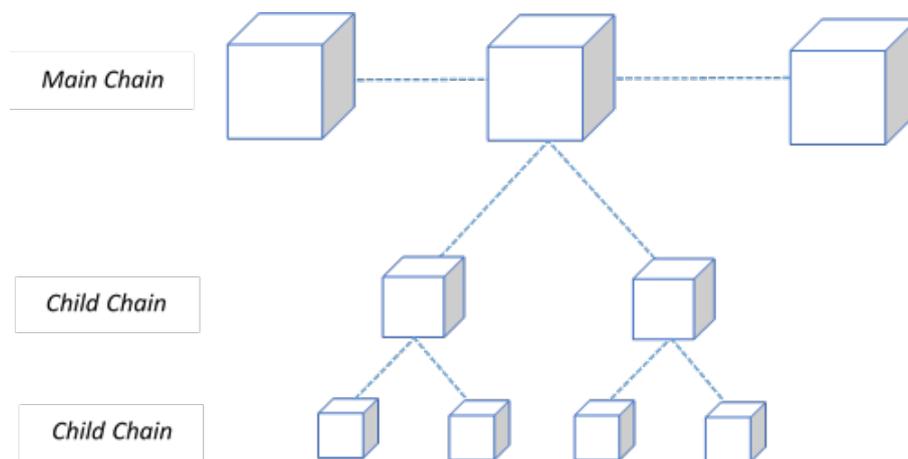


Figure 2.7: Child Chain

Similar to off chain solutions, transactions are not processed on the main chain but rather on the child chains. Transaction works like this: the child chain processes the transaction and puts all the transactions into a block. Then the merkle root hash of the transactions is computed and is pushed to the parent chain.

As stated above, the fundamental difference of this solution is that processing takes place in the child chain, which the scalability problem benefits from the most. "The advantage of this approach is that the process becomes much lighter, since only the child chain's merkle root hash is moved to the parent chain, and the parent chain can divide the transactions to be processed in different child chains" [27]. However, verification process of child chain is troublesome due to the fact that child chain is required to track all parent chains in order to confirm their non-fraudulent status.

2.6.1 Ethereum Plasma

One novel example of child-chain solution is in Ethereum called Plasma. "Plasma is a way to achieve scalable computation on the blockchain with the structure of creating economic incentives to autonomously and persistently operate the chain without active state transition management by the contract creator and the nodes themselves are incentivized to operate the chain" [28].

Plasma consists of five core components which are an incentive layer, the structure allows child chain to form a tree format, a MapReduce framework, root chain dependent consensus mechanism, and a bitmap-UTXO. An incentive layer is designed for continuous processing of contracts in an economic way while the structure allowing child chain to

form in a tree format is to secure low-cost efficiency. MapReduce is a program which consists out of mapping and reducing procedures, thereby constructing fraud proofs of state transitions within the created nested chains. Root chain dependent consensus mechanism exists for attempting to replicate the outcome of Bitcoin consensus incentive and finally a bitmap UTXO is to secure correct state transitions out of the main blockchain in the meantime minimizing the mass-exits costs.

In addition to introducing Plasma, Plasma blockchain is also introduced by the Plasma creators. Plasma blockchain is a unique framework that includes multiparty off-chain connections that can hold state on the behalf of others. This framework explicitly allows deposit and withdrawal of funds into Plasma chain. Thus one can deposit and withdraw with basis of Plasma block matching the funds in the root chain. In this structure, one can execute high amounts of transactions on the Plasma since Plasma prevents high amount of data directed to the root blockchain. Any participant within the system is able to transfer funds freely to anyone, even to users not existing among the current participants. Transfers of these funds are allowed to pay or withdraw funds within the native coins of blockchain root. Plasma creators were inspired by the structure of the current judicial court system, and based their structure on it. Within the Plasma, higher and lower courts exists in order to maximize availability and minimize costs that arise in non-Byzantine states. In the case of a chain being Byzantine, then a chain has the flexibility to move across any parent chain including the root blockchain to resume execution or exit with current status. Plasma structure includes a structure of fraud proofs to ensure state transitions and balances between parent-child structures. The current structure of Plasma chain provides benefits over other solutions because it is light. Plasma allows state transitions that will be created in the system committed not constantly but periodically to the parent child. In this way, Plasma introduces highly scalable structure in terms of computation and account state as it does not transfer all the data, except the raw data to parent or root chain in Byzantine situations.

Plasma structure is constructed with smart contracts and merkle trees in a way that it is possible to create infinite number of -ethereum blockchain- like child chains. Each of these child chains can be expanded with additional child chain, forming a tree structure. With this form of structure, Plasma is highly scalable to process and complete billions of updates per second. It recursively creates a Plasma Chain with a parent-child structure, which processes transactions at the child-chain, then transforming all transactions into a single root hash and pushing it to the parent-chain.

First of all, smart contracts are created on the Ethereum main chain, existing as the root of the Plasma child chain structure. After the process of rooting child-chain in the main chain, the child chain is created in Plasma chain that will become the parent chain. This plasma chain will deposits the coin through smart contracts and will only return completed transactions to the root chain. The created child-chain in the process behaves as an ordinary blockchain structure and transactions processes within the created child-chain are stored in the parent-chain as a merkle root hash.

Thanks to this novel structure, Plasma can perform more complex operations on the child chain without disturbing the main Ethereum Chain. This plasma structures also processes transactions faster with lower fees than the main chain. Due to the fact that plasma chains do not need to be duplicated across the entire Ethereum blockchain, and since transactions are not processed on the main chain.

Finally the hash of this Plasma BC (that includes 2 child chains) is computed and updated on the Root chain (Ethereum). At the end this transaction will be completed. As one can observe the transaction process is done on the child chain and not on the main Ethereum chain [28].

2.6.1.1 Advantages and Limitations

Plasma structure is potentially scalable to billions of updates per second because it is able to handle and process larger data sets. Furthermore, the Plasma structure is very light since Plasma does not push high amounts of data to the main chain, it rather pushes only the merkle root hash to the parent chain. Therefore, the parent chain will not be busy with unnecessary data. In addition, Plasma has a very low transaction cost, within Plasma structure, there is no need to transfer all the information to all participants of Ethereum Chain. Finally, Plasma can run more computer intensive applications because it will not run on the main chain, rather it will run on the child chains which act like main chain and execute the computations.

As Plasma has many advantages, it has also downsides. Fundamentally, Plasma exits may cause troublesome situations because there is a possibility of overloading on the main chain. Plasma chains allow plasma exits anytime therefore, it is possible to face with high amount of exits from the chain to the main chain at once. Plasma is currently at the development stage. There may be possible drawbacks yet not discovered during experimentation and they might be revealed at further stage of development and release [28].

2.7 Scalable Consensus Mechanisms

So far we have only spoken about techniques that change the block size, protocol or chain structure where the transactions are processed. However, a solution that changes the consensus mechanism, which can greatly increase scalability, hasn't been explored yet in our report. In this chapter we will have a look into different consensus mechanisms that promise to replace current most utilized consensus mechanisms PoW and PoS. Consensus mechanisms Delegated Proof-of-Stake and Byzantine Fault Tolerance are based on different ideas, however they all engage to increase scalability [30].

2.7.1 Delegated Proof-of-Stake

Delegated Proof-of-Stake (DPOS) [32] is a consensus mechanism that solves the "rich getting richer problem" of Proof-of-Stake (PoS) mechanism. There are many implementations of this algorithm but they all follow the same idea. In this consensus mechanisms stakeholders, nodes that represent active users, vote for witnesses and delegates. Witnesses have responsibility to create and validate blocks in the network. On the other hand, delegates have the job to watch over the network and propose changes in the network such as: witnesses fees, block size or block intervals. Those changes are then presented to the other users in the network, which is followed by voting for those changes. In most implementations, delegates do not get any incentives for their work, whereas witnesses receive fees for creating blocks. Since re-voting of witnesses occurs in a predetermined amount of time (e.g., 1 hour or 1 day), witnesses are in constant fear of being replaced by other users. This actually represents the main incentive for them to be fair in the network and in that way keep their popularity high. In many implementations, votes are weighted by the amount of assets that the user possess. The number of witnesses and delegates depends on the implementation. The reason why this consensus mechanism can solve scalability problem in the blockchain is that no miners are necessary since no complex cryptographic puzzles have to be solved. As a result, blocks can be created every several seconds, in most implementations it is 3 seconds, resulting in high throughput of the network.

One of many blockchain examples that use Delegated Proof-of-Stake consensus mechanism is Steemit.

2.7.1.1 Steemit

Steemit [31] is a blockchain social media platform. On the contrary to the Bitcoin where rewards are granted for creating blocks, in Steemit users that publish content on the platform are rewarded for their effort. Furthermore, upvoting content and commenting also brings assets to the users. There are 3 types of tokens in Steemit blockchain: Steem, Steem Power and Steem Dollars. Steem represents assets that can be bought and sold on open market, like BTC or ETH. Steem Power represents assets that grants user the ownership of the network. The more Steem Power user has, the more assets he/she gets on daily basis since the current creation of coins works as follows: every day certain amount of Steems is created and 90% of that amount is distributed to the holders of Steem Power, the rest is distributed to the creators, voters and commentators of the content. Steem Power can not be sold for 2 year from the time they are obtained. In this way, the price of cryptocurrency keeps stable because if sometimes fluctuations in price occur and stakeholders driven by their fear would like to get rid off their assets (and mass selling of cryptocurrencies would crash its price) they would not be allowed to do that. Steem Dollars also represents assets that are granted for users in the network. For their participation, posting, voting or commenting users get 50% of reward in Steem Dollars. They can be sold at any time. Since we speak about blockchain implementations, naturally, these tokens can be used for transactions. Steemit relies on Delegated Proof-of-Stake and Graphene technology so it is theoretically capable to handle up to 100 000 transaction per second [31]. In practice that number is much lower, around 24 transactions per second.

2.7.1.2 Advantages and Limitations

Delegated Proof-of-Stake (DPOS) has many advantages over PoW and PoS. First of all, it does not require its users to have high computing power resulting in high energy efficiency and speed. Furthermore, this consensus mechanism allows every participant in the network to vote and because of this it is considered as the most democratic and decentralized consensus mechanism. This also solves the "rich get richer problem" in PoS because the determining factor for someone to be elected as witnesses is his/her reputation rather than amount of assets possessed [32].

On the other hand, if users in the network are not interested in participation and voting fair witnesses then the whole concept of DPOS has no purpose. Also there is a fear, like in every democratic society, that the users will start organizing in cartels, like pools in PoW. This will lead that same witnesses are re-elected over and over again which can lead to less decentralized and resilient network [32].

So far, with raising number of implementations and users, DPOS looks like as a promising consensus mechanism for scalability problem in blockchain.

2.7.2 Byzantine Fault Tolerance

Blockchains are decentralized ledgers which are not controlled by any central authority. Due to their structure and the values within them, blockchains are prone to include bad actors who may seek economic incentives to try and be a source of faults in the system. In this case, any actor can produce false transactions within the system and highly decrease the reliability of the blockchain. With this scenario in mind, a need for a mechanism that creates constant consensus between different distributed systems, thereby avoiding Byzantine Fault in which components of the overall system fails and depicts various outcomes or false symptoms arises. Byzantine Fault Tolerance therefore comes forward as a unique solution to tackle Byzantine Fault. Byzantine Fault Tolerance is a replication algorithm that can be used to tolerate Byzantine Faults within the systems.

Byzantine Fault Tolerance formal definition within the literature follows: *"BFT is the first Byzantine-fault-tolerant, state machine replication algorithm that is safe in asynchronous systems such as the Internet: it does not rely on any synchrony assumption to provide safety"* [33]. The fundamental difference that Byzantine Fault Tolerance provides is a constant consensus among peers within distributed systems, in spite of the existence of bad actors within the network who seek to harm the system. Byzantine Fault Tolerance not only protects the whole network from failures but also maintains liveness at most $(n-1)/3$ out of total n faulty replicas, meaning that the actor is able to obtain the correct response to the request that posted. The system service might not be able to process replies during DoS attack, but it is capable of providing the response to the actor when the attack ends [33].

The BFT algorithm is designed as a replicated state machine among the nodes within the distributed system. Every replica in the system includes the service state and executes the service operations. BFT makes sure that all correct replicas process the operations in the same order when actors provide a request to execute operations for replicas. Due to the deterministic nature of replicas, they start in the same state. Replies with identical results for each operation are sent by all non-faulty replicas. Later, an actor holds until obtaining $f+1$ replies with the same outcome from various replicas. As more than one non-faulty replica exists, the correct outcome of the operation is obtained. Finally BFT ensures an operating system, as long as there are no more than one third faulty replicas of the whole system. Therefore, the algorithm tolerates malicious actors with certain capacity to be able to maintain network security and usability [33].

2.7.2.1 Practical Byzantine Fault Tolerance Mechanism

However, the bound of faulty nodes in the network introduced in BFT is not enough for the systems that have long existence because this bound is prone to be exceeded. Therefore, it is worth it to discuss the improved version of Byzantine Fault Tolerance algorithm, Practical Byzantine Fault Tolerance mechanism. Practical Byzantine Fault Tolerance is a high performing variation of Byzantine Fault Tolerance. Practical Byzantine Fault Tolerance has a recovery mechanism within itself to be able to process faulty replicas to behave correctly in a network. It is observed that faulty replicas may behave correctly even though they are the faulty ones. Creators of Practical Byzantine Fault Tolerance designed the recovery mechanism to function proactively in order to avoid bad actors within the network that could disturb the overall system by damaging the one third of the replicas without detection. The designed recovery mechanism functions as a mechanism that recovers replicas even if there is no specific point to suspect that they are faulty [33].

2.7.2.2 Zilliqa

Zilliqa is one of the projects that benefits from the Practical Byzantine Fault Tolerance consensus mechanism. It was designed and created in order to increase the amount of transactions. It is a scalable blockchain platform that can execute thousands of transactions per second. Because its structure allows it to increase the transaction throughput at a linear rate as participating nodes within the network increase. Therefore, Zilliqa, with its design, maintains decentralization because it does not limit the number of nodes that can participate in the system. Founders of the Zilliqa solution chose BFT for their consensus protocol design to make sure that the resulting blocks are definitive, without the requirement of long confirmation times as required in the popular "longest-chain" rule in existing cryptocurrencies [34].

Zilliqa is a scalable blockchain platform that can execute thousands of transactions per second. Because its structure allows it to increase transaction throughput at a linear rate as

participating nodes within network increases. Therefore, Zilliqa, with its design, maintains decentralization because it does not limit number of nodes that can participate the system. Additionally, Zilliqa significantly decreases energy usage. The main difference between Bitcoin and Zilliqa, is that while Bitcoin uses PoW to mine every block, Zilliqa uses PoW to create above-mentioned mining entities. Lastly, Zilliqa consensus structure based on PBFT provides finality of transactions. The advantage of finality also allows Zilliqa to be very efficient regarding storage requirements since it does not keep the whole transaction history but latest one.

2.7.2.3 Advantages and Limitations

First of all, Practical Byzantine Fault Tolerance mechanism is extremely energy efficient. Since this structure mechanism is able to achieve distributed consensus without executing complex mathematical computations. Secondly, within this structure each and every node participates response in process to the request sent by actor. In this way, each node can benefit from shared incentivize status. Therefore, there is low reward variance among nodes involved in the execution process. Lastly, a fundamental advantage that Practical Byzantine Fault Tolerance provides is transaction finality. In the case of PoW mechanism in Bitcoin, each node is required to verify all of the transactions in order to be able to add a new block on the blockchain. This process may take quite long and is dependent on the amount of approving entities. On the contrary, Practical Byzantine Fault Tolerance mechanism does not require such a lengthy process, it rather requires that confirmation completed by nodes in the system to add the proposed block. With this the confirmation proposed block is final.

As it will be discussed later, PBFT has downsized in terms of efficiency of larger network sizes and Zilliqa utilizes novel approaches on its consensus mechanism to achieve higher scalability levels. To tackle the downsizes within Practical Byzantine Fault Tolerance and thus improve efficiency, Zilliqa benefits from digital signatures rather than Method Authentication Codes. Additionally, it introduces EC-Schnorr multisignatures to collect multiple signatures into bigger sized one. Zilliqa is also embedded with the combination of PoW and sharding into its Practical Byzantine Fault Tolerance consensus mechanism [35].

Although Practical Byzantine Fault Tolerance mechanism has many advantages compared to other mechanisms, it has limitations that are worthy to discuss. First of all, Practical Byzantine Fault Tolerance mechanism is prone to face with sybil attacks. A sybil attack is an attack where a network or system is disturbed by a node in a network that operates multiple identities and sabotages the network power. A harmful threat of sybil attacks might be minimized with larger networks sizes. The mechanism must be combined with another consensus mechanism due to the fact that scalability and throughput capacity will be reduced with a larger network size. Secondly, Practical Byzantine Fault Tolerance mechanism is not efficient in large networks. Because each node in the system is required to communicate to maintain security of the network. Although the proposed system uses Method Authentication Codes in order to eliminate performance issues, MACs are not capable of being in an efficient format that can be used for authenticating messages within large consensus groups.

2.8 Conclusion

This paper discussed various approaches to mitigate scalability issues in blockchains. On chain solution modify the main chain, its block size and protocol. RapidChain ([12]) and SegWit ([18]) change the protocol as on chain technique to achieve scalability. RapidChain

applies sharding, whereas SegWit creates a new structure to handle the size of the signatures in a block. Bitcoin NG ([15]) changes the types of blocks in the chain to improve scalability. Next, we discussed off chain solutions, meaning that they improve scalability through processing transactions off the main chain. This solution is particularly suitable for microtransactions. The Bitcoin Lightning Network ([10]) and Raiden Network ([21]) are implementations of such off chain approaches. Side chain solutions have the main idea to provide faster exchange of assets between different blockchains, such as Rootstock ([24]) or Blockstream ([25]). The next solution that changes the core operational principle of the blockchain is the child chain solution. As seen before, the main idea in child chain solution is to have parent-child structure. Transactions occur in the child chain and results are recored in the parent chain. This follows a very similar logic to the off chain solution. A project utilizing child chains is Ethereum Plasma ([28]). Last but not least, we mentioned another approach to fix scalability problem in the blockchain. It represents changes of the consensus mechanisms that the blockchain use. Some of the examples that we presented, like Delegated Proof of Stake, which is used in Steemit blockchain social media platform ([31]), and Byzantine Fault Tolerance, with its example Zilliqa ([34]) show that there are huge interest in moving in this direction for solving scalability problem of blockchain.

This paper has not focused on other types of distributed ledgers but it is necessary to mention that blockchain is only one type of distributed ledger among others. Therefore, it is also possible to adapt the structure of the blockchain and change it to another type of ledger that maintains more or less the same characteristics. As an example, Directed Acyclic Graph (DAG) has a linear data structure where the transactions remain independent, this would theoretically allow an infinite number of transactions. Using Directed Acyclic Graphs, some novel solutions have been developed such as IOTA, an open-source distributed ledger or Nano. In the recent years, the rise of blockchain technologies made it clear that there is a need for decentralized and permissionless systems [36].

The core idea of IOTA was to solve the blockchain inefficiencies to allow high scalability of transactions. IOTA wants to work with Internet of Things (IOT) and must therefore be able to process microtransactions quickly and without fees. IOTA introduced a new ledger type: the Tangle. The idea behind the Tangle is that transactions form a stream, and these transactions are linked together in a network structure. This is the main difference with blockchains: transactions are not grouped in blocks anymore. This Tangle structure being per se lighter combined with a user only validating two transactions using PoW in order to participate in the network makes the solution more scalable [36]. However, in practice it has been shown that this solution is not as promising as advertised, for example double spending issues arise. Furthermore DAGs are not a perfect alternative to blockchains, since there are entities within the Tangle which serve as a central control entities, leading to centralization.

Although our paper discussed many of the techniques that are currently being researched and implemented, it is not difficult to see that many of these implementations are flawed in their own way. The presented solutions could be viewed as a "band-aid" for the bigger problem at hand. Blockchain development was slowed down and received an immense pressure to develop and deliver scalability by the mainstream interest since 2017. Which makes one wonder if any of the solutions discussed could actually prevail, it might be needed that some combination of techniques has to emerge in the future in order to be completely capable of solving the problem. Another view of the situation is that a combination of techniques would also not suffice and a complete reinvention of the structure has to be made in order to be scalable. This could lead to newer platforms, that do not have to deal with an outdated idea to be quick risers and dominate the market in a few years.

A further point to consider is that scalability is not the only one of the challenges that blockchains is facing. There are still some other issues regarding privacy, security, and the mining protocol. Even if blockchain transactions are considered anonymous, it's possible to back track and find which user carried out the transactions. Thereby stripping away anonymity, which is one of the features that users find appealing from blockchain. Another prevalent issue that blockchain technology faces is selfish mining, where groups of miners are able to collude and with the goal of increasing their revenue [1].

This paper largely focused on blockchain as a transaction platform, with the goal of becoming as big as Visa. With that dream in mind, scalability is an issue. However, there are other, useful, applications for blockchain in which scalability might not be so much of an issue. For example small business who want to keep a distributed ledger in the form of blockchain on hand to track changes made, production cycles, etc. In cases like those, the blockchain is not public, and nodes can be trusted, therefore it is easier to adapt the protocol in such a way to make the network extremely fast and scalable. Using blockchain in small to medium sized businesses, in a private network, does not make it necessary to be able to reach 1000 transactions per second, as the number of nodes on the network will be extremely small and fewer transactions will occur than in a public network. Another example of blockchain being used outside of a transaction platform is for diamond verification using Everledger. Diamond verification is a very niche application of blockchain technology and in this case scalability is not considered, since there are not hundreds of diamonds being verified each second. From these two examples it becomes evident that blockchain is mostly seen as a transaction platform, however it has many useful cases besides transactions, where scalability is not an issue.

Bibliography

- [1] Z. Zheng, S. Xie, H. Dai: *An Overview of Blockchain Technology, Architecture, Consensus and Future Trends*, IEEE 6th International Congress on Big Data, 2017, pp. 557-564
- [2] Ripple: *XRP The Digital Asset for Payments*, <https://ripple.com/xrp/>, last visit May 3rd 2019
- [3] A. B. Bondi.: *Characteristics of scalability and their impact on performance*. Proceedings of the second international workshop on Software and performance-WOSP, 2000, p. 195.
- [4] T. Bocek, B. Stiller: *Smart Contracts - Blockchains in the Wings*, in C. Linnhoff-Popien, R. Schneider, M. Zaddach (Edts) "Digital Marketplaces Unleashed", Springer, Berlin, Germany, 2018, pp. 169-184
- [5] R. Dennis, J. P. Disso: *An Analysis into the Scalability of Bitcoin and Ethereum*, 4rd International Congress on Information and Communication Technology, Singapore, Singapore, February 2019, pp. 619-627
- [6] S. Kim, Y. Kwon, S. Cho: *A Survey of Scalability Solutions on Blockchain*, International Conference on Information and Communication Technology Convergence (ICTC 2018), Jeju Island, Korea, October 2018, pp.1204-1207
- [7] BitInfoCharts : *Bitcoin (BTC) statistics*, <https://bitinfocharts.com/bitcoin/>, last visit April 5th 2019
- [8] BitInfoCharts: *Ethereum /Ether (ETH) statistics*, <https://bitinfocharts.com/ethereum/>, last visit April 5th 2019
- [9] Everledger: *Real-world solutions for all industries*, <https://www.everledger.io/industry-applications>, last visit April 7th 2019
- [10] J. Poon, T. Dryja: *The Bitcoin Lightning Network: Scalable Off-chain Instant Payments*, 2016, <https://lightning.network/lightning-network-paper.pdf>, last visit April 14th 2019
- [11] Bitcoin Wiki : *Script*, <https://en.bitcoin.it/wiki/Script>, last visit May 25th 2019
- [12] M. Zamani, M. Movahedi, M. Raykova: *RapidChain: Scaling Blockchain via Full Sharding*, ACM Conference on Computer and Communications Security (CCS 2018), Toronto, Canada, October 2018, pp. 931-948.
- [13] Siddhartha Sen and Michael J. Freedman. *Commensal cuckoo: secure group partitioning for large-scale services*, ACM SIGOPS Operating Systems Review 46, 1 (2012), 33-39.

- [14] Bitnodes: *Global Bitcoin Nodes Distribution*, <https://bitnodes.earn.com>, last visit April 15th 2019
- [15] I. Eyal, A. Efe Gencer, E. Gün Sirer, R. van Renesse: *Bitcoin-NG: A Scalable Blockchain Protocol*, 2016, <https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf>, last visit March 18th 2019
- [16] S. Nakamoto : *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008, <https://bitcoin.org/bitcoin.pdf?>, last visit May 6th 2019
- [17] J. Yin, C. Wang, Z. Zhang and J. Li : *Revisiting the Incentive Mechanism of Bitcoin NG*, 2018
- [18] L.Eric, L.Johnson, and W.Pieter: *Segregated Witness* , <http://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>, last visit March 29th 2019
- [19] Crypto Finance: *Bitcoin's Lightning Network & Innovative Business Models*, <https://www.cryptofinance.ch/en/bitcoins-lightning-network>, last visit April 14th 2019
- [20] J. Garza: *The Lightning Network Cross-Chain Exchange: A Decentralized Approach for Peer to Peer Exchange Across Blockchains*, 2017, <https://dspace.mit.edu/bitstream/handle/1721.1/119736/1078689077-MIT.pdf?sequence=1>, last visit April 15th 2019
- [21] Raiden Network : *What is the Raiden Network?*, <https://raiden.network/101.html>, last visit April 15th 2019
- [22] Hackernoon : *Project Spotlight: Rootstock*, <https://hackernoon.com/project-spotlight-rootstock-6bc1144b835b>, last visit April 15th 2019
- [23] Blockonomi : *What is Rootstock*, <https://blockonomi.com/what-is-rootstock/>, last visit April 15th 2019
- [24] RSK : *Rootstock*, <https://www.rsk.co/>, last visit April 15th 2019
- [25] Blockstream: *Introducing Liquid: Bitcoin's First Production Sidechain*, <https://blockstream.com/2015/10/12/zh-introducing-liquid/>, last visit April 13th 2019
- [26] J. Dille, A. Poelstra , J. Wilkins: *Strong Federations: An Interoperable Blockchain Solution to Centralized Third Party Risks*, <https://blockstream.com/sidechains.pdf>, last visit April 12th 2019
- [27] S. Kim, Y.Kwon, S. Cho: *A Survey of Scalability Solutions on Blockchain* , 2018, pp. 1204-1207. 10.1109/ICTC.2018.8539529.
- [28] J.Poon, V.Buterin: *Plasma: Scalable Autonomous Smart Contracts*,<https://plasma.io/plasma.pdf>, 2017.
- [29] S. Bano, M. Al-Bassam, G. Danezis: *The Road to Scalable Blockchain Designs*, Winter 2017, <https://www.usenix.org/publications/login/winter2017/bano>, last visit March 18th 2019
- [30] Master The Crypto: *Blockchain Scalability Solutions: Overview of Crypto Scaling Solutions*, <https://masterthecrypto.com/blockchain-scalability-solution>, last visit April 13th 2019

- [31] Steem: *Steem: An incentivized, blockchain-based, public content platform*, <https://steem.com/>, last visit April 14th 2019
- [32] Bitcoin Wiki: *DPOS*, <https://en.bitcoinwiki.org/wiki/DPoS>, last visit April 13th 2019
- [33] M.F.Castro, B.Liskov: *Practical Byzantine Fault Tolerance and Proactive Recovery* , ACM Trans. Comput. Syst. 20, 2012, pp.398-461.
- [34] The Zilliqa Team: *The Zilliqa Project: A Secure, Scalable Blockchain Platform*, <https://docs.zilliqa.com/positionpaper.pdf>, 2018.
- [35] The Zilliqa Team: *The ZILLIQA Technical Whitepaper*, <https://docs.zilliqa.com/whitepaper.pdf>, 2017.
- [36] IOTA: *What is IOTA ?*, <https://www.iota.org/get-started/what-is-iota>, last visit April 16th 2019

Chapter 3

An Analytical Study of Enterprise Resource Planning (ERP) Systems in the Supply Chain Industry

Hülya Hüsler, Moritz Wittwer, Ramon Huber

This paper gives an overview of the Enterprise Resource Planning (ERP) Systems, its evolution since the 1970s, the changes in its architecture, and its application in Industry 4.0. It analyses the products of the leading ERP System providers and compares the proprietary and open source ERP Systems in the market. It shows how the functionalities and possibilities with the new technologies in Industry 4.0 change the way of production and deployed systems. The use of IoT devices and how they improve ERP systems regarding supply chain planning and manufacturing is explored. The current state of ERP systems regarding IoT devices is analysed and an example of retrofitting factories is discussed. It also briefly introduces existing IoT applications in SAP. In the next section, the blockchain technology is briefly introduced, followed by the potential use of the blockchain in ERP Systems. A concrete concept is given that shows how an ERP System that uses blockchain could appear and then some requirements and pitfalls are discussed. The potential that the new system has is depicted. At last some examples of companies that already use blockchain in their ERP Systems (mainly for supply chain monitoring) are shown.

Contents

3.1	Introduction	71
3.2	ERP Systems	71
3.2.1	What Is An ERP System?	71
3.2.2	Evolution Of ERP Systems	72
3.2.3	Evolution Of Architecture	73
3.2.4	ERP System Architecture	73
3.2.5	Industry 4.0	74
3.2.6	Proprietary And Open Source ERP Systems	76
3.2.7	Comparison Of Leading Proprietary ERP System Software	78
3.2.8	Challenges And Opportunities	79
3.3	IOT in ERP	80
3.3.1	IoT Introduction	80
3.3.2	Application Of IoT In Manufacturing	81
3.3.3	Readiness Of ERP Systems For IoT	82
3.3.4	Integrating Supply Chain Planning	83
3.3.5	RFID In Production Management	83
3.3.6	IoT Today's ERP's	84
3.4	Integration Of Blockchain And ERP	86
3.4.1	Blockchain Technology	86
3.4.2	Applications Of Blockchain Technology	86
3.4.3	Concept For The Integration Of Blockchain Into An ERP System	87
3.4.4	Examples Of Blockchain In Operation In ERP / Supply Chain	93
3.5	Conclusion	95

3.1 Introduction

The era of big data, Internet of Things (IoT), and Blockchain play an important role in the evolution of Enterprise Resource Planning (ERP) Systems. The aim of this paper is to give an overview about the ERP Systems, its architecture, its different types, and its application in Industry 4.0. Additionally, it analyses how ERP Systems are related to IoT and Blockchain technology in two separate sections. After a short introduction of IoT devices the paper explores how they interact with ERP systems. Existing, as well as future concepts including IoT in ERP systems are discussed with a focus on supply chain management and manufacturing. The blockchain technology is briefly introduced, its potentials and pitfalls in the use in ERP systems are shown, and a concrete concept for the successful implementation of a blockchain-ERP system is given.

3.2 ERP Systems

3.2.1 What Is An ERP System?

ERP is the abbreviation of Enterprise Resource Planning and was coined by Gartner Group in the 1990s [1]. It is the integration of organisations departments and functions in a single computer system. ERP manages, stores, and traces resources. It leads to an operation efficiency and tremendous cost savings. In fact ERP ensures that all relevant and exact needed amount of material for a business or its production is available in the right place and at the right time. For example, a supermarket has to balance demand and inventory by ensuring that the shelves are not empty, but also perishable goods are not spoiled. In this case the ERP system can help in optimizing the order process.

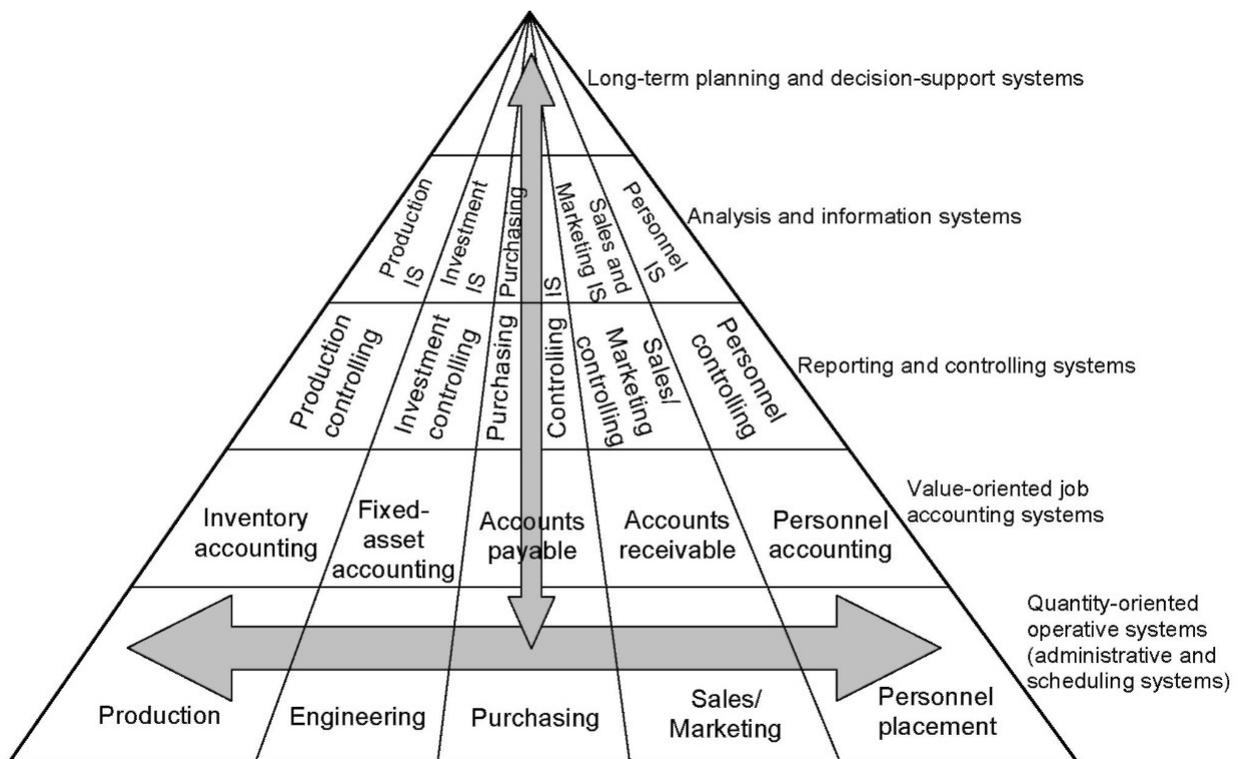


Figure 3.1: Integrated Information System [1]

The Fig. 3.1 shows different levels and departments of an enterprise. In the quantity-oriented level, the ERP System is collecting all information, and transactions that are

happening, such as the number of tyres in production. Then the respective values are mapped in a value-oriented level. The information in those two levels can be aggregated automatically without the adoption of an ERP System. Utilizing the ERP System the aggregation to the next top levels can be accessed. Such that the reporting and controlling level, analysis and information systems and strategic level get the aggregated information in real time [2]. The arrows are showing the vertical and horizontal integration of the different levels and departments.

The aims of an ERP system are Data Integration, Function Integration, Process Integration and Program Integration [3].

Data Integration: The state of data should be unique.

Function Integration: With the integration of data the integration of function is possible. This eliminates redundant work or combines different duties, allowing those duties to now be handled by a single person.

Process Integration: Once having the overview of all the duties and order it is possible to optimise the process.

Program Integration: Integration of all the programs such that there is only one user interface for all the programs working behind that system.

3.2.2 Evolution Of ERP Systems

A kind of ERP System was in use since the 1970s in the form of material requirement planning (MRP) and in the 1980's as manufacturing resource planning (MRPII). In 1990s ERP systems were even able to integrate all departments and levels. Later in the 2000s the supply Chain Management (SCM) is even integrating external components, like suppliers or vendors [4]. (Fig. 3.2) With the breakthrough of the digitalization era, the life cycle of the traditional ERP Systems has almost currently ceased in the existing form and shifted to a new dimension; ERP System in Industry 4.0.

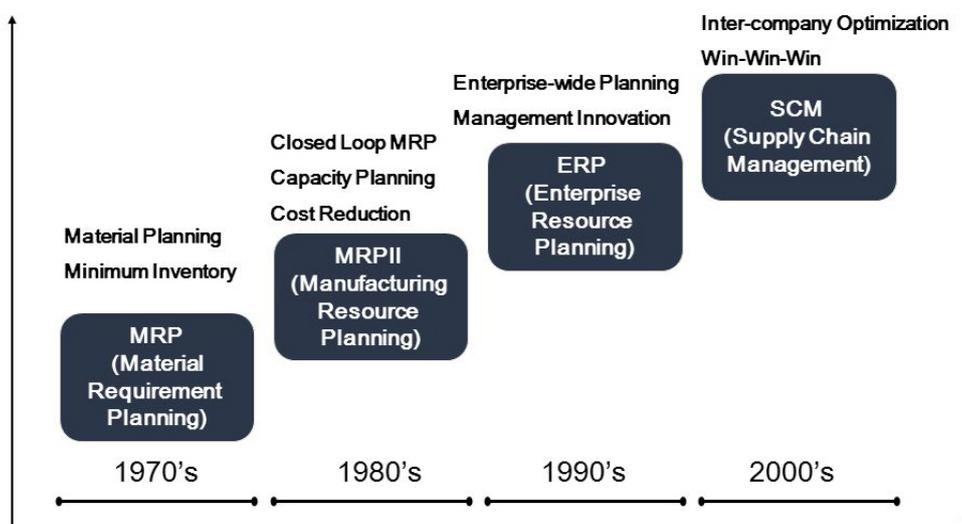


Figure 3.2: Evolution of ERP Systems [4]

3.2.3 Evolution Of Architecture

Fig.3.3 illustrates the three broad evolution stages from the architecture perspective, which are Mainframe Systems, Client-Server Architecture and Service Oriented Architecture.

3.2.3.1 Mainframe Systems

In the 1960s/1970s the hardware usually consisted of mainframe computers [5]. The disadvantage was the scalability, making only a limited amount of users and operations possible at the same time.

3.2.3.2 Client Server Architecture

In 1980s it was possible to use 3-tier architecture by separating the presentation, application and data layers [5]. The different layers could run independently which led to greater scalability and flexibility. According to the business's requirements, it was possible to increase the number of servers, increasing number of PC's able to connect via the Internet.

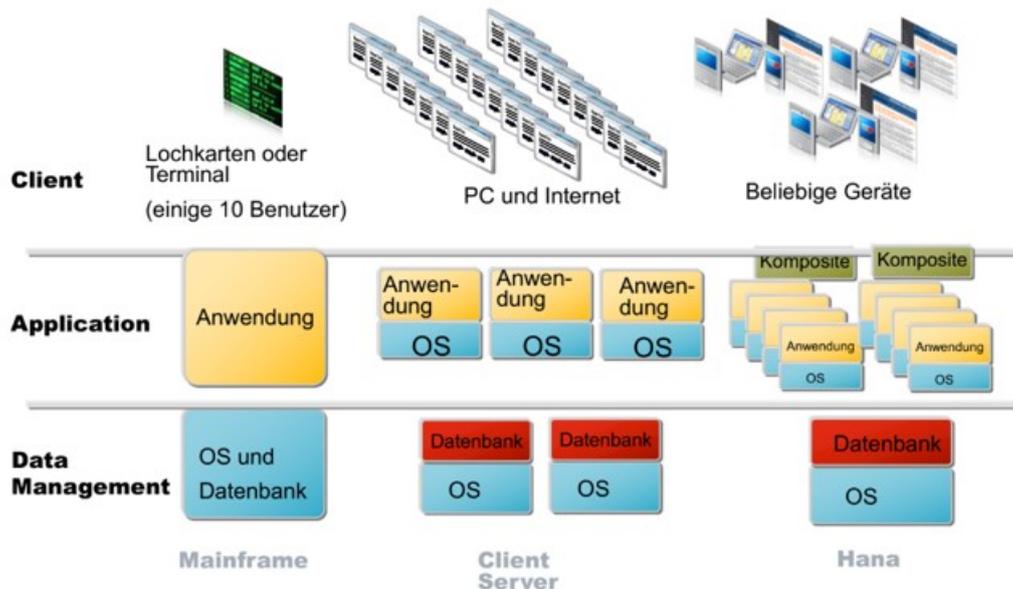


Figure 3.3: Evolution of Enterprise System Architecture [5]

3.2.3.3 Service Oriented Architecture

Later in the 2000s the usage of new technologies helped to integrate or link many different client-server systems together. Companies were able to use the so called Service-Oriented Architecture by connecting to infinite number of composite applications [5]. (Fig.3.3)

3.2.4 ERP System Architecture

In the era of Industry 4.0 the architectures need to be adjusted. In Fig.3.4 is one example with cloud deployment in supply chain management in Industry 4.0.

3.2.4.1 Architecture In Supply Chain Management In Industry 4.0

In Fig. 3.4 the architecture is based on cloud solution. We can observe here two separate clouds. The one on the top is serving Product Life Cycle Management (PLM), Manufacturing execution System (MES) and ERP [6]. The other one, called the Fog/Edge cloud,

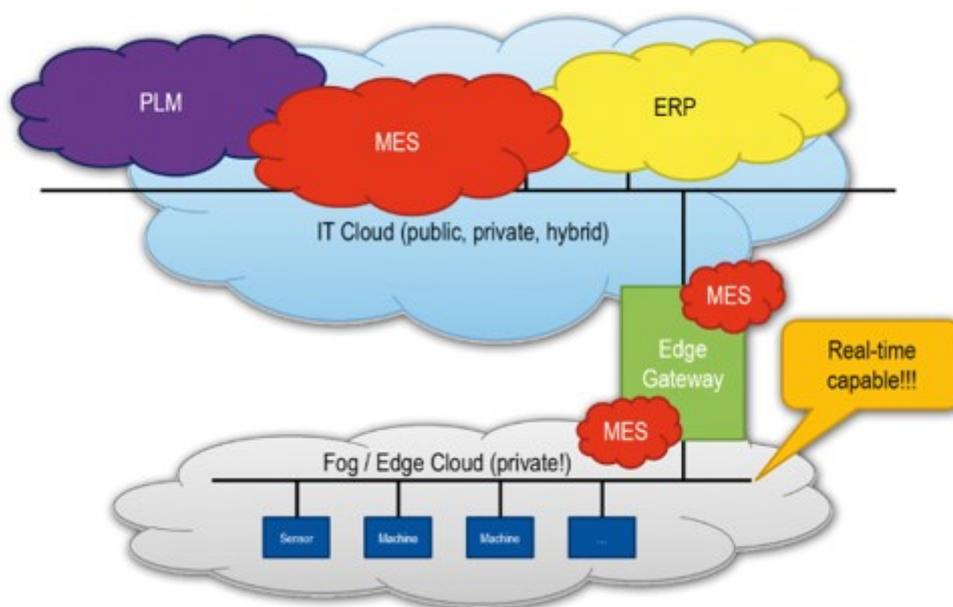


Figure 3.4: Architecture in Supply Chain Management in Industry 4.0 [6]

is used to optimize the speed in collecting and analysing data of the IoT. For example, the communication of the machines should not be slowed down or even put to standby due to the other concurring occurrences such as video conferencing. Instead, a real time data exchange is provided by separating the clouds. Therefore, the edge gateway plays an important role in coordinating the flow of information, also in regards to security.

3.2.5 Industry 4.0

"The term 'Industry 4.0' was coined to mark the fourth industrial revolution, a new paradigm enabled by the introduction of the Internet of Things (IoT) into the production and manufacturing environment" [8]. "Industry 4.0 aims for the full automation of the life cycle of industry products. This operation starts with the production of the raw material, continues with the production process, and ends with the recycling process" [7]. Fig.3.5 depicts an overview of Industry 4.0 with all its components of the value added chain.

The machines are equipped with sensors and complete information about their efficiency or failures can be identified in real time. IoT in industry 4.0 the so called the industry IoT needs to be robust as they can be used in robust environment such as in factories. Through automation, the production with linked equipment in smart factories is possible without human help. The inventory level or stock information is always available. The enterprise is connected to the provider. Therefore, all information can be easily exchanged even with external partners. This information leads to enormous data streams. The middleware takes an important role in collecting, analysing, developing and providing all fields with necessary information. Without the new technology, called In Memory Database, this would not be possible to handle today. In-Memory technology processes data in In-Memory base. Older systems are based on disk storage and relational databases using SQL query. This does not meet today's standards concerning real time process. Whereas

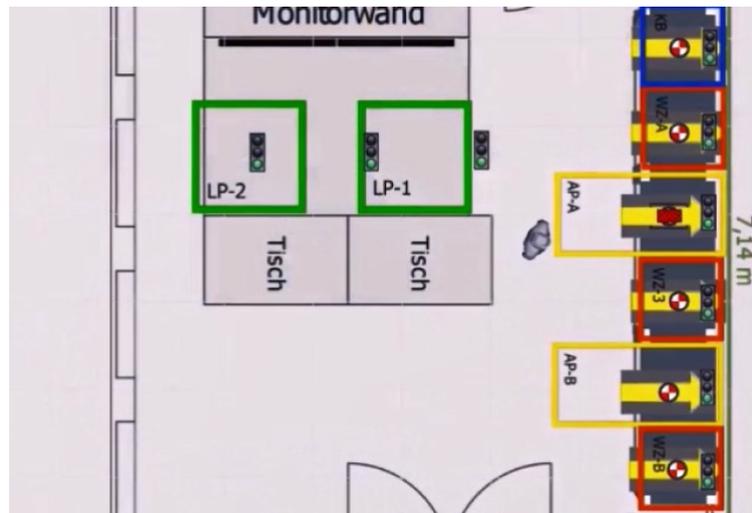


Figure 3.6: Monitor shows the real time stages of mounting process [9]

- Finally, the container is brought to the logistics zone

This process leads to producing products with less manpower, Overall the process is less error-prone as the system is controlling the correctness of the process. By recording every step (even automatically without touching the computer or the parts), the process proceeds faster. All the information is recorded in real time and is only available to the dedicated departments.

Further use cases can be explored on the website: <https://www.platform-i40.de>, which has been introduced by Germany to support projects towards Industry 4.0.

3.2.6 Proprietary And Open Source ERP Systems

There are two different types of ERP Systems in the market, the proprietary and the open source types. The total revenue of the proprietary type was 25 billion dollars in 2013, and in 2017 the revenue increased to 82 billion dollars [10]. This indicates the rapidly increasing importance of ERP Systems in the industry.

The worldwide leading companies are SAP, Oracle, Intuit, FIS global, Infor and Microsoft (MS). Considering the revenue in different countries, this listing may slightly change. For example, in Germany the ranking would be: SAP, Oracle, Sage, and MS. There is no real evaluation about the usage of the open source ERP Systems. However, some research shows the number of downloads, giving an indication of the popularity of each brand. For example, Openbravo registered about 427'203 downloads within a year, while Compiere has been downloaded 135'128 times (November 2017 - November 2018) [11]. The number of downloads vary also from country to country. Though it should be considered, the download does not necessarily mean that the respective companies deployed that software in their organization.

3.2.6.1 Criteria For Selection Of The ERP Software Type

Companies select the ERP System depending on various criteria. Here are the important ones [11]:

- Proprietary [11]
 - Vendor support: Software companies are here to support their customers.

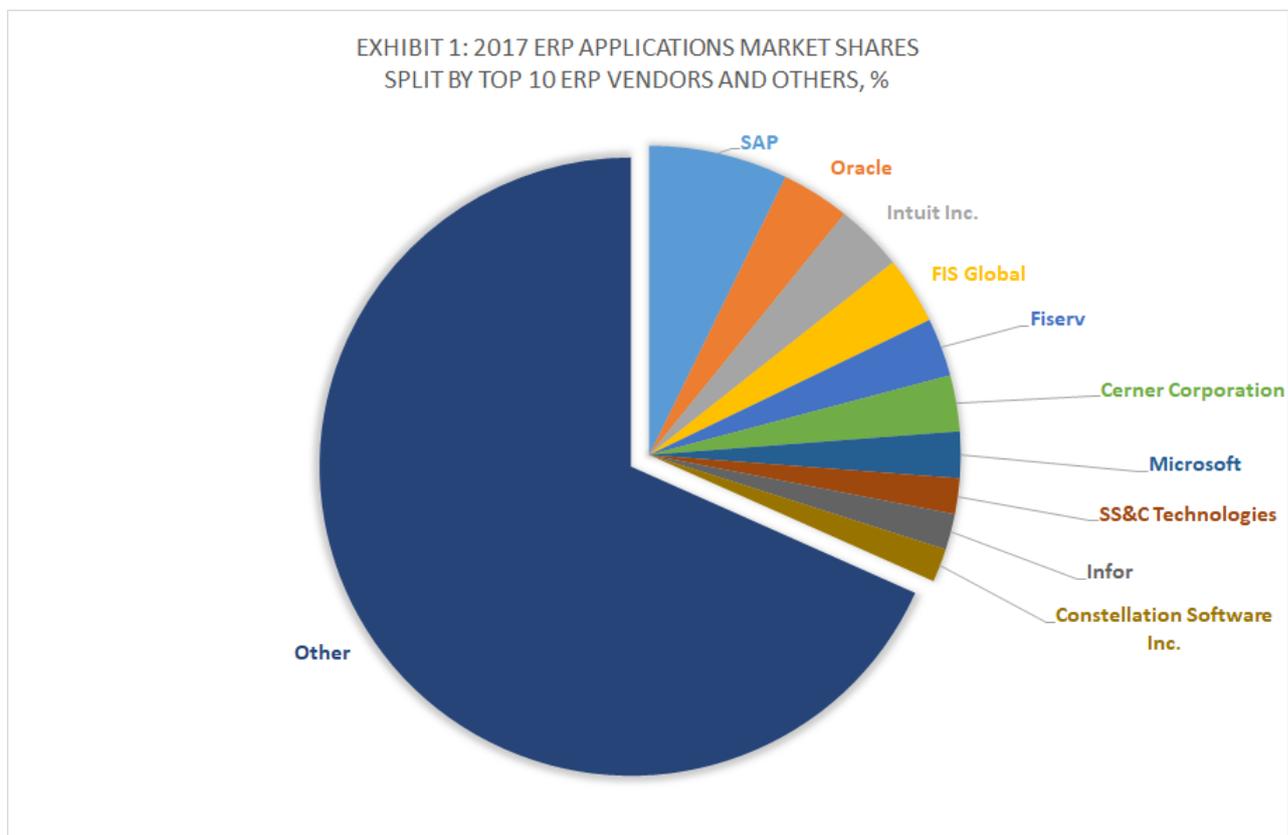


Figure 3.7: World's Leading ERP Proprietary Systems [10]

- Reliability: Companies with a good reputation enjoy commercial confidence and reliability.
- Open Source [11]
 - Ease/speed of implementation: Usually the deployment is much faster.
 - No cost for the software: Though costs for adoption and further implementation to the changing business have to be considered.
 - Independency: There is no guarantee that the ERP proprietary companies will also exist in the future. Upgrading and maintaining when the owner of the software disappears can cause significant problems.
 - Implementation: The adaptation process is also an important aspect in proprietary software. Open source software gives the freedom to adopt independently to the company's needs. There is a trend towards using opensource in developed countries.

It has been observed that there is a trend towards Open Source ERP Systems in developed countries. The underdeveloped and developing countries prefer the proprietary ones [11]. This is a consequence of lacking manpower with technical know-how [11]. Some of the leading Open Source ERP System providers offer maintenance and support services against payment, such as Openbravo [12]. However the majority of Open Source ERP Systems provide only a forum platform or wikipedia page to find help and solutions among other users [12].

3.2.7 Comparison Of Leading Proprietary ERP System Software

This section demonstrates an overview of the leading ERP vendors in terms of their revenue, culture, product considerations, and strength. The revenue given in Fig.3.8 is the total revenue of the vendor. That means in case of Microsoft (MS), the revenue of products other than just ERP products is included. Therefore, these figures only provide an idea about the size of the company but not the ranking of ERP revenue. With its culture of being an innovative industry leader in terms of technology and functionality SAP dominates the market. Nevertheless, the complex end-to-end solution requires a long implementation duration (twelve or more). In this regard, the competitor MS Dynamics (MSD) offers a better option. The deployment takes less time and is easier to apply. Oracle is trying to catch up with the cloud functionality and to deploy Ifor, additional applications are required for certain products. In addition to the main functionalities each vendor provides cloud options. Oracle and MS are one step ahead concerning IoT. The Business Intelligence (BI) and Artificial Intelligence (AI) are the latest development in the industry and are provided by SAP and Oracle (only AI)

Vendor	Annual Revenue	Culture	Considerations	Strength
SAP	\$27.4 billion	Innovative industry leader in terms of technology and functionality	This complex, end-to-end solution requires a long implementation duration (12+ months)	<ul style="list-style-type: none"> • BI, AI functionality • Full suite of business applications (S/4HANA for large organizations, Business One for SME) • Full suite of solutions, including HCM, CRM, EAM, etc.
ORACLE	\$39.8 billion	-	Cloud functionality has not yet been fully developed	<ul style="list-style-type: none"> • AI and predictive analysis • Cloud App • Access to sensor and log data from equipment and external environmental data
Infor	\$2.9 billion	Focused on customer experience and user interface	Tier II applications often require add-on applications for scheduling, advanced warehouse management and inventory	<ul style="list-style-type: none"> • Supports ETO, MTO, MTS and repetitive manufacturing • Provides cloud ERPs for discrete and process manufacturers • Full suite of solutions, including HCM, CRM, EAM, PLM, etc. • Cloud Suites hosted on Amazon Web Services (AWS)
Microsoft Dynamics	\$90 billion	-	IP development is still underway from ISV channel for certain last mile niche functionality	<ul style="list-style-type: none"> • Quick product configuration • IoT integration for field service • Ability to detect issued and troubleshoot them remotely • Hosting possible on Microsoft's cloud platform (Azure)

Figure 3.8: ERP Systems Report 2018 based on [13]

According to another research in 2015 by Mehran University various features of MSD and SAP ERP products were compared [14]. This study shows that all features of MS Dynamics are equal to or better than SAP. With this knowledge, it is surprising that the Market Share of SAP is more than double the size of MSD (Fig. 3.9). The research offers this explanation: 'SAP is the biggest and most noticeable program available' [14].

	<i>Microsoft Dynamics AX</i>	<i>SAP</i>
<i>Market Share</i>	<i>11%</i>	<i>24%</i>
<i>Average Time for Implementation</i>	<i>10 Months</i>	<i>15 Months</i>
<i>Cost of Implementation</i>	<i>Low</i>	<i>High</i>
<i>Base of Usage</i>	<i>High</i>	<i>Mid</i>
<i>Target Market</i>	<i>Large and Mid-Small Sized Companies</i>	<i>Large sized Companies</i>
<i>Customization</i>	<i>High</i>	<i>Low</i>
<i>Flexibility</i>	<i>High</i>	<i>Low</i>
<i>Returns on Investments</i>	<i>High</i>	<i>Mid</i>
<i>Ease of Integration with other Software</i>	<i>High</i>	<i>Low</i>
<i>Performance</i>	<i>High</i>	<i>High</i>
<i>Risk</i>	<i>Low</i>	<i>High</i>
<i>User-Satisfaction</i>	<i>High</i>	<i>High</i>

Figure 3.9: Comparison between Microsoft Dynamics AX And SAP 2018 based on [14]

3.2.8 Challenges And Opportunities

Overall there are tremendous opportunities like big data in real time, customized production, optimization of resources, automation, elimination of manpower mistakes, and saving tremendous costs. That being said, there are real challenges that must be considered, such as, agility, integrity, ease, security, scalability and consistency.

3.3 IOT in ERP

3.3.1 IoT Introduction

Under the umbrella of the Internet of Things are technologies which allow for nearly every physical device to be represented in the networking world. While there exist many different definitions for the Internet of Things it can be agreed on, that in its centre is a shift which moves away from data created by users to data created by machines. In this sense one definition would be: "An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment". [20] To make the Internet of Things reality, three categories of devices are needed from a high-level perspective:

- sensors, actuators and embedded communication hardware [20] .
- on demand storage and computing tools for data analytics [20] .
- new and easy to understand visualization and interpretation tools which can be widely accessed on different platforms and which can be designed for different applications [20].

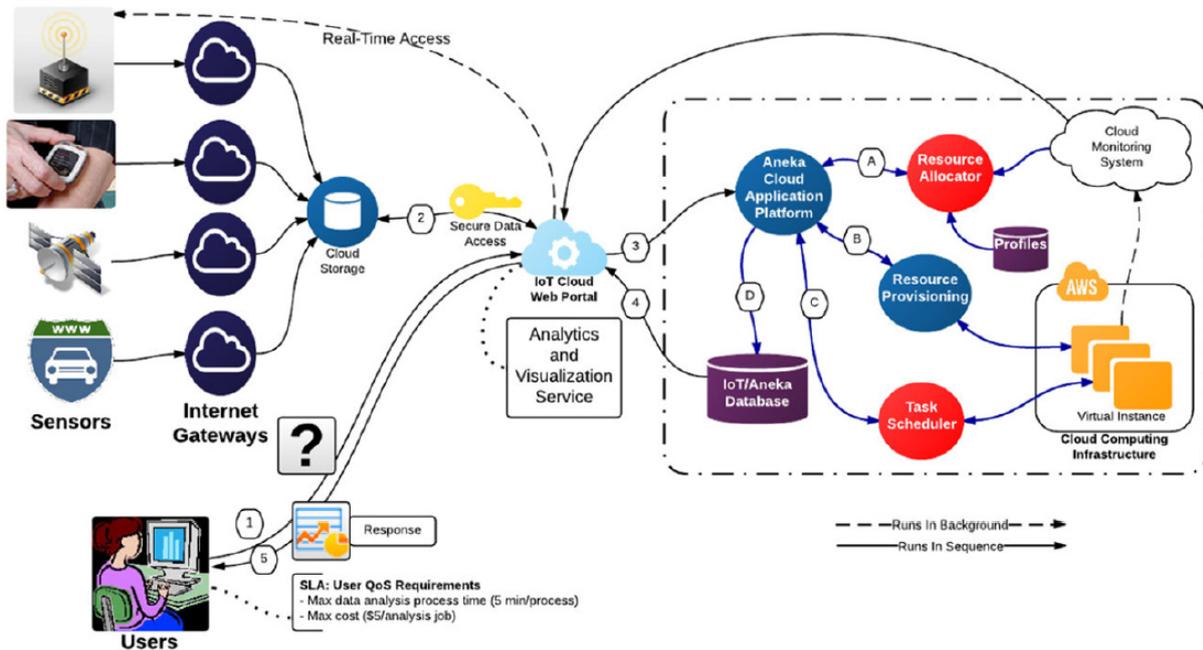


Figure 3.10: Interactions in Cloud Centric IoT framework [20]

Under category (a) fall device such as RFID chips, which are basically an electronic barcode. They allow for cheap identification of anything they are attached to and they don't need electricity because they are passively powered by the reading devices. This makes them very viable in supply chain management because in theory every product can be outfitted with one. The range of other sensor devices is nearly endless. They could measure temperature, proximity, smoke and much more. To transfer all this additional gathered data, different networking technologies can be used. From standard technologies such as Wi-Fi and Bluetooth to more specific solutions like ZigBee which benefits from very low implementation cost. The capturing of all this data calls for an unprecedented amount of (b) Middleware to on one hand store the information in data centres, but also to make use of the collected information. Neural networks, artificial intelligence and data

analytics in general will have to make sense of the stored data. As a last step (c) the gathered information must be presented to the users in an understandable manner. So, users will be able to interact with it. [20]

IoT devices in general have an enormous potential to improve ERP systems. They can provide real time data which can either be accessed instantly or after being preprocessed and analysed in bulk. These improvements won't only affect producers and factories but will also bring utility to the customers. The ERP systems today rely on a huge amount of collected data to be effective and IoT can bridge the gap to provide them with the necessary information. Among the potentially gathered data there will be information to improve business intelligence, customer service as well as inventory management. But to get to these mountains of data, it is of crucial importance to first equip their resources, machines and products with the necessary sensors and build an infrastructure to handle the created data flow, while keeping additional costs in check.

3.3.2 Application Of IoT In Manufacturing

Originating from an RFID system presented by MIT Auto-ID Labs in 1999 IoT has developed from only interconnecting things to moving the physical world into the web . In manufacturing it can be used to improve connection, communication, computing and control of resources and capabilities. [25]

Figure 1.11 proposes a concept of how to integrate and connect IoT devices to enterprise information systems. enterprise information systems is a general term containing different software including ERP's which are used by employees. It focuses on the activities inside the workshop and how information is passed to these systems with the goal of automatic control of the workshop floor.

The proposed manufacturing execution layer focuses on access, identification and control of the production process from materials to the finished product. Described in figure 1.11, the IoT-enabled manufacturing layer containing different sensor and network components passes the gathered data to enterprise information systems which in turn feed information to the automatic control layer. On an enterprise level it calls for heavy integration of product related information and management information into enterprise information subsystems. This helps building a platform for solving problems regarding large scale collaboration by interconnecting all kind of resources. [25]

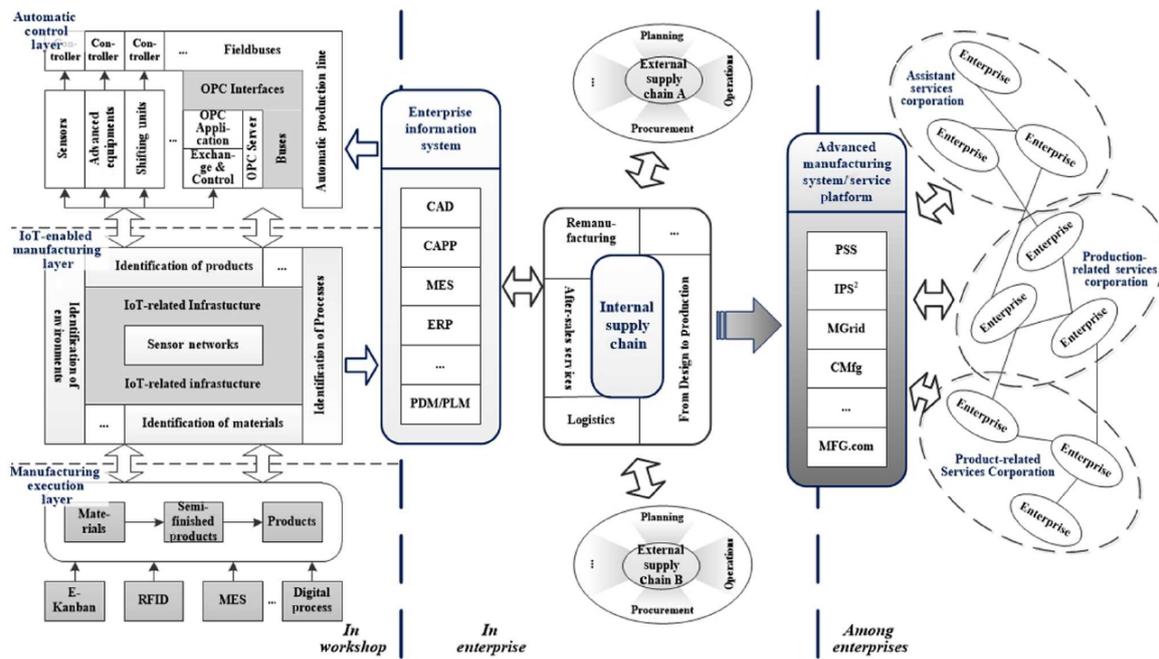


Figure 3.11: Applications of IoT in manufacturing [25]

3.3.3 Readiness Of ERP Systems For IoT

Future customers will have many new needs concerning ERP systems. Because collaborative manufacturing is more prevalent than ever before, one requirement for ERP's will be, that they completely integrate supply chain systems in order to ensure the quality standards of the product by guaranteeing that the correct raw materials were used during production. [22]

Therefore continuous communication between humans, machines and processes is needed. But it's still not clear how ERP will improve communication. As mentioned later, SAP is already building some application to facilitate this, by integrating machines into ERP systems to track maintenance. But today ERP systems are still relatively bad at predicting events.

A case study including different manufacturing companies in Egypt came to the conclusion that in general today's ERP systems should be ready to support smart factories. While nearly all daily business is already controlled via ERP from product planning to quality control, data is mainly still entered by hand. While many ERP manufacturers say that their frameworks for sensor and machine integration are ready, organizations also were concerned about the interfaces connecting their equipment to the ERP system. At the moment the task would simply be too big. Because many manufacturers use their own communication standards between their machines a lot of middleware would have to be constructed to allow all the different brands of machines to communicate with each other. [22]

3.3.4 Integrating Supply Chain Planning

While modern ERP system have already come a long way with regards to Marketing and Sales, Finance and Accounting, Manufacturing as well as Human Resources, they still lack full integration of Supply chain planning. Up to now they mostly had little if not no interaction with outsourcing suppliers as well as customers. But focusing on integrating these aspects could eliminate a hug amount of duplicated work. [26]

Supply Chain Planning follows the transformation of goods from raw materials to the finished product while exchanging information with customers as well as suppliers. By tracking raw materials from the moment they enter the factory, while being stored as finished product in the warehouse and finally when the product is distributed, many new possibilities arise for optimizing production. It allows for fewer materials being stored and a more exact amount produced as well to name a few. Because information flows up, as well as downstream the supply chain not only internal issues are affected but information can also flow to suppliers to optimize deliveries or to customers for additional information about their order. This can lead to cost reductions, additional customer satisfaction as well as competitive advantages. [26]

For tracking these processes, RFID chips have shown to be the perfect solution, because they allow for data capturing at every gate and are already widely available for a competitive price.

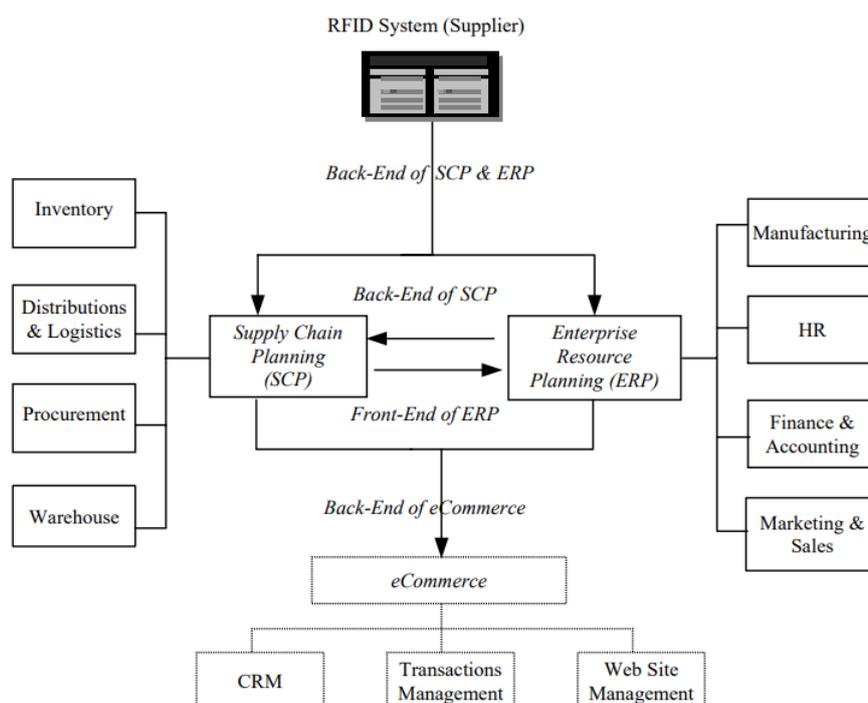


Figure 3.12: Relationship of ERM and SCP [26]

3.3.5 RFID In Production Management

To monitor the flow from raw materials to a finished product up to the customer different implementations are suggested. Most focus on RFID tags and scanners at every gate. An application could work the following way. Every raw material is immediately outfitted with an RFID tag as soon as it arrives at the factory. This RFID chip is then entered into the company system with all the necessary product information. When the material is moved to the warehouse it has to move through an RFID scanner gate. There the item is immediately registered and from now on it's possible for everyone looking at monitoring terminals, to determine where the product went. [27]

When leaving the warehouse, the raw material is scanned again as well as when entering the production line. There it's possible to scan the materials at every essential assembly step to exactly monitor how many parts exist in which cycle of production. The same happens for intermediate storage as well as when the finished product finally leaves the factory. [27]

The gathered information can be made available on different levels. On one hand it should of course be made available for workers at every step in the factory through local terminals, which show them information relevant for them. On the other hand the information can also be used from a high level perspective to monitor a whole factory when the data is gathered and visualized properly. [27]

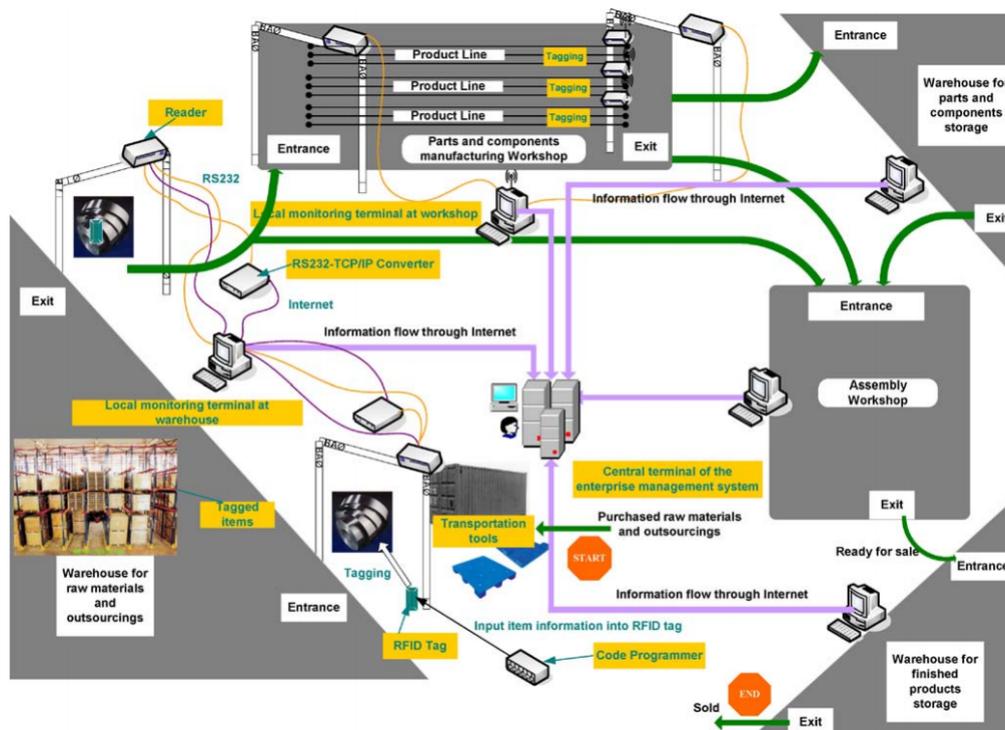


Figure 3.13: Conceptual layout of RFID in a factory [27]

The mentioned application was retrofitted into an existing factory. Therefore, it was necessary to heavily rely on wireless technology because it's just too expensive to completely rework a factory floor just for a new layer of applications. In said example the data from the RFID readers was transmitted by Bluetooth if the range allowed for it, otherwise wireless LAN was used to transfer data to local terminals. From there on it was then mostly cost effective to use wired solution to store Data on a central Database. [27]

3.3.6 IoT Today's ERP's

Considering the different applications for IoT in the future its interesting to have a look at what's already possible today with existing systems. SAP and Oracle with their leading role in ERP systems already pave the way with IoT enabled Services.

3.3.6.1 SAP

In SAP there exist the following four main applications, inspired by IoT.

- SAP Connected Goods allows users to connect, monitor and control different devices used by their customers. It allows for maximised performance by tracking how,

where and how often certain materials are used. This allows for better operation condition as well as improved customer service. [23]

- SAP Vehicle Insights allows to map and analyse sensor data in vehicles. Using real time GPS data in combination with enterprise and customer data it allows for optimized fleet utilisation. [23]
- SAP Predictive Maintenance Software combines sensor data from ERP, CRM and enterprise asset management. Using this data, it aims to improve maintenance planning and therefore reduce downtime of resources to improve production. [23]
- SAP Asset Intelligence Network allows operators to access up to date maintenance strategies from manufacturers and they also automatically receive usage and failure data of their products. It focuses on collaborative management as well as exchanging asset performance data. [23]

3.3.6.2 Oracle

Oracle provides similar programs.

- Production Monitoring Cloud Service provides a real time view of the shop floor to track performance of machinery and predict failures to take preventive actions [28] .
- Asset Monitoring Cloud Service provides tracking information of equipment also with the goal to ensure uptime by predicting failures [28] .
- Fleet Monitoring Cloud Service allows to track vehicles in real time to manage a fleet and improve routes as well as safety [28] .
- Connected Worker Cloud Service allows to track workers in real time in their environment. It allows to ensure regulatory compliance as well as environmental conditions with the goal to reduce employee accidents. [28]
- Service Monitoring for Connected Assets Cloud Service allows to track different information in assets and make the information available to support workers to improve their customer support [28] .

3.3.6.3 Comparison

Both companies provide very similar features. They both aim to improve maintenance in factories, provide better customer support and track vehicle fleets. The main difference being that Oracle allows to not only track goods and vehicles but also each individual worker on a customized factory layout map similar to how vehicles are tracked. Qualitative differences in their features would have to be further examined.

3.4 Integration Of Blockchain And ERP

3.4.1 Blockchain Technology

The Blockchain is a data structure, where blocks of information are concatenated. It is not possible to remove a block from the blockchain once it was successfully added. Therefore, the blockchain keeps track of all transactions that have ever happened since its creation. The blockchain is a distributed data structure, where many or even all participants store a copy of the whole chain on their servers and check that it's never tampered with. Whenever a server wants to add a block to the blockchain, it must propose this block to all the other servers, which then validate it and add it to their chain if they found it to be valid. The validation procedure is defined by the protocol and therefore all servers following the protocol will always end up with the exact same blockchain in the end (sometimes two different valid blocks are proposed at the same time by different servers, since only one block can be added at a time, one must be rejected. Which one will be rejected is also clearly defined). [16]

The reason the blockchain can be interesting compared to a classical database, is because due to the decentralization and the validation of each block, it is easy to verify for each reader of the blockchain, if the information in the blockchain is valid or has been modified. The state of the blockchain is verifiable by anyone allowed to read it, all the time. If the blockchain is distributed among enough nodes, then changing it after it has been accepted by most nodes would cost so much effort, that it's practically impossible to do so, we can say the blockchain is immutable and no transaction is reversible. Resulting from that, there is a clear version of the truth and no one can claim something different. Therefore, there is no need for a third party in-between anymore that enables trust between two parties that want to transact, but don't know each other. [15]

3.4.2 Applications Of Blockchain Technology

Currently there is a lot of research on where the blockchain can be used. One of the most promising applications besides the use in the financial sector seems to be the supply chain [17]. Supply chain management is a key element for many companies, since being good in processes that happen inside the own walls of the company isn't enough to be successful in the market and companies must rely heavily on their suppliers to deliver high quality goods on time [17]. Most things we consume have a very long and non-transparent supply chain. Components for a product can come from very different regions in the world and no one can really know if all steps were taken responsibly. Often customers are concerned that humans or animals weren't treated fairly and responsibly, or that the producer doesn't take care of the environment. Other concerns are for example if food or medicaments were stored correctly and with the right temperature such that they are in good quality when they arrive at the customer. Many problems could be solved by making sure that suppliers must act responsible and face consequences when not doing so. With the use of the blockchain in the supply chain many engineers are now trying to make the processes in the supply chain more transparent by tracking who is doing what, when and where [17]. The use of IoT devices makes it a lot easier to monitor the processes in the supply chain, by automatically recording information without the need of humans entering the information. For example, by monitoring the temperature of the delivery bus that delivers temperature sensitive medicaments. This makes the supply chain monitoring more tamper proof. [15]

The blockchain builds trust among participants of a supply chain without the need of a third party, since it's possible for one participant to track what the others are doing and knowing that this information is correct and immutable. With the elimination of third-

party and the increase in transparency, costs are reduced (e.g. because food will less often be spoiled, each company knows exactly when to expect deliveries etc.), efficiency is increased, and everyone has a better view of the big picture of what's currently going on. [17]

3.4.3 Concept For The Integration Of Blockchain Into An ERP System

3.4.3.1 State Of Research

Research about the Integration of the blockchain into an ERP system is only in its beginning and in the future many technical and integrative challenges will need to be solved. There are first projects and attempts that try to make a use of the blockchain in an ERP system, but currently there is no ERP system that truly uses the blockchain as a standard and the technology is not widespread yet. It is difficult to start using the technology because there are no standards yet and most likely all the business partners of a company still don't use blockchain technology such that there are no network effects that are necessary to really profit from the technology. [18]

3.4.3.2 Potential Of The Blockchain

In future ERP systems should have an interface to allow communication over blockchain with other ERP systems. It is much too inefficient and costly to run a whole ERP system on a blockchain, such a system would be much slower than a traditional database while not having many advantages over a traditional database. The strength of the blockchain lies in building trust and storing information immutably, but inside a company those qualities are not demanded. Those qualities have however high demand when it comes to communication between different parties. [18]

The realization of a complete integration of Blockchain and ERP is therefore neither scheduled nor possible at the moment. Instead it is suggested that the ERP system that handles processes inside the company still uses a traditional database (e.g. S/4HANA) but for the communication between ERP systems of different partners blockchain technology should be used. This way the ERP system is very performant for company internal tasks through the main memory database and doesn't need much computing power like a blockchain would use while the potential of the blockchain is still used in the communication with business partners. [18]

The potential use of the blockchain is as a cross-company data-pool, where all important transaction data of business partners is immutably stored, this way it's used as a single-source-of-truth that proves what agreements were made and what actions were taken in the past. Every participant of that system can then always access the important transaction data and be sure that no one can temper it and be sure that no one can argue that is isn't correct. Because the blockchain ensures that all participants that use the correct algorithms always end up with the exact same chain, the data among all participants will always be consistent and therefore many misunderstandings and unnecessary costs can be prevented. [18]

The increased trust and the consistent data among partners is not the only advantage the blockchain brings in the communication and coordination between partners, another advantage is that it is much easier to know in which state the business partners are in and therefore also to know in which state a order is in. This is because important documents and actions that are of interest for others are uploaded to the blockchain immediately, as soon as the documents are created or as soon as the action is taken. If a partners wants to see how much of a part his supplier has in stock right now, he can check this in the

blockchain, or if a restaurant wants to know which fish a fisher fished and when, they just check that in the blockchain. [18]

When the state of other partners is always known, processes become much faster, cheaper, more transparent and administrative overhead is decreased. For example, it can be automatically tracked over the blockchain when certain parts of the supplies leave their warehouse and when they should arrive at the own warehouse. Preparations can then automatically be tasked for an efficient process flow. The next possible advantage is that processes are monitored and controlled at a central place, namely the blockchain. The whole supply chain of a good can be easily traced with the transaction data that is stored in the blockchain and it is even easier to automate processes, since the necessary information is uniformly and in real time written to the blockchain. [18]

Because all this information is accessible among business partners and all the information is immutable and consistent it is much easier to have trust in business partners and intermediary organizations that were previously used to build trust among business partners are no longer needed and need no longer be payed. [18]

The increased trust lowers the barriers to enter a market which brings profit to the end customer and to the companies that do a good job and are important for the creation of value. This is due to the fact that the reputation is a bit less important because reputation is replaced by data in the blockchain to a certain degree. [18]

3.4.3.3 Requirements

With the goal in mind to improve the procure-to-pay process for the automobile industry, Linke and Strahringer worked together with the Daimler AG to create a prototype of a system that fulfills the above advantages. The procure-to-pay process covers everything from goods sourcing to goods arrival to invoice arrival to payment. The problem that the procure-to-pay process currently has is that it normally uses many different communication channels, which causes slower processes, inconsistencies in data and administrative overhead. Linke and Strahringer made interviews with different stakeholders of the procure-to-pay process and experts of ERP and blockchain and found that the functional requirements of a procure-to-pay process with blockchain are:

- That the blockchain simplifies Process automation,
- Makes easy traceability of transactions possible,
- Ensures uniform representation of data,
- Ensures trustworthiness of transaction data, meaning that the data is correct and immutable,
- And the blockchain should be used as the single source of truth, so all important transaction information is always stored in the blockchain and is binding. Further they found the technical requirements to be:
- Interoperability, meaning that different ERP systems can work together even though they may represent and process data in a different way internally
- Integration in existing systems and processes, meaning that the ERP systems that are in use today should not have to change much, since this would be very difficult and costly
- loose coupling, to ensure that the ERP system and the blockchain technology can be changed independently

- Scalability, for making sure that the system works for big companies
- High transaction throughput, such that many transactions can take place at a time without congestion in the network
- Harmonize the different transaction systems of ERP-systems and the blockchain, such that they can flawlessly work together
- Data synchronization between blockchain and ERP-systems, such that important data is automatically fetched by the ERP once it's available in the blockchain (then it can be further processed) and the important data is automatically pushed from the ERP to the blockchain
- Mature authentication and access management, such that only authorized people can read and write to the blockchain

Daimler used SAP S/4HANA for internal tasks and employed an intermediary company that helped them with the communication with their suppliers. This intermediary company operates the Daimler Supplier Portal, where many of the transactions between Daimler and its partners happen. The problem is that there are multiple other channels (for example email or telephone) where transactions can happen as well. The variety of communication channels can lead to inconsistencies in data and misunderstandings. Another issue is that it can be hard to prove what was agreed upon on the telephone and even over email, because sometimes the data is just not available anymore, or the data would be available but finding it is very expensive and difficult. In the graphic the communication channels between Daimler and its partners are depicted. Although most transactions happen over the Daimler Supplier Portal, there are also many transactions over telephone or email directly between the partners. It is also to be seen from the graphic that different partners use different ERP systems, which underscores the technical requirements that different ERP systems need to be able to connect to the same Blockchain system and that the existing systems need to be considered. [18]

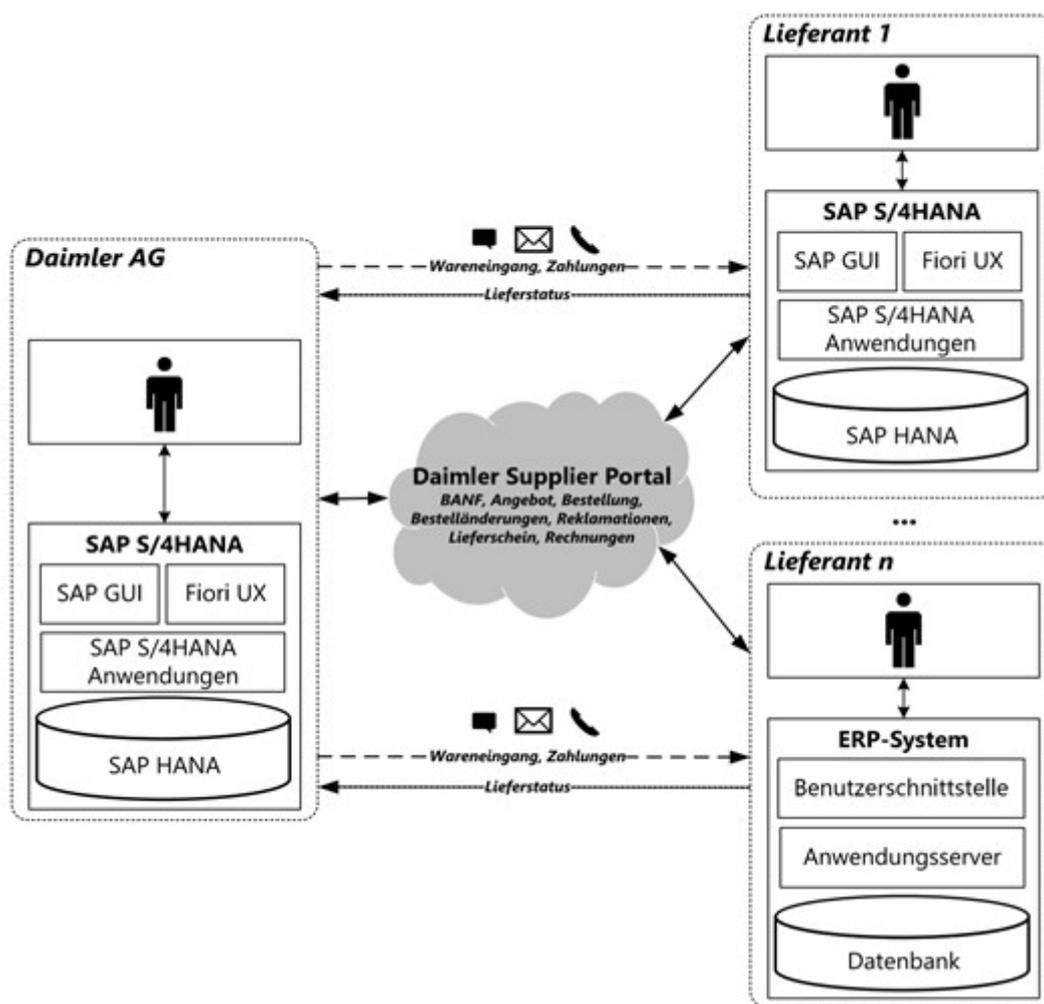


Figure 3.14: Daimler System Today [18]

Although the processes within Daimler work very well and fast because of the support of the SAP S/4HANA ERP system, the processes that happen between Daimler and their business partners could be enhanced. The concept that Linke and Strahringer suggest solves the communication problem with the following setup: [18]

3.4.3.4 Company Internal ERP System

Inside the company borders, the traditional ERP systems are used to manage the internal steps of the procure-to-pay process like it was before. Inside the own company processes are already standardized and making all transactions within the company over the blockchain would cost too much. Furthermore, the used database for the ERP system already represents a single-source-of-truth inside the company, since trust is assumed within a company and the ERP database is the only place where data is stored, which assures consistency. When multiple business partners transact with each other, they use different databases to store their data, which can lead to inconsistencies much faster. [18] Another reason to stick to the classical database for processes within the company is that a classical database is still more stable, performant, has higher throughput and shorter latency than a blockchain, therefore with a classical database processes are as fast and cheap as it's possible right now. [18]

The suggestion is that the company internal ERP's should be changed as little as possible for the deployment of the blockchain, because this would only lead to unnecessary effort. It is difficult to change all the different ERP systems, because those are huge systems from different vendors that evolved over the years, it is however much simpler to make a

new piece of software that handles the communication between the systems as they are. The potential of the blockchain is in the cross-company collaboration as a single-source-of-truth and should be a standalone software that uses an interface to the ERP systems in order to achieve loose coupling. [18]

It is critical that the blockchain technology and the ERP system are very well geared to each other to ensure that processes can be automated, and no steps will faulty. With the distributed ledger, the transparency, the high availability and the immutability of the blockchain, the blockchain is well suited to replace intermediaries between companies to safe cost and make a more direct and faster communication possible. In the case of Daimler, the blockchain technology will replace the Daimler Supplier Portal. Cross-company transactions and processes like offers, orders, goods receiving, complaints, invoices and payments will then be done over the blockchain while transactions within the company that shouldn't be public to any business partner are still done over the traditional ERP system that is already deployed and in heavy use in the Daimler AG. [18]

The blockchain technology is better suited for the job that the Daimler supplier Portal operators did, since it enforces a uniform representation of data, which is especially important if processes that should be automated, and it builds trust to business partners, because of its immutability. With the usage of smart contracts, the automation of processes can be simplified further. [18]

3.4.3.5 Blockchain Technology

For the blockchain technology itself, a private blockchain should be used, such that only authorized business partners can access the data. Different partners will have different rights, concerning what they can read from the blockchain and what they can write to the blockchain. The rules will be made by the blockchain operators. To deal with performance issues there could be a database belonging to the blockchain such that big detailed documents can be stored on that database instead of on the blockchain, while the blockchain will store the reference and the hash-value of that document. This way the blockchain will keep a smaller size and is easier to store and operate for all business partners. The graphic shows how the communication between the different business partners should take place. The Daimler supplier portal is replaced by the blockchain technology, but all partners keep their ERP system, that they used before. The communication often happens indirectly and doesn't need a human to human or human to portal to human interaction. Because all the data is in the blockchain it is easier to find and is uniformly stored. [18]

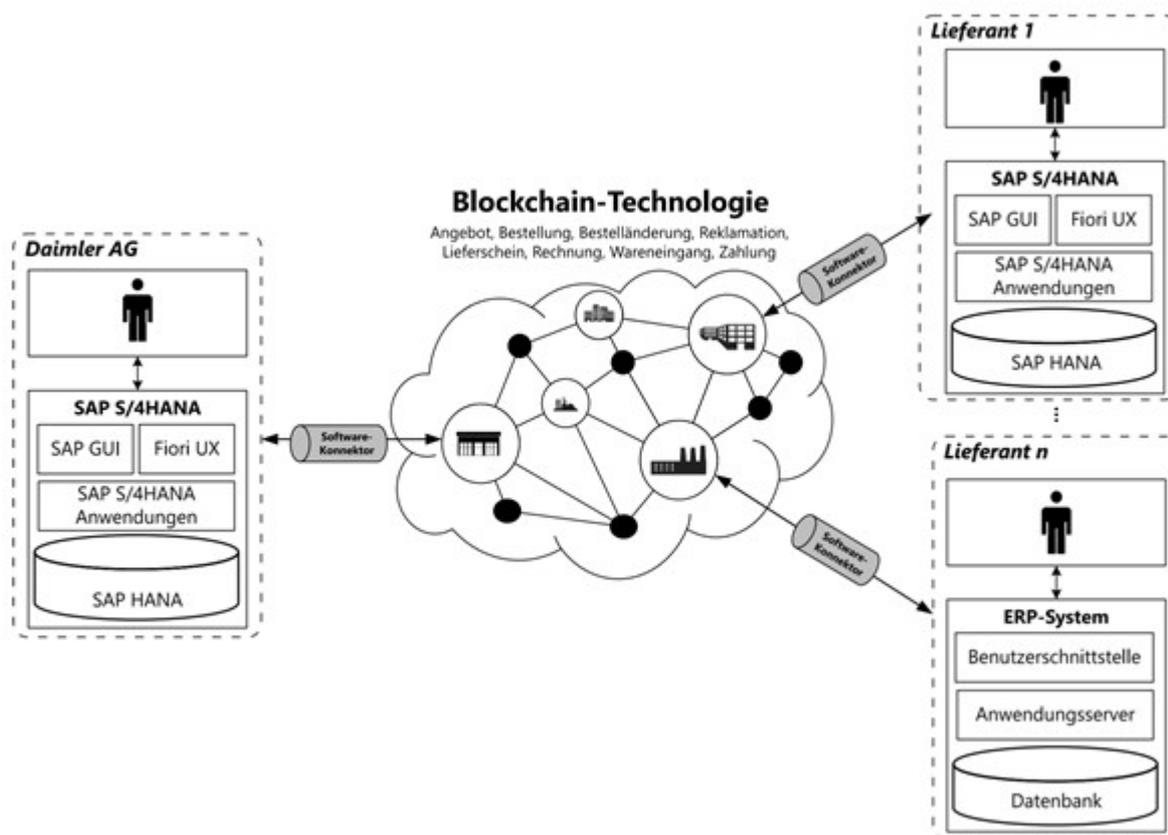


Figure 3.15: Daimler System With ERP [18]

3.4.3.6 Connection Between ERP And Blockchain Technology

The blockchain and the ERP system will be connected via a software connector. The connector is needed, such that the ERP system and the Blockchain don't have to be directly connected and therefore ensuring that the variability of the Blockchain and the independent evolution of the ERP system is guaranteed. The connector is a separate piece of software and should work together with ERP systems of different vendors. The connector software handles the transmission, processing and transformation of transaction data, such that the ERP system and the blockchain technology don't have to be adjusted too much but can rely on the connector software to adjust the data. The connector can use the interfaces that all big ERP systems already supply. SAP for example already has an interface on the basis of OData where transactions with the ERP system can be received, changed or created and using ODBC or JDBC one can receive data directly from the database that SAP uses. The connector should also have a function to read and write the blockchain using a standardized interface, therefore making it possible to exchange the blockchain technology without much effort. The connector also manages authorization and validation such that only authorized organizations can read and write the blockchain. [18]

The graphic shows that the Software connector interacts with different ERP systems via an interface. The connector is also connected to the blockchain system via an interface. The connector uses adapters to make the communication between different ERP systems and blockchain systems possible. If a system needs to be exchanged, then only the adapter in the connector needs to be adjusted, which will be much easier than if the blockchain technology would be embedded in the connector or even in the ERP. The evolution of the whole system is therefore much easier. [18]

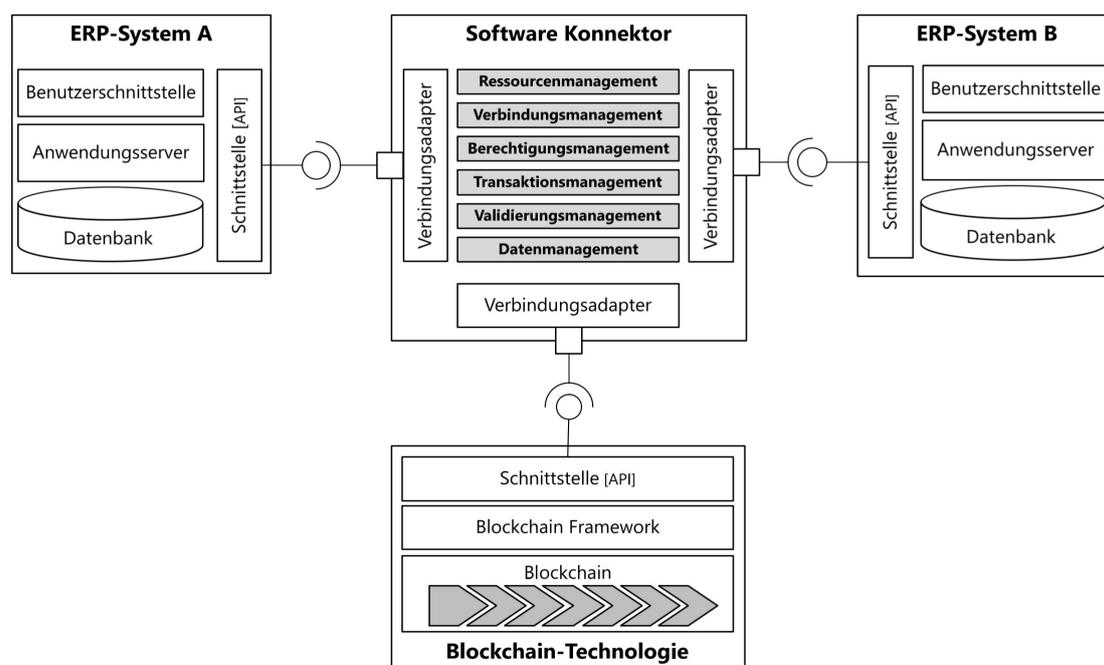


Figure 3.16: Software Connector [18]

3.4.3.7 Conclusion

Using this concept intermediaries can be replaced by the blockchain technology, since it enables a tamper-proof, transparent and trustful way to make transactions. Replacing intermediaries reduces cost, furthers independence and reduces the risk of failure. Market entry barriers are reduced because due to the single-source-of-truth, new companies are more trustworthy. Because all important data is stored in the blockchain, data consistency and uniformness are assured, and processes become faster and more reliable. Transactions and supply chains are easily traceable since all the necessary data is stored in one place, namely the blockchain. [18]

In future the governance of the system should be researched, for example which institution is responsible for the operation and the maintenance of the blockchain. The collaborative administration of business processes or the standardization and maintenance of smart contracts are other topics. There is also a need for models, case studies and technical solutions for the integration of blockchain and ERP systems. [18]

3.4.4 Examples Of Blockchain In Operation In ERP / Supply Chain

So far we have seen a concept of a system that uses blockchain for ERP tasks, but what is still missing is a real system that uses blockchain for ERP and is in operation. We saw in the previous section, that it isn't performant enough and has few advantages to base a whole ERP system on the blockchain, but using a combination of a traditional ERP system that utilizes blockchain technology in combination with a classical main memory database can be very promising. In this section I will present some examples I have found of companies that use blockchain technology to monitor and improve their supply chain.

3.4.4.1 Maersk

Maersk is one of the biggest container shipping companies in the world and a good example of a company that successfully incorporated the blockchain into their processes. Before they had the blockchain they had shelves and shelves full of paper, used to track where

their containers are at the moment and also for other organizational issues like to clear the goods for import. This is not only very costly to manage (doing the paperwork could cost as much as moving the goods inside the container), it can even delay many shipments due to lost papers or even containers. With the new system, everything is stored in the blockchain, and everyone who needs access to this information can read it from the blockchain. For example customs authorities can upload a copy with a digital signature of every document they signed off. This allows for faster information flow between Maersk, its customers, and authorities, as well as better information storage, since everyone can apply to the blockchain to see what happened in the past, and be sure that the information has not changed. Maersk recorded what container was where to which time and in what condition (e.g. temperature) it was. Frauds are very common in global shipping market (e.g. copying lading bill), with blockchain money will be saved by decreasing fraud, increasing information flow and trust. [17]

3.4.4.2 Modum

The swiss startup Modum worked together with the University of Zurich to make a system, which monitors shipments of medicaments. A regulation of the EU requires that shipments of medical goods requires the companies to report to the receiver if there have been temperature deviations during the shipment. The only way to ensure that the shipper knows if there have been deviations is either to use the much more expensive refrigerated transport, or to constantly monitor the temperature. When there were deviations, the medicament might not be usable anymore. There are three categories of medicines, ones that need to be stored cold (-20°C), ones that need to be stored cool (-2°C), and ones that can be stored at ambient temperature ($15-25^{\circ}\text{C}$). Each medicine has data called "stability data" that specifies how long a medicine can be stored at what deviation from the temperature it should be stored at. Modum focuses on monitoring the transport of medicine that can be stored at ambient temperatures, since monitoring is cheaper than using a refrigerated transport, and the refrigerated transport is not really necessary for this medicine. The system of Modum saves cost, since less people are involved in the shipping process, less paperwork is required and cheating in the process gets much more complicated. The sensors of Modum automatically send data to the Ethereum blockchain, where a smart contract is set up, that automatically informs the sender and the receiver, if the ambient deviations during the shipment were too big. If the deviations are okay, the product is automatically released. [17]

3.4.4.3 Walmart

Walmart made a project, where they monitored Chinese pork with the blockchain. With the blockchain they could track individual pork products with attributes like details of the farm, factory, batch number, storage temperature and shipping information using the RFID tags, barcodes and sensors they already heavily use in their markets and processes. With the recorded information it was possible to track the individual pork within minutes, instead of days, like it was before the use of the blockchain. If spoiled food was detected, it is now possible to know exactly which food they must take away from their shelves and take back from the customers, instead of having to take back the whole product line, like it was normal in the past. With the richness of information, they have about the pork, they can now also judge whether the pork was counterfeit or actually from the authenticated factory and it is easier to assess the expiration date of the pork. With the much more strategic approach it is possible to save cost and enhance food quality. [17]

3.4.4.4 SAP

Sap uses the blockchain to let their customers track goods along the whole logistics chain (also for pharmaceutical goods that have to comply with the US Drug Supply Chain Security Act), they also offer a blockchain service that simplifies shipping management (network optimization). [19]

3.5 Conclusion

In this paper an introduction to ERP Systems is given. ERP is the integration of organizations departments and functions in a single computer system. With the efficient management, storing and tracing of resources, ERP leads to tremendous cost savings. Indeed, ERP supported processes in Industry 4.0 lead to produce products with less manpower, and are less error-prone as the system is controlling the correctness of the process. ERP ensures that all relevant and exact needed amount of material for a business or its product is available in the right place and at the right time. Another important benefit of the ERP system is that all relevant information is available in real time. The main goals of an ERP system are data integration, function integration, process integration and program integration. This paper gives an overview of the evolution of ERP systems and their architecture from the 1970s until today. Today the ERP systems are pushing the limits of what is possible in Industry 4.0, the vision of full automation of the life cycle of industry products. That starts with the production of the raw material, continues with the production process, and ends with the recycling process. This research compares the leading ERP systems. The three worldwide leading vendors are SAP, Oracle and Intuit. There exists two different types of ERP Systems, the proprietary and the open source types. The main benefits for the proprietary systems are the vendor support and the reliability. Cost savings, ease and speed of the implementation, no cost for the software and the independency from the suppliers are the advantages when deploying the open source ERP Systems. It has been shown that the number of open source ERP users are increasing especially in developed countries as it requires technical manpower. The leading open source systems provide maintenance and support services. A future research could be done by exploring all available open source ERP systems, identifying their scope of functionalities which provide support services.

We explored how IoT can solve the huge demand that ERP systems have for data. Sensors that are used everywhere produce a lot of data for the ERP to use. IoT try to represent physical devices in the digital world. Three categories of devices are needed in IoT, namely Hardware, Middleware and Presentation. Information needs to be recorded, analyzed and presented. We explored which technologies can be used for IoT and their benefits. Often, they do not even need electricity since they are passively powered. In manufacturing IoT can be used to improve connection, communication, computing and control of resources and capabilities. We discussed the readiness of ERP systems for IoT, proposed improvements to ERP systems using RFID, and looked at the case of SAP

We gave a quick introduction of Blockchain technology, talked about some applications of blockchain technology in supply chain monitoring and presented a concrete concept. The benefits of the single-source-of-truth, the immutability and the consistent representation of data have been presented. The requirements that the blockchain needs to fulfill in order to succeed in the ERP system where shown. At last some examples of actual examples where blockchain technology is used in ERP systems have been presented.

Bibliography

- [1] F. Robert Jacobs, F. C. 'Ted' Weston Jr.: *Enterprise resource planning (ERP) - A brief history*, Science Direct, Journal of Operations Management, March 2007, <https://www.sciencedirect.com/science/article/abs/pii/S0272696306001355>.
- [2] August-Wilhem Sheer: *Wirtschaftsinformatik Studienausgabe, Referenzmodelle für Industrielle Geschäftsprozesse*, Springer, Berlin, Germany, 1995.
- [3] Kenneth C. Laudon, Jane P. Laudon, Detlef Schoder: *Wirtschaftsinformatik, Eine Einführung*, Pearson, Hallbergmoos, Germany, 2016.
- [4] Euiho Suh: *Enterprise Resource Planning*, POSTECH Strategic Management of Information and Technology Laboratory, Dept. of Industrial and Management Engineering, February 2012, <https://fdocument.pub/document/enterprise-resource-planning-erp-568506b08ec61.html>.
- [5] Simha R. Magal, Jeffrey Word: *Essentials of Business Processes and Information Systems*, Library of Congress Cataloging-in-Publication Data, Wiley, USA, 2009, pp. 29.
- [6] Markus Diesner: *Architektur Modell für Fertigungs-IT, Das Internet of Things produktionsnah strukturieren*, IT & Produktion Online, Das Industrie 4.0 Magazin für erfolgreiche Produktion, March 2019 last visit, <https://www.it-production.com/allgemein/architekturmodell-fuer-fertigungs-it-das-internet-of-things-produktionsnah-strukturieren/>.
- [7] bbv Software Services: *Neues bbv Poster zum Thema Industrie 4.0*, March 2019 last visit, <http://www.bbv.ch/images/bbv/pdf/Publikationen/BBV-Poster-Uebersicht-Industrie-4.0.pdf>.
- [8] B. Tjahjono, C. Esplugues, E. Ares, G. Pelaez: *What does Industry 4.0 mean to Supply Chain?*, Manufacturing Engineering Society International Conference 2017, MESIC 2017, Vigo (Pontevedra), Spain, 28-30 June 2017.
- [9] FIR an der RWTH Aachen : *RTLS 2.0 - ERP-Systeme mittels Echtzeitlokalisierung berührungslos bedienen*, March 2019 last visit, <https://www.youtube.com/watch?v=0QQ-77LkVoM>.
- [10] Apps Run The World, Apps Research and Buyer Insight : *Top 10 ERP Software Vendors and Market Forecast 2017-2022*, March 2019 last visit, <https://www.appsruntheworld.com/about-us/>.
- [11] Björn Johansson, Frantisek Sudzina, : *Choosing Open Source ERP Systems: What Reasons Are There For Doing So?*, Copenhagen Business School, Center for Applied ICT, Frederiksberg, Denmark June 2009, https://www.researchgate.net/publication/220724903_Choosing_Open_Source_ERP_Systems_What_Reasons_Are_There_For_Doing_So.

- [12] Vittorio Gianni Fougatsaro : *A Study of Open Source ERP Systems*, Thesis for the Master's degree in Business Administration, School of Management Blekinge Institute of Technology, Spring 2009, <http://www.diva-portal.org/smash/get/diva2:832902/FULLTEXT01.pdf>.
- [13] Panorama Consulting Solutions : *2018 Top 12 Manufacturing ERP Systems Report*, March 2019 last visit, <https://cdn2.hubspot.net/hubfs/4439340/Top-12-Manufacturing-ERP-Systems-5.pdf>.
- [14] Soobia Saeed, Asadullah Shaikh, Syed Mehmood Raza Naqvi : *Assurance due to the Usage of Two ERP Methods: Microsoft Dynamics AX and SAP*, Mehran University Research Journal of Engineering and Technology, Mehran University of Engineering and Technology, Jamshoro, Pakistan, 2018, 37 (2), pp.337 - 350. https://www.researchgate.net/publication/324144493_Assurance_due_to_the_Usage_of_Two_ERP_Methods_Microsoft_Dynamics_AX_and_SAP.
- [15] Pilkington, M.: *Blockchain Technology: Principles and Applications*. 2015. Research Handbook On Digital Transformations.
- [16] Antonopoulos, A. M.: *Consensus algorithms, blockchain technology and bitcoin*, https://www.youtube.com/watch?v=fw3WkySh_Ho. 2016.
- [17] Kshetri, N.: *1 Blockchain's roles in meeting key supply chain management objectives*. International Journal of Information Management, Vol. 39, 2018, pp. 80-89, ISSN 0268-4012, URL: "http://www.sciencedirect.com/science/article/pii/S0268401217305248"
- [18] Linke D., Strahringer S.: *Integration einer Blockchain in ein ERP-System für den Procure-to-Pay-Prozess: Prototypische Realisierung mit SAP S/4Hana und Hyperledger Fabric am Beispiel der Daimler AG*. HMD Praxis der Wirtschaftsinformatik, Vol. 55, Issue 6, 2018, pp. 1341-1359.
- [19] SAP, 2019, <https://www.sap.com/swiss/products/leonardo/blockchain.html>
- [20] Jayavardhana Gubbi, R. B.: *Internet of Things (IoT): A vision, architectural elements, and future directions*, 2013, Australia, Elsevier.
- [21] M. Aabid A Majeed, T. D.: *Internet of Things (IoT) Embedded Future Supply Chains for Industry 4.0*, 2017, United Kingdom, Sheffield Hallam University.
- [22] Moutaz Haddaraab, A. E.: *The Readiness of ERP Systems for the Factory of the Future*, 2015, Sweden, Norway, Elsevier.
- [23] SAP: *Internet der Dinge*. Retrieved from <https://www.sap.com/swiss/products/leonardo/iot.html>, april, 2019
- [24] Somayya Madakam, R. R.: *Internet of Things (IoT): A Literature*, 2015, Mumbai, India: Scientific Research Publishing.
- [25] Fei Tao, Y. C.: *CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System.*, 2014, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 10, NO. 2.
- [26] Ming-Ling Chuang, W. H.: *An empirical study of enterprise resource management systems implementation From ERP to RFID.*, 2008, Connecticut, USA, Emerald Group Publishing Limited.

- [27] Shouqin Zhou, W. L.: *An RFID-based remote monitoring system for enterprise internal production management.*, 2006, London, UK, Springer-Verlag
- [28] Oracle: *Oracle Internet of Things (IoT) SaaS Applications.* Retrieved from <https://www.oracle.com/internet-of-things/saas-applications.html>, june, 2019

Chapter 4

On the Legal Validity of Blockchain-based Smart Contracts

Fabian Kueffer, Kilian Werder, Pascal Kiechl and Lukas Mueller

Blockchain-based smart contracts are a very polarizing topic, since they are a new technology that attempts to digitize contracts. Naturally, the legal validity of smart contracts is being called into question. This paper gives insight into how the legality of smart contracts is being handled in different countries, a comparison to traditional contracts is drawn, and additionally, examples of how smart contracts can be applied are presented. In conclusion, many legal decisions have not been taken yet. However, the growth in popularity of smart contracts is not going unnoticed and the legislature is attempting to close the gaps.

Contents

4.1	Introduction	101
4.2	Blockchain & Smart Contracts	101
4.2.1	Blockchain	101
4.2.2	Ethereum	103
4.2.3	Smart Contracts	104
4.3	Legal Technicalities	106
4.3.1	Real Contracts	106
4.3.2	Legality of Smart Contracts	107
4.3.3	Current Debates	108
4.4	Examples for Smart Contracts	109
4.4.1	E-auction: Bidding	109
4.4.2	Will and Testament	110
4.4.3	Service Level Agreement	111
4.4.4	Real-World Examples	112
4.4.5	E-Voting on Blockchain in Italy	114
4.5	Exemplifying Chances and Risks concerning a potential Adaptation of Smart Contracts	115
4.5.1	Chances	115
4.5.2	Risks	116
4.6	Trust Issues	117
4.6.1	The Role of Trust in the Context of Real Contracts	118
4.6.2	The Role of Trust with regards to Smart Contracts	118
4.6.3	Trust regarding Oracles	118
4.7	Conclusion	119

4.1 Introduction

Blockchain-based smart contracts are one of the most contemporary topics concerning the blockchain technology and are seen by the industry and academia as the most transformative blockchain application at the current moment [1, 2].

The concept of smart contracts is not new and has existed before the advent of the blockchain-technology [3]. However, for the sake of readability, in the rest of this paper, unless stated differently, let smart contract refer to blockchain-based smart contracts.

According to some smart contract proponents, the economic world might benefit from the adaptation of smart contracts. Common points amongst the benefits of smart contracts include aspects such as security, based on the immutability of the blockchain, efficiency and reduced costs, due to the elimination of intermediaries, and transparency, as the terms and conditions of any smart contract are visible, in case of a public blockchain, to all involved parties [4, 5, 6].

However, if one looks closer at some of the claims that are being made, one cannot help but notice, that smart contracts have some limitations [7, 8]. In addition to that, due to their novelty, their legal validity is, for the most part, not given [2, 9].

This legal aspect of the smart contracts is the main topic of the presented paper. Thus, it presents examples of the laws concerning smart contracts, legal use cases of smart contracts and potential applications are discussed. However, to be able to discuss these topics, the reader needs a basic understanding of smart contracts, their benefits and their limitations.

Hence, let Section 4.2 serve as an introduction to the concept and technical workings of a smart contract. Since the smart contracts in question inherit many of their properties from the blockchain technology, that section will also cover the basic functionality of the blockchain. In Section 4.3, we will delve into the legal technicalities of smart contracts. Section 4.4 will present the reader with a plethora of both real, as well as potential applications of smart contracts, whilst Section 4.5 evaluates the chances and risks coming with the adaptation of smart contracts. In Section 4.6, the issue of trust in real contracts versus trust in smart contracts is explored and finally, in Section 4.7, a conclusion is drawn.

4.2 Blockchain & Smart Contracts

As hinted on in Section 2.1, it is imperative that the reader has a fundamental understanding of the technical aspects of smart contracts, and of the blockchain technology, since the smart contracts this paper deals with run on blockchain. Thus, this Section first establishes a high-level understanding of how the blockchain operates, takes a more in-depth look at the Ethereum platform, which was designed with smart contracts in mind and lastly, the technical workings of smart contracts themselves are elaborated.

4.2.1 Blockchain

Ever since the world was introduced to the blockchain technology via Bitcoin in 2008, the blockchain technology has become more prominent and mature, as it has inspired numerous other cryptocurrencies and spawned ideas, that, if realizable, have the potential to influence every industry [10, 11].

One of those ideas is the smart contract, or more precisely, the smart contract running on blockchain technology. As already hinted at in Section 4.1, many of the properties of smart contracts stem directly from them running on the blockchain, making a basic understanding of blockchain essential.

At its core, a blockchain is a distributed system of computers, called nodes, that maintains a distributed ledger [12, 13]. That ledger acts as an immutable record of all transactions and events that have occurred on the blockchain [12]. Each transaction is verified by the network of nodes and is only stored in the ledger once the majority of the nodes reach consensus that the transaction is valid [12].

As such, the properties of blockchain that are particularly impactful with regards to smart contracts are:

- **Immutability:** once an entry is made to the blockchain, it cannot be altered, unless the majority of the nodes reaches a new consensus about the record [14].
- **Decentralization:** the records are not kept in a centralized location, but each node has the full records available [13, 15].
- **Transparency:** the fact that each node has its own copy of the full records combined with the consensus mechanism allows all nodes to verify the records whilst guaranteeing its integrity, thus creating transparency about all transactions within the blockchain [14, 15].

The procedure by which transactions are stored on the blockchain includes the following steps:

- A transaction is started on a single node, which, in addition to starting the transaction, also signs said transaction with its users private key [10, 16].
- The transaction is then propagated to peers via the usage of the Gossip protocol, a flooding protocol that distributes any new information in the network, in this case the new transaction, to all connected nodes [10, 17]. Every node validates the transaction before forwarding it to the next node[16]. Should a node come to the conclusion that the transaction is not valid, it discards the transaction and does not forward it any further [16].
- If the transaction is found to be valid, it is stored inside a so called candidate block, together with other transactions that have been validated within the same, predetermined amount of time [10, 16]. This process of creating a candidate block is called mining [16].
- The mining node for this particular candidate block then propagates said block across the network of nodes [10, 16]. Each node checks whether all the transactions contained within are valid and whether the received candidate block references the hash of the previous block [16]. If the candidate block fulfills both criteria, it is appended to the blockchain and the block receives its first confirmation [10, 16]. Should one of the specified creteria not be met by the candidate block, it is discarded [16].
- Upon the creation of each new block, the transactions get reconfirmed. For example in the Bitcoin network, a transaction needs to be confirmed six times to be seen as final [10].

It is worth noting that the nodes also need to reach a consensus on the order of the transactions when evaluating a candidate block, as otherwise, the individual copies of the blockchain, stored on the different nodes, can start to differ from node to node, thus ultimately no longer presenting a single version of the truth [10, 16].

The way this is achieved is by the use of a so called distributed consensus mechanism [10, 16]. The problem is that this mechanism needs a way to prevent a single entity

from being able to take control of the network [16]. Imagine a blockchain network where each node validating a candidate block gets to vote on the order of transactions in that block [16]. Such a network could easily fall prey to a so called Sybil attack, where one entity creates multiple identities on the network, thus receives the right to vote multiple times and therefore can disproportionately influence the decisions of the network from a minority position [16, 18].

There are different types of distributed consensus mechanism, used by different blockchains, such as the Proof of Work mechanism used in the Bitcoin network, utilizing the fact that a single entity has limited computational resources by making mining computationally expensive and thus rendering the creation of multiple identities useless [16]. Another distributed consensus mechanism is the Proof of Stake, where the chance to mine the next block for any given node is influenced by the amount of the networks' cryptocurrency held by that node, as well as how long the coins have been held [10, 16]. It is worth noting that this is less of an issue for private blockchain networks, as the risk of a Sybil attack is not present, due to users being whitelisted [16].

4.2.2 Ethereum

One of the most famous blockchains is Ethereum, which together with Bitcoin holds the majority of the cryptocurrency market and likewise, is based on a public blockchain [19, 20, 21]. Ethereum was designed with smart contracts in mind and therefore allows developers to create their own consensus-based applications, thus, being the underlying technology of several smart contract based applications [22]. Examples of such applications are presented in Section 4.4, which could rely on Ethereum smart contracts for their implementation.

Of interest to smart contract platforms are features that include the properties of Determinism, Halting and Isolation [23]:

- Determinism leads to a given input always having the same output, which is important for smart contracts, since a contract should be consistent [23].
- The Halting problem is also serious, since we cannot know, if a given program will halt, but this can be avoided altogether in various ways, like Turing incompleteness, timers, or step and fee meters, which make sure, in one way or another, that the program will always halt [23].

This problem was solved in Ethereum through an implementation of the concept of processing cost, in Ethereum called gas, thus every computing step costs a predefined amount of gas and when it exceeds the limit, the computation halts, if the execution had not yet halted [24, 25].

- The third property is Isolation, which in the case of Ethereum is applied through the smart contract code running on the nodes in virtual machines, and thus avoiding the possibility that the system is being tampered with, or viruses [23].

The most prominent high level language for Ethereum is Solidity that compiles the code into bytecode, its syntax is similar to JavaScript [21, 25]. In contrast to Bitcoin, Ethereum provides Turing-completeness in its programming language [22]. Turing completeness is defined as:

“A programming language that is Turing complete is theoretically capable of expressing all tasks accomplishable by computers; nearly all programming languages are Turing complete if the limitations of finite memory are ignored [26].”

Some smart contracts platforms allow the smart contracts to be written in a Turing complete language, like Ethereum, Corda or Hyperledger-Fabric [27, 28]. Although smart

contracts do not require Turing completeness, it allows for additional functionality such as conditional repetition or conditional jump, such as while, for or goto, or infinite loops [26, 29].

However, Taylor Rolfe, 2019, argues that “Turing completeness carries an unnecessary and burdensome attack surface” [26], since at the time of his article there were 16 known attack vectors on Solidity, and Turing completeness being an enabler for these security risks [30]. Another potential drawback of Turing completeness is, since it allows for more complex applications, that it can lead to a huge overhead in the blockchain, which is not desirable, since data gets stored on the blockchain [31].

Naturally, more data on the blockchain or complex computations lead to an increase of processing time for the miners, who might not be incentivized to lend a huge portion of their processing capabilities for executing a smart contract, which in turn can lead to inaccessibility of the smart contract service, due to a rejection of mining [31].

On the contrary, Turing *incompleteness* solves the halting problem, and reduces security concerns [23, 30]. Additionally, some Blockchain developers argue, that real world use-cases of smart contracts with Ethereum could also be implemented with Bitcoin’s *Turing-incomplete* programming language, and that Bitcoin’s more conservative approach to protocol changes is an advantage, since changes tend to happen more slowly and thus are more predictable [32].

Further, in 2016, Ethereum appeared infamously in the news, in conjunction with the “The DAO incident” [33]. The “DAO” was a Decentralized Autonomous Organization, an organization that did not need a governing body due to the blockchain, which was made possible through smart contracts on Ethereum [33]. Their idea was to operate a venture capital fund for cryptocurrencies and to take advantage of transparency and of the lack of authority needed in Blockchain technology [33]. The “DAO” gained huge popularity and over \$250 million were invested, although shortly after, a recursive call bug was found [34]. This exploit was used to successfully drain the cryptocurrency Ether with an equivalent of \$70 million at the time, which then lead to multiple discussions of essentially rolling back what had happened, and to finally hardforking Ethereum Classic out of the Ethereum blockchain, due to the notion of the rejection of immutability in blockchains [33, 35].

With potentially millions of dollars at risk, it is easy to see, that smart contract security is of the utmost importance, and auditing as well as code analyzing projects have emerged to make sure that bugs and attack vectors are minimized in smart contracts, which will be something to look out for in the near future [36].

4.2.3 Smart Contracts

An example of a smart contract platform was presented in Section 4.2.2, Ethereum, but naturally there are also other platforms, like Cardano, Neo, Hyperledger Fabric, or others [23].

Though, going through the history of Blockchain technology quickly reveals, that the idea behind a smart contract is much older, and actually dates back to 1994, when Nick Szabo wrote:

“A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions [3].”

He then went on to formalize this idea in 1996, and wrote that the idea behind it is, that the contracts should be robust against vandalism and breach, although ultimately, there was no ideal platform then, that could be used to implement this idea [37, 38].

However, with the rise of Blockchain technology, this idea found an ideal platform to be substantiated on, and many distributed ledgers like Ethereum, IBM’s Hyperledger Fabric, or Corda can be used today to execute smart contracts, and many smart contracts are being developed today [39, 38].

In a simple and theoretical example, Figure 4.1, Bob wants to sell his car via a smart contract and Alice wants to buy a car. Bob defines the terms of the contract and goes on to sign it. In this example, a smart lock would be used, that has access to this very smart contract, and as such, the contract can unlock the car if a set amount was transferred to the contract. Later on, Alice, who is searching for a car on the internet, finds Bob's car, and is immediately interested. From her blockchain address she then goes on to send the amount that Bob wants for his car to Bob's blockchain address. The nodes on the blockchain network verify the transaction, and in the case that the transaction went through, Bob receives the money for the car, Alice would be set as the new owner of the car in the blockchain, and would be able to unlock the smart lock device with her key.

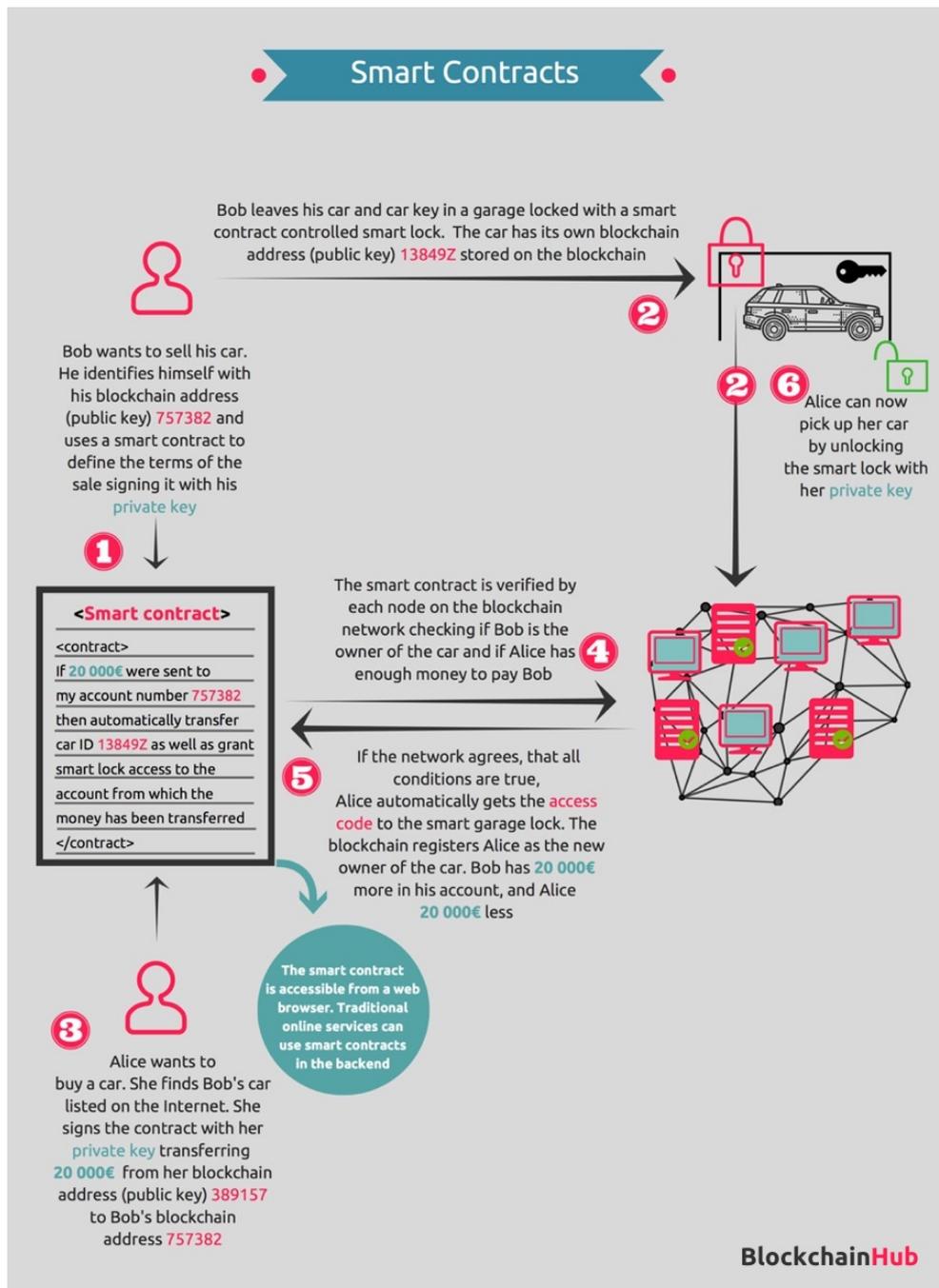


Figure 4.1: A simple and theoretical smart contract example [40]

It is noticeable from this simple example Figure 4.1, that no trusted third party was needed, Bob and Alice interacted with the blockchain via a smart contract and they did

not need a trusted third party like a notary to process the transaction and ownership of the car. Section 2.4 goes on to explore examples in more depth.

Technically, a smart contract is identified by a specific address on the blockchain, and users can invoke, and interact with it by sending a transaction to this address, which will then get checked by the smart contract [41]. If the transaction is successful and the contract accepts its invocation, then the mining nodes, those that take part of the network, will all execute the smart contract and decide on the output [41].

However, in the case of blockchains that use a coin, if one wants to invoke a transaction, one must own a coin appropriate to the blockchain, for instance for Ethereum, its coin “ETH” is needed [42].

Therefore, a smart contract is essentially code, a program, that runs on a blockchain and executes predefined orders based on predefined criteria, and cannot be altered afterwards or during execution, and its execution is checked for its correctness through the blockchain’s consensus protocol [4, 43]. This means, that the contract can automatically trigger its execution when predefined events happen, as will become clear in Section 4.4, where examples for such smart contracts will be shown [41].

Consequently, the idea behind smart contracts combined with their implementations with Blockchain technology, lead to the fact, that smart contracts become the enforcer of contracts, that are built into code, they remove dependencies on intermediaries and offer cost-effectiveness, which sparks heavy enthusiasm [38, 43]. Proponents of smart contracts illustrate accuracy, transparency, speed, security, trust, saving and even paper free as apparent advantages, and a lack of lawyers, witnesses, banks, or other intermediaries is seen as an advantage of smart contracts, compared to traditional contracts [4]. However, as remains to be seen in Section 4.3, the important question arises about their legal validity.

4.3 Legal Technicalities

To take advantage of smart contracts in our everyday life it is necessary, that they are legally recognized. Since smart contract came up again with the blockchain hype, the law in most countries lags behind the technical development. This part tries to focus on the existing law and the current debates in different countries and to give an overview over the legal situation.

4.3.1 Real Contracts

To compare smart contracts to real, paper-based contracts, we first need to build up a basic understanding of contracts.

A contract is a declaration of intent to determine legal or economic circumstances. A contract can be done between two or more contractual partners.

To make a valid contract, some requirements have to be fulfilled:

- All parties must be in agreement (after an offer has been made by one party and accepted by the other).
- A value must be exchanged (cash, services, goods, promise) for something else of value [44].

4.3.1.1 Types of Contracts

A real contracts is the most common understanding of a contract. It is mostly written on paper as a medium and needs to be signed by hand. The contract is fulfilled when all

contractual partners confirmed the contract. It's possible that a neutral third party (for instance a notary) also needs to sign the contract. This depends to the the construction of the contract. A real contract can also be valid if it is not written but spoken. But in that case the contractual partners must agree directly after the speech. An agreement at a later point of time is not possible.

The E-Contract is pretty similar to the real contract. The e-contract uses a digital medium to state the contract conditions. This is often used for software license agreements and e-commerce.

The smart contract uses the blockchain as medium and the fulfillment is generated by a computer. It can be used for everything that we can build a digital condition for. In real world examples we therefore need trustworthy sources for real world data (e.g. sport game results). Neutral third parties are no longer needed in smart contracts.

4.3.2 Legality of Smart Contracts

The following are examples of how different jurisdictions approach the topic of smart contracts in terms of their legality.

4.3.2.1 Switzerland

A Federal Council report released on 14 December 2018 discusses distributed ledger technology and blockchain in Switzerland. Smart contracts are also part of this report:

“Contrary to what its name suggests, a Smart Contract, as the doctrine largely agrees, is not a contract in the sense of the Swiss Code of Obligations, but rather a computer “technology” for contract execution.[45]”

Therefore Switzerland does not legally accept smart contracts. Smart contracts are not enforceable. There are two main reasons. The first reason is the following:

”Each party expresses an intent and the system serves as an intermediary. [...] Therefore, although the computer system plays an important role in the contract formation process, it is not a contracting party. According to prevailing doctrine, a party cannot conclude a contract solely with the computer system, as this does not have a legal personality within the meaning of the Civil Code.”

In other words, as a human being I cannot build a valid contract with a computer system, because the computer system is not a legal personality. The second reason why smart contracts are not legally even to real contracts is the liability:

”In the event of poor contract execution, therefore, the question of liability arises, e.g. liability for programming errors or machine errors despite correct programming.”

The question of liability is one of the key question, that has to be answered to legally accept smart contracts [46].

4.3.2.2 Arizona

In April 2017 the Arizona House of Representatives passed a bill from Jeff Weniger. The bill legalized blockchain signatures and recognized the enforceability of smart contracts:

”Smart contracts may exist in commerce. A contract relating to a transaction may not be denied legal effect, validity or enforceability solely because the contract contains a smart contract term.”

This means that the legislation is recognizing the legal validity of smart contracts. Smart contracts are equal to real contracts before the law according to that bill. The bill does not contain any information about liability. So this question remains open at this point [47].

4.3.2.3 Italy

Italy has the aim to be the first country worldwide which gives a legal value to technologies that are completely based on distributed ledgers such as blockchain technology and smart contracts [48]. Italy takes a lot of effort now to push this idea. First step was the green light from the Chamber at the DI Simplification. From then on, the blockchain-based technologies and the definitions of smart contract will be legally functional in Italy. This is the first time in the world a nation recognizes the innovation and maybe it has a wide-ranging turbulent effect [48].

On January 23, 2019 the declaration was released to the Senate and Parliament. One day later, the documents went to the Simplification decree. This was the start of the hole process. Currently, the decree has been fired already and the next step in the House is anticipated. After that the provision will be changed into law if it will be endorsed. The Agency for Digital Italy will define in three months the technical standards which has an influence of the provision, in order to have an affectual legal value. So, there are still a lot of processes and work in progress [48].

At the moment, the Chamber's Budget Service dossier precisely says [48]:

”This last rule establishes that the temporary legal validation cannot be denied the legal effects and the admissibility as evidence in court proceedings for the sole reason of its electronic form or because it does not meet the requirements of qualified electronic temporary validation. It is also envisaged that a qualified electronic time validation enjoys the presumption of accuracy of the date and time indicating and integrity of the data to which this date and time are associated. Finally, it is established that an electronic time validation issued in a Member State is recognized as a qualified electronic time validation in all Member States [48].”

Italy did a lot of work in the last one and a half years for this specific technology. It seems that Italy is convinced that the blockchain technology connected with smart contracts gives so many possibilities in the future. Andrea Bianconi, an international business lawyer, is persuaded that the amendment will make Italy to the first country in the world which set a smart contract equal to a written traditional contract [48].

4.3.3 Current Debates

- US state Connecticut and Ohio seek to legalize smart contracts, which means that they passed a bill, but from the article it follows that it does not seem likely that the bill will be passed [49, 50].
- UK Law Commission is investigating in the legality of blockchain based smart contracts, they also mention that there are questions about data protection law (GDPR) [51].

4.4 Examples for Smart Contracts

This section of the paper will focus on the use of smart contracts in the real world. It will show multiple examples to see how to use smart contracts and to express the possibilities of it. Of course, a lot of the examples are hypothetical and only a few smart contracts are really in use today. Although the ideas behind are quite interesting. First, we will present some theoretical examples for using a smart contract for a bidding in E-Auction, a will and Service Level Agreements. These are not in use currently but the idea behind it have a lot of potential in the future. Afterwards we go in some presently used smart contract projects.

4.4.1 E-auction: Bidding

Today a bidding system is a very common platform. A lot of people daily use such a platform like eBay, Ricardo or Yahoo to buy and sell products [52]. As such platforms were created the main advantage was to reduce the transmission cost and makes it much easier to integrate between the seller and the buyer [52]. All these platforms work with a middle party, the platform provider [52].

One advantage of smart contracts is to have no intermediate party [52]. Like in this case we need no broker nor such a platform [52]. The picture 4.2 shows the different roles in a e-auction. The idea is to save money and reduce the impact of a third party[52]. As a result, the dependency on a third party will also be minimized [52].



Figure 4.2: Roles in a e-auction [52]

Then today with the bidding service provider there are two essential problems [52]. First, we have the intermediary party which increases the transmission costs, like eBay will be paid for its service [52]. Further, the service provider must save the whole data of the bidding [52]. It means all information of the bidding, of the buyers and of the auctioneers are saved on a server, which may have privacy and security problems [52].

To investigate the other critical point, we need the understanding of a sealed bid. A sealed bid is an auction form in which each buyer can just hand in one closed envelop with the amount of his bidding. So, no one knows the other bids and there is no evidence how much the highest bid now is. The winner is the buyer with the biggest price, and it will be announced after the deadline. In case of a sealed bid, the third party is quite important because it handles all given bids and we must trust it, that it doesn't put out any information [52].

The idea now is to load a smart contract in the decentralized system of the blockchain [52]. Then each interested buyer can call this contract without an intermediate broker and can bid [52]. Hence, an E-auction system with a smart contract must satisfy the following requirements such that a successful use is guaranteed [52]:

- The identities of the bidders must be anonymous to everyone.
- It should be possible to check if a bid is correct, complete and a given seal bid should not be modifiable.

- It should be not possible to bid in another name instead of his/her own.
- A regression of a given bidding is permitted.
- It must be provable that the winner has get the product.
- The seller should get the money from the winner and not from the other bidders.
- The offer must be delivered in time.
- For fair transaction the smart contract should check the delivering and payment.
- Before the end of the auction a sealed envelope must be private, and nobody should be able to read it.
- On must define a fair solution if two times the same price is bidden.
- Bidders must store their bid in the smart contract.

The smart contract can be implemented and saved on the Ethereum platform for example. Ethereum is one of the most used blockchain for smart contracts because it supports a lot of tools and is Turing-complete [29]. The smart contract should work if a bidding comes in or the deadline is achieved. To describe such a smart contract on can use Solidity, Serpent, LLL or EtherScript. JSON format is an interface to send the contract to all nodes in the blockchain [52].

Then use the Ethereum Wallet with Watch Contract to invite other people to bid. By end of the deadline, the smart contract will automatically open all envelopes and determinates the winner. Each bidder must deposit the amount of his bid. If he wins, the amount will be payed to the seller, otherwise the smart contract returns the bid [52].

Here two problems occur. First, the use of a fixed cryptocurrency is necessaire that we can guarantee the handling via a blockchain. Second, the bid will be paid to the seller as soon as the arrival of the product is confirmed. But if the product is a widget and must be delivered by the post how can the arrival be detected? Of course, the information from the post office can be used, but this are information out of the blockchain, and we have again other human interaction involved which we want originally to avoid [52]. Under this circumstances, the only option is to trust the post system.

So just for this hypothetical example it's hard to find one perfect solution. Unfortunately, perfection and complete automation is not possible yet. In this case the only option is to make trade-offs.

4.4.2 Will and Testament

The next instance is maybe a bit more important and more personal than the bidding system. The concept is to represent a will through a smart contract. Normally, a testament is written on a paper in which a person can express what happens with his or her estate after the death. Unfortunately, such a paper is very vulnerable and can be tampered very easy. Furthermore, the execution of the will needs a lot of time by the government [53]. Furthermore, there is a lot of paper work and so many human interactions needed. More months waiting is of course not rare. Another option is to engage a lawyer to deal with the legal issues and let him write a formal will. The lawyer can also lock up the paper until it is needed. But again, a third party is involved, which needs a lot of time and money. The idea is to use a smart contract to reduce all these factors and get a faster execution.

There is a theoretical idea how such a "smart will" system can look like and how it would work. For this are several assumptions to the environment required. These are important and indispensable [53]:

- Everyone needs a blockchain account, for example Bitcoin, and the account details should be available through the government.
- This list of accounts must be published public by the government such that people can search other people by providing his details. This is useful to define an heir.

Obviously, these requirements are not fulfilled today which means that, as already mentioned, this is again just a hypothetical idea.

The general idea is to have a smart will program as a smart contract on the blockchain which you can access via a web browser [53]. It should be very easy to log in and very intuitive to create a formally correct will. The program has 3 main parts: "Create will", "Update will" and "Probate will" [53].

Create will: After the creation of the testament, it will be submitted and uploaded to the blockchain as a smart contract. With this step the testament is safe and tamper-proof. In addition, the smart contract has a death flag, which will be set only after checking the official register of deaths [53].

Update: After checking that that right node will change this smart will, it will get the information of the testament out of the blockchain and can update it. After that the user will submit the will again, it will be checked once more that the user is permitted to update the will and then the new smart contract will be written to the blockchain and the old one will be invalid [53].

Probate: Any person can probate a will with putting the public key and a valid death certificate id to the smart contract. The system will check if the death certificate is valid. Then all transaction will be performed within a few minutes and the testament will be displayed to the user. This procedure is only once possible. If another person probates the will it will just show the testament and a notification that all transactions are successfully performed [53].

As assumed, the government has access to the blockchain and so he can see all transaction and, of course, calculate the taxes [53]. So, one further idea is to include an automatically taxes system right into the smart contract. This will improve and accelerate again the execution of a will including the payment of the taxes, which then can be done automatically.

The big advantage of that system is the increasing of the speed and the safety of the testament [53]. The idea is to construct a very easy graphical user interface to create a will and so to reduce the need of a lawyer. To enlarge this idea, we can make a blockchain based property system to register property owner and handle also estate with a property [53]. As a conclusion, this idea for a smart will is very interesting and has a lot of advantages and good points. Nevertheless, it's today not possible to use such a system because the assumptions are not satisfied.

4.4.3 Service Level Agreement

To finish this first part of examples we will have a short look at the service level agreements (SLA). Service Level Agreement describes the obligation that service providers must deliver a predefined service [54]. Like a cloud computing server must have a 99.99% of availability. Or the throughput has to more than 1Mbps otherwise the client will get a payback for the occurred damage.

To use such Service Level Agreement multiple systems are in use but also a lot of manual effort and interaction is needed. This manual impact produces a lot of static behaviour and prone to errors [54]. If the provider violates this agreement it provides a lot of work to proof the violation and transfer the payback [54]. So, we have again so many transmission cost and need to trust a third party, most the service provider itself which is also a dependency problem.

In that case a smart contract is a very good idea to reduce these problems and the impact of the third-party. For this context we get two crucial aspects: the guarantee of contract enforcement and tampered-free of the data [54]. With smart contracts and blockchains we have the possibility to find a solution for these problems and on the other hand assures the service provider that the client pays the subscription fee [54].

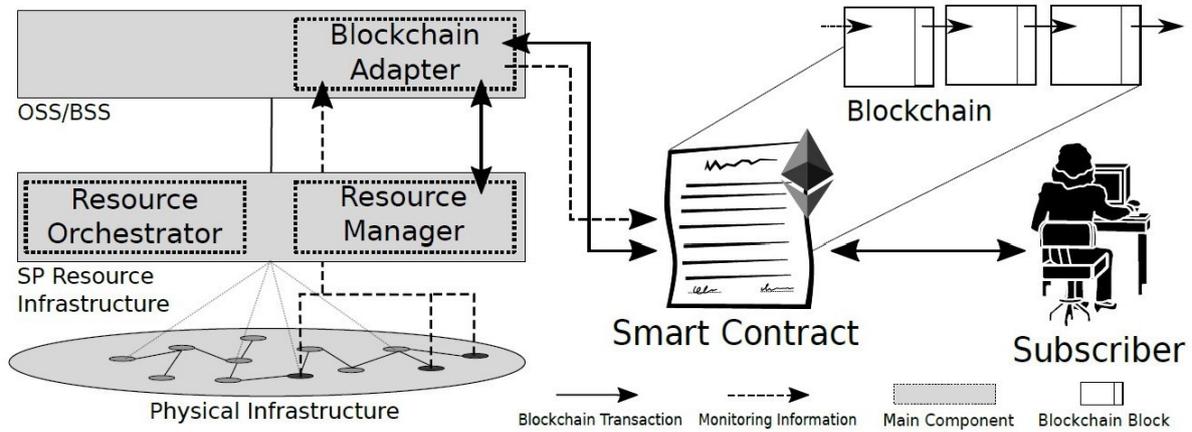


Figure 4.3: Smart Contract-based SLA Approach [54]

Figure 4.3 depicts the proposed approach. The service provider must develop a smart contract for the according service level agreement [54]. For this he will use the "Blockchain Adapter", which contains other information as: the agreement as a code, details about the checked resourced and the duration and fee for the service [54]. The client can subscribe to the smart contract [54]. The service will start with the payment of the fee for the service by client through the smart contract.

With this paying he will accept the terms and booked the service. Right after that the "Resource Manager" will deploy the requested resources in the physical infrastructure. The resource manager can communicate with the smart contract such that the smart contract knows the physical status. With this information the smart contract can detect all undercutting's of predefined services and so automatically pay out the compensation [54].

Smart contracts in a context of service level agreements have a lot of capability to get commonly in used because in a service level agreement the conditions are strictly predefined. For this it is easier to write an algorithm. Nevertheless, until now there is no such smart contract available. But a lot of research is still in progress.

4.4.4 Real-World Examples

The examples presented on the past sections shows promising research ideas. However, they are still in progress and need a lot of development yet. They might be in use in the future. This section will present real world examples which are already in use.

4.4.4.1 Slock.It and Share&Charge

Slock.It changes the process of renting and sharing [55]. It strictly does these things via smart contracts. In this case it automates sharing, payments and rentals [55].

Slock.It created recently Share&Charge which use the smart contract technology of Slock.It to automate the payment to rent a vehicle charging station [55]. A further project of Slock.IT is the development of a Universal Sharing Network (USN). This should be an open source network where blockchain application modules can be deployed [55]. The

final product should allow third parties to add different products to the network to share them. The aim is that this process is very easy and available for all categories of clients [55].

4.4.4.2 Fizzy AXA

The insurance sector has been following the development of smart contracts as well [56]. The French airline developed a smart contract for flight insurances. Today it's very hard to get a compensation for a delayed flight although you have a travel insurance. Fizzy AXA will solve this problem with using smart contracts [55]. If your flight is more than 2 hours late and the information about that flight will be loaded by the airline to the application, then you will be informed by the system that you can choose a compensation type [55]. After you make your decision the money can directly send to your credit card. The idea behind this system is parametric insurance which means the user is not compensated for total loss but loss outside the traditional rules of insurance [55]. In this case here, the insurance executes if the flight has 2 hours or more delay [55]. Now, this insurance is only available for flights between the United States and Paris. But they have plans to expand the insurance. Of course, if one desires to use the insurance one must pay a subscription fee [55].

4.4.4.3 Etherparty

Etherparty is a startup from Vancouver [55]. It describes his own by "The smart contract creator" [55]. The business idea is to bring a tool to all user with which they can create smart contracts. The best is you need no prior smart contract programming knowledge. The program includes create peer-to-peer escrow contracts, wagering, token creation, supply chain management, real estate agreements, contractor agreements and more [55]. So, this product allows companies to adapt very easy existing process to a smart contract system. This means a lot of opportunities for clients. Especially for companies who want integrate smart contracts in their business but haven't the required knowledge to do this by their own.

4.4.4.4 Propy

Buying and selling real estates is an often topic according to smart contracts. Propy is one of the first companies which achieved to make a transaction concerning an estate through a smart contract [55]. They had their first transaction in September of 2017, when someone bought an apartment for \$60'000.

This new marketplace allows sellers to post their properties and buyers can search and negotiate the sale [55]. Buyer and seller participate in a smart contract which handle the process fair and legal to buy the property.

One example: The buyer can reserve a property by paying \$5'000 [55]. Because of smart contract the buyer will get that money automatically back if the seller refuses the offer. This means that all paperwork and signatures work through the smart contract and make it easy to buy or sell the estate wherever you are on the planet [55]. Further companies, like licensed money transmitter, are involved to check transactions and legitimize the process [55]. The certificates are still sent through the local government. This ensures also that every follows necessary and legal protocols [55].

4.4.4.5 Populous

Invoice financing companies take over the unpaid invoices [55]. This means they give the companies the money of unpaid invoices and gets paid then by the original debtor [55].

Populous takes this process and make it much more global and easier with the use of smart contracts in this context. One advantage of reducing the middle man and use a smart contract is speed [55]. The whole process is much faster than before. So, we can reduce the risk. Furthermore, we can reduce manual human errors and duplications of invoice financing [55]. Populous creates a platform on which once the invoice seller uploads the invoice and terms, buyers simply choose an invoice and provide the money. All other transactions execute automatically through a smart contract [55].

4.4.4.6 PolySwarm

PolySwarm is the first company that creates a decentralized threat intelligence marketplace, which runs seamlessly thanks to smart contracts [55]. Companies and ambassadors upload their bounty and they specify the payment for the person who uploads the correct assertion. In case that the assertion is determined to be accurate then through smart contract the reward will be released automatically [55]. Hence, it can be emphasised to incentivize innovation within cyber security, providing more accurate and effective threat protection [55].

4.4.5 E-Voting on Blockchain in Italy

All over the world a lot of society's resources are used to perform the voting process of civil right [57]. It's a lot of money. For example, Italy alone spent over 389 million Euros to process the 2013 parliamentary elections. For this reason, it's very interesting for countries to find an efficient way to process voting. Therefore, some nations do a lot of research to investigate the use of a blockchain-based smart contract system for e-voting in the future [57].

Voting is an important point for every country [57]. It demonstrates democracy and, in this sense, more freedom and rights for the population. For instance, Switzerland has direct democracy. Therefore, the Swiss population can vote their representatives but can also pounce referendum or set petition for a referendum. One sees that voting systems play an important role in the society and it is a critical point if something got tampered. Today there are a lot of traditional voting systems in use, like those now used in Italy [57]. Unfortunately, they are very expensive, old and of course can be manipulate. Further, there are problems such as double counting, forgery, fraud and many others [57].

Up to now, in traditional electronic voting systems there are computers where votes will be placed. Normally, these results will be sent to a central server, which then calculates the final result [57]. Although, in these systems are many security issues. For example, a hacker can listen to the communication and then interrupt or change the information which are sent.

For all these reasons some countries focus on the possibility to use a blockchain to reduce these problems [57]. Italy as well as other countries and cities like Zug in Switzerland focus on this solution. Maybe this is just the beginning and much more countries and companies will start research in that topic. Because the potential of this system is obvious, and some results are already available.

For example, Italy rolled out a distributed ledger technology-based e-voting system for inhabitants aboard to vote [57]. Italy takes much effort to push the development. For that, Italy raised a meeting of approximately 30 experts to discuss some fundamental questions. After that the government of Italy launched a technical discussion between different departments for the approval of guidelines for the blockchain based e-voting system [57].

On January 22, 2019, Italy validates distributed ledger technology transaction and smart contracts official as legal [57]. This is the first time that an Italian report contains

blockchain based technology and a legal definition of smart contract. Concrete, this means smart contracts will be treated equal to written contracts in future. More details about the law adaptations in Italy are described in Section 4.3.2.3. So, the way is free for smart contracts in combination with a e-vote system [57].

Until now, Italy does still a lot of research and developing for an e-voting system. They will call it "E-Vota". The system should base on a smart contract which runs on a blockchain. Although there is no final system released, Italy constructed a very good framework to realise this project. Especially because of the adaption of the law the government did at the beginning of this year.

4.5 Exemplifying Chances and Risks concerning a potential Adaptation of Smart Contracts

Given the aforementioned examples, this section further explores advantages and benefits that can be achieved through smart contracts over traditional contracts, but also describes some given disadvantages and risks.

4.5.1 Chances

This section gives insight into possible advantages and chances that can be gained through the use of smart contracts over traditional contracts, such as lower costs, security benefits and emerging new markets.

4.5.1.1 Lower Costs

Admittedly, hiring lawyers and solicitors is not a cheap undertaking, but on the contrary, they are rather expensive, and ideally, this is where smart contracts can vastly benefit over traditional contracts, since they do not require a trusted third party, since blockchain based smart contracts allow untrusted entities to transact with one another [42].

Additionally, smart contracts can also reduce transaction costs, since intermediaries are removed, but they are also able to simplify contract enforcement due to the lack of the need of courts, and potentially, law protection as well [7].

4.5.1.2 Security

Taken the decentralization of blockchain technology into account, one can also see that it lowers security concerns compared to traditional contracts, given that a single point, of failure is not at hand, but rather a network of decentralized, and distributed nodes [7, 42, 58].

In addition, smart contracts are self-enforcing, pre-defined events automatically trigger their execution, this obviates human intervention, for instance a refusal to pay or a potential noncompliance with the contract [7]. This goes hand in hand with their feature of being tamper-proof, the contract cannot be modified in any way, or being forced to stop, and does not take any external events into account, and thus, is incorruptible unlike traditional contracts [7].

However, as will be shown in Section 4.5.2, these "security advantages" are not perfect, they also introduce disadvantages by themselves.

4.5.1.3 New Markets

As was shown in Section 4.4 examples of how blockchain-based smart contracts can be, hypothetically *and* realistically, used. These are new ideas, which have emerged lately.

The use and rise in popularity of smart contracts has created new markets and new companies, which in turn has been sparking further development and research of smart contracts, from which the whole industry profits. A report from 2018 has predicted, that smart contract markets are to grow by a compound annual growth rate (CAGR) of 32% from the period of 2017 to 2023, and by then reach a global market of approximately \$300 million [59].

Additionally, even from security issues, which are shown in Section 4.4, new companies are emerging that are attempting to audit and to analyze smart contract code to find security issues [60].

4.5.2 Risks

Not only offer smart contracts advantages, but there are also currently profound drawbacks over traditional contracts, that hinder their legal validity, which will be explored in this section.

4.5.2.1 GDPR

One novel issue of blockchain technology is the General Data Protection Regulation (GDPR), which concerns itself with the data protection and regulations in the European Union (EU), which took effect in 2018 [61]. One of its most important implications is, that public blockchains are not compliant with the GDPR per se, since personal data can leave the EU, instinctively, since nodes can be located all over the world [62].

Besides, the regulations mention the erasure of data, though not in a concise manner, but this notion immediately contradicts the immutability of blockchains [61, 62].

To circumvent this issue, and more importantly, to make a blockchain GDPR compliant, one could implement a blockchain in such a way, that the blockchain would only contain references to personal data and the data would be then stored off-chain, effectively making the blockchain a “lookup-table” [62, 63].

This workaround would be GDPR compliant, but also comes with some negative effects, not only can data be erased, but it also decreases transparency and it is not immediately clear anymore who owns the data, even more so in case of *public* blockchains [62].

Further, the added complexity increases attack vectors through the additionally needed infrastructure changes [62]. One cannot help but notice, that these regulations severely limit the potential of blockchain applications.

4.5.2.2 Security and Code Correctness

Although the blockchain might be in most cases incorruptible (taking “sybil attacks” into consideration), the smart contract code might not be, as can be seen from the “DAO incident” (Section 4.2.2), since the smart contract code is often run on top of a blockchain, and is thus not subject to the security benefits of blockchains [42, 63].

Some possible solutions to establish smart contract code correctness have been presented in the literature, amongst them are *semi-automation*, to translate the contract into smart contract rules, general *writing guidelines* for correct smart contracts, and lastly, *formal verification techniques*, that validate and ensure correctness [42].

Moreover, the advantages of being *tamper-proof* in combination with being *self-enforcing* also have a negative side to them, while on the one hand they protect against malicious human interference, on the other hand they also remove the possibility to remove coding errors, and one can argue, that statistically seen, every program contains errors, thus, contract malfunctions are bound to happen [7, 64].

Further, as was shown in the examples in Section 2.4, smart contracts often rely on peripheral data, which come from various sources [54]. Naturally, it does not suffice that

the smart contracts are secure, if the external data can be tampered with. It must be ensured accordingly, that this data's integrity is held to the same standards as the smart contracts' are.

Additionally, in comparison to legal law, which allows contracts to be modified and or terminated under certain circumstances, such as *rescission by agreement*, *termination by right*, or *rescission by court* [42, 65]. Thus, the lack of flexibility with smart contracts is problematic from a legal point of view and a serious concern [66, 67].

To remedy this problem, a set of standards were successfully presented (for Ethereum-based smart contracts), that are able to modify and reform smart contracts under various terms and introduce flexibility to assimilate smart contracts to traditional contracts [65]. Even so, can these standards account for all possible scenarios that may occur?

4.5.2.3 Matching Agreement Intention

Further, even if coding errors can be avoided, there is still the problematic issue of ensuring, that the implemented contract matches the intention of the agreement [7, 68]. Misunderstanding the agreement, or even malice may occur, and in the common case that the parties are not code-savy, contract verification is even more so inaccessible [7]. One can see, that there is an apparent difficulty in writing *correct* smart contracts.

Ultimately, traditional contracts are not set in stone, they are dynamic, flexible and can be adapted to external events, words have different meanings depending on context, contain gaps, and most importantly, there is room for interpretation [7, 69]. Smart Contracts struggle in comparison, they lack the nuance of understanding the law, while complying to both contractual parties' agreement, and discrepancies can easily occur, even if both lawyers and programmers work together to mirror the contract [7, 69].

4.5.2.4 Archival Bonds

Moreover, it is possible, that after the *execution*, or *formation* of a contract, one party might raise issues with the contract [70].

The ISO 15489 archival definition of authenticity points out, that in order for a record to be an authentic record, it must provably contain what it purports to be, by the person who created it and also to be created or sent at the time it states [70]. This is problematic for smart contract, and the problem lies therein, that it is legally no easy task to prove, that a contract even exists, and even so, that a legally binding contract has been made at all [70].

To facilitate legal validity, Lemieux and Sporny propose to use the blockchain for archival principles, as then, the identity of documents as records can be used as evidence of transactions [70].

Darra Hofman notes, that without an archival bond, one can not know if a contract has been formed, since relations of a record can not be reconstructed, and one can not know, if acceptance of a contract was actually an acceptance, or a revoked offer, and it is not possible to prove admissibility for documentary evidence [70].

Ultimately, using archival bonds as a semantic layer in the blockchain and adding limitations and liability, as well as disclaimers of warranties to the smart contracts would clear legal problems and advance their validity [70].

4.6 Trust Issues

The issue of trust raises some interesting questions with regards to smart contracts, such as how the role of trust differs with regards to smart contracts versus with regards to

real contracts and what mechanisms exist for smart contracts to make data coming from external sources more trustworthy.

4.6.1 The Role of Trust in the Context of Real Contracts

In order to establish a real contract, as described in Section 4.3, with another party, whether that be a customer, a business partner or a service provider, a certain amount of trust in the other party and its ability to fulfill their part of the contract is required [71]. For some cases, previous interactions or experiences may be enough to verify the trustworthiness of the other party [13]. However, there are cases where not enough or no information at all is available to reach the required threshold of trust and consequently, the services of a third party, such as an auditor or a bank, are required to verify the trustworthiness of the other party or to ensure that any agreed-upon transactions are being performed properly [13, 71].

This leads to increased transaction-costs and time, in addition to once again needing trust in the chosen third party to perform their services properly [13, 71]. As such, any additional costs caused by having to hire third parties can be led back to uncertainty about the transaction or service, and thus, in part, back to trust [71].

Consequently, minimizing the degree by which trust is required to establish a contract without the involvement of third parties or even eliminating the need for trust in its entirety, leads to lower transaction-costs, if it means no longer requiring the services of aforementioned third parties [13, 71].

4.6.2 The Role of Trust with regards to Smart Contracts

This is where the so-called trustless or trust-free nature of smart contracts, thanks to them being based on the blockchain technology, comes into play. Trustless means that both parties can establish a contract together without needing to trust each other [13, 72, 73]. This does not mean that trust is altogether removed from the equation, instead trust is provided by the blockchain technology, due to its attributes highlighted in Section 2.2.1 [13, 16].

Immutability, the property that ensures that the content of a smart contract cannot be altered once deployed, unless the majority of the network's nodes reach a new consensus about that specific smart contract, means that the contract will be enforced exactly as it has been agreed upon by both parties [14]. In combination with the decentralized, peer-to-peer nature of the blockchain, and thus every node in the system having the full records available, transparency is created, therefore building trust [13, 14, 15].

4.6.3 Trust regarding Oracles

However, in cases where a smart contract needs to access data that comes from outside the blockchain, oracles, interfaces that allow a smart contract to access external data sources, come into play [10]. This leads to a problem where, whilst the blockchain and thus the smart contract itself is trustless, the integrity of the data delivered by the oracle, and based on which the smart contract decides how to or whether to execute its predefined behavior, is not necessarily guaranteed [10].

These oracles are trusted entities, which communicate with the smart contract by the means of a secure channel [10]. Whilst oracles can proof the authenticity of the delivered data's origin, as well as guarantee that the data has not been modified by the oracle itself, that still does not resolve the issue of trust, since the data provided by the source, even if it is a trusted source, still comes from a centralized entity [10].

There are different ways to increase trust in the data delivered by oracles, such as the oracle itself comparing the data from multiple sources, thus increasing the chance that, if multiple sources deliver the same data, said data is accurate [10].

Another method to increase trust in oracle-delivered data are decentralized oracles, which are based on some distributed mechanism, or alternatively have a different blockchain-network as data source [10].

Yet another approach, however only working for oracles where data gathered by physical devices is used, is to ensure the authenticity of the data via making the physical devices tamper-resistant, by including a mechanism that disables the device should any tampering attempts be made [10].

4.7 Conclusion

Undoubtedly, the increasing popularity of blockchain and, more recently, of smart contracts, has fuelled the discussions surrounding the legal viability of smart contracts. As such, various aspect of the debate have to be considered in order to draw an exhaustive conclusion.

Most importantly, one has to consider the *status quo* of the legal landscape. As it stands, in most jurisdictions, smart contracts are not considered to be legal documents in every case. However, there are cases, such as the one regarding Arizona, outlined in Section 4.3.2.2, where specific applications of smart contracts are granted legal validity [47].

Even so, smart contracts offer a range of advantages compared to traditional contracts, namely reduced costs, decreased waiting times until the contract is enforced, as well as increased security through the obviating of human errors, the tamper-resistance of the system and the property of self-enforcement [7].

Admittedly, with increasing complexity of a given contract, those benefits start to diminish, as the risk of human error is heightened through the increased difficulty in setting up the smart contract in such a way that it executes as intended. Additionally, the tamper-resistance combined with the property of self-enforcement can become problematic, should there be the desire to alter the content of the contract, even if all involved parties were to agree on the matter [7].

Moreover, as can be seen from the current *General Data Protection Regulation* situation, described in Section 4.5.2.1, regulation changes are always imminent, and it is deducible from the manner of how smart contracts were handled, that, at least at the moment, the legislature is not taking smart contract matters into account when introducing changes.

In addition to that, there is the matter of trust. As hinted at in Section 4.6.3, the main point of contention with regards to external data integrity for smart contracts are the oracles [10]. To reiterate, the oracle can guarantee that the data it delivers stems from the source and has, whilst stored in the oracle, not been modified, but there is no way for it to assert that the data has not been tampered with at the source or is faulty due to a malfunction. Consequently, the trustless nature of smart contracts does not necessarily result in trustworthy data.

For the time being, many possible use-case scenarios for smart contracts can be disregarded from the start. Nonetheless, smart contracts offer intriguing aspects that one might wish to explore, as such, working under the assumption that the legal validity of smart contracts is undergoing changes in ongoing debates, it might be beneficial to not regard the current situation as absolute and to continue investigating future applications for smart contracts. Ultimately, despite all of the uncertainties surrounding smart contract use, they have the potential to find their place in the market, as more research is being conducted and the legal situation keeps developing. Even if they may end up not replacing traditional contracts, as some proponents of smart contracts have envisioned, the insight gained

through aforementioned research is valuable nonetheless and might be applicable to other technologies as well.

Bibliography

- [1] M. Iansiti, K. R. Lakhani: The truth about blockchain; Harvard Business Review, (Vol. 95, No. 1), 2017, 118-127.
- [2] S. McLean, S. Deane-Johns: Demystifying blockchain and distributed ledger technology-hype or hero?; Computer Law Review International, (Vol. 17, No. 4), 2016, pp. 97-102.
- [3] Nick Szabo: Smart Contracts; [On-line]; <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html>; accessed April 14, 2019.
- [4] ChainTrade: 10 Advantages of using Smart Contracts; [On-line]; <https://medium.com/@ChainTrade/10-advantages-of-using-smart-contracts-bc29c508691a>; accessed April 14, 2019.
- [5] Smart Contracts Explained: What are the Benefits of Smart Contracts?; [On-line]; <https://smartereum.com/8248/smart-contracts-explained-what-are-the-benefits-of-smart-contracts/>; accessed April 18, 2019.
- [6] Smart contracts work to counter possible human error; [On-line]; <https://www.bizjournals.com/bizjournals/how-to/growth-strategies/2017/09/smart-contracts-work-to-counter-possible-human.html>; accessed April 18, 2019.
- [7] E. Mik. Smart Contracts: Terminology, Technical Limitations and Real World Complexity. Law, Innovation and Technology, (Vol. 9, No. 2), August, 2017, pp. 269-300.
- [8] K. O'hara: Smart contracts-dumb idea; IEEE Internet Computing, (Vol. 21, No. 2), 2017, pp. 97-101.
- [9] M. Raskin: The Law and Legality of Smart Contracts; 1 GEO. L. TECH. REV. 305, 2017.
- [10] I. Bashir: Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained; Packt Publishing Ltd, 2018.
- [11] How It All Began: A Brief History On Bitcoin & Cryptocurrencies; [On-line]; <https://www.ledger.com/2019/03/20/how-it-all-began-a-brief-history-of-bitcoin-cryptocurrencies/>; accessed May 27, 2019
- [12] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman: Blockchain technology: Beyond bitcoin; Applied Innovation, (Vol. 2, No. 6-10), 2016, 71.
- [13] Z. Church: Blockchain, explained. An MIT expert on why distributed ledgers and cryptocurrencies have the potential to affect every industry; 2017.
- [14] I. C. Lin, T. C. Liao: A Survey of Blockchain Security Issues and Challenges; IJ Network Security, (Vol. 19, No. 5), 2017, pp653-659.

- [15] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, C. Yang: The blockchain as a decentralized security framework [future directions]l IEEE Consumer Electronics Magazine, (Vol. 7, No. 2), 2018, 18-21.
- [16] K. Christidis, M. Devetsikiotis: Blockchains and smart contracts for the internet of things. Ieee Access, 4, 2016, pp.2292-2303.
- [17] What is Gossip Protocol?; [On-line]; <https://www.btcwires.com/round-the-block/what-is-gossip-protocol/>; accessed May 28, 2019.
- [18] Sybil Attack; [On-line]; <https://www.geeksforgeeks.org/sybil-attack/>; accessed May 29, 2019.
- [19] CoinMarketCap.Cryptocurrency Market Capitalizations. [On-line] <https://coinmarketcap.com/>; accessed March 12, 2019.
- [20] D. Vujicic, D. Jagodic, S. Randic: Blockchain technology, bitcoin, and Ethereum: A brief overview; In 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), 2018, pp. 1-6
- [21] M. Bartoletti, L. Pompianu: An empirical analysis of smart contracts: platforms, applications, and design patterns. In International Conference on Financial Cryptography and Data Security , Springer, (Cham), 2017, pp. 494-509.
- [22] V. Buterin: A next-generation smart contract and decentralized application platform; white paper, 2014.
- [23] Blockgeeks: Different Smart Contract Platforms; [On-line]; <https://blockgeeks.com/guides/different-smart-contract-platforms/>; accessed April 14, 2019.
- [24] G. Wood: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 2014, 1-32.
- [25] P. Dai, N. Mahi, J. Earls, A. Norta: Smart-contract value-transfer protocols on a distributed mobile application platform; [On-line]; <https://cdn.bitturk.com/whitepaper/qtum.pdf>; accessed May 21, 2019.
- [26] T. Rolfe: Turing Completeness and Smart Contract Security; [On-line] <https://medium.com/kadena-io/turing-completeness-and-smart-contract-security-67e4c41704c>; accessed April 14, 2019.
- [27] Corda Documentation; [On-line]; <https://docs.corda.net/releases/release-M7.0/data-model.html>; accessed May 28, 2019.
- [28] T. Friebe: Bitcoin, Ethereum, and Hyperledger Fabric - which one wins?; [On-line]; <https://medium.com/blockchainspace/3-comparison-of-bitcoin-ethereum-and-hyperledger-fabric-cd48810e590c>; accessed May 28, 2019.
- [29] J. Gilcrest, A. Carvalho: Smart Contracts: Legal Considerations; 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 3277-3281.
- [30] Solidity Security Blog; [On-line]; <https://github.com/sigp/solidity-security-blog>; accessed April 14, 2019.
- [31] Coindesk: Turing Complete Smart Contracts; [On-line]; <https://www.coindesk.com/turing-complete-smart-contracts>; accessed April 14, 2019.

- [32] Coinjournal: Debunking Bitcoin Cannot Do Smart Contracts Myth; [On-line]; <https://coinjournal.net/debunking-the-bitcoin-cannot-do-smart-contracts-myth/>; accessed April 14, 2019.
- [33] S. Falkon: The Story of the DAO - Its History and Consequences; [On-line]; <https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee>; accessed April 14, 2019.
- [34] Coindesk: Understanding DAO Hack; [On-line]; <https://www.coindesk.com/understanding-dao-hack-journalists>; accessed April 14, 2019.
- [35] A. M. Antonopoulos, G. Wood: Mastering ethereum: building smart contracts and dapps; O'Reilly Media, 2018.
- [36] Smart Contract Security Has Some Gaping Holes - Here's How To Plug Them; [On-line]; <https://www.investinblockchain.com/smart-contract-security/>; accessed May 28, 2019.
- [37] Nick Szabo: Smart Contracts: Building Blocks for Digital Markets; [On-line]; http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/L0Twinterschool2006/szabo.best.vwh.net/smart_contracts_2.html; accessed April 14, 2019.
- [38] M. Giancaspro: Is a 'smart contract' really a smart idea? Insights from a legal perspective; Computer law & security review, (Vol. 33, No. 6), 2017, pp. 825-835.
- [39] C. D. Clack, V. A. Bakshi, L. Braine: Smart Contract Templates: essential requirements and design options; arXiv preprint arXiv:1612.04496, 2016.
- [40] BlockchainHub: Smart Contracts; [On-line]; <https://blockchainhub.net/smart-contracts/>; accessed May 28, 2019.
- [41] L. Luu, D. H. Chu, H. Olickel, P. Saxena, A. Hobor: Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (ACM), October, 2016, pp. 254-269.
- [42] M. Alharby, A. van Moorsel: Blockchain-based Smart Contracts: A Systematic Mapping Study; CoRR, abs/1710.06372, October 2017.
- [43] U.S. Senate: Building a Secure Future, One Blockchain at a Time; [On-line]; https://www.jec.senate.gov/public/_cache/files/aaac3a69-e9fb-45b6-be9f-b1fd96dd738b/chapter-9-building-a-secure-future-one-blockchain-at-a-time.pdf; accessed April 14, 2019.
- [44] Vertrag Definition; [On-line]; <https://wirtschaftslexikon.gabler.de/definition/vertrag-49761>; accessed April 18, 2019.
- [45] Der Bundesrat: Rechtliche Grundlagen fuer Distributed Ledger Technologie und Blockchain in der Schweiz; Schweizerische Eidgenossenschaft, Dezember, 2018.
- [46] Legal framework for distributed ledger technology and blockchain in Switzerland; [On-line]; <https://www.news.admin.ch/newsd/message/attachments/55153.pdf>; accessed April 18, 2019.
- [47] <https://www.azleg.gov/legtext/53leg/1r/bills/hb2417p.pdf>; [On-line]; <https://www.azleg.gov/legtext/53leg/1r/bills/hb2417p.pdf>; accessed April 18, 2019.

- [48] Simplification Decree: Italy Accelerates Blockchain & Smart Contract Adoption; [On-line]; <https://coinidol.com/simplification-decree-italy>; accessed May 17, 2019.
- [49] H. Partz: US State of Connecticut Introduces Bill to Authorize Smart Contract Use in Commerce; Cointelegraph. [On-line] <https://cointelegraph.com/news/us-state-of-connecticut-introduces-bill-to-authorize-smart-contract-use\\in-commerce/amp>; accessed March 12, 2019.
- [50] Y. Khatri: Connecticut Lawmakers Seek to Legalize Blockchain Smart Contracts; [On-line]; <https://www.coindesk.com/connecticut-lawmakers-seek-to-legalize-blockchain-smart-contracts>; accessed March 14, 2019.
- [51] W. Zhao: UK Begins Research on Law Reform for Use of Blockchain Smart Contracts; [On-line]; <https://www.coindesk.com/uk-begins-research-on-law-reform-for-use-of-blockchain-smart-contracts>; accessed March 14, 2019.
- [52] Y. Chen, S. Chen, I. Lin: Blockchain-based Smart Contract for Bidding system; IEEE International Conference on Applied System Invention (ICASI 2018), (Chiba, Japan), April, 2018, pp. 208-211.
- [53] P. Sreehari, M. Nandakishore, G. Krishna, J. Jacob, V. S. Shibu: Smart will converting the legal testament into a smart contract; In 2017 International Conference on Networks & Advances in Computational Technologies (NetACT). IEEE, July, 2017, pp. 203-207,
- [54] E. J. Scheid, B. Stiller: Leveraging Smart Contracts for Automatic SLA Compensation-The Case of NFV Environments; 12th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2018), (Munich, Germany), June, 2018, pp. 70-74.
- [55] PolySwarm: 5 Companies Already Brilliantly Using Smart Contracts; [On-line]; <https://medium.com/polyswarm/5-companies-already-brilliantly-using-smart-contracts-ac49f3d5c431>; accessed March 14, 2019.
- [56] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, V. SantamarÃa: Blockchain and smart contracts for insurance: Is the technology mature enough?; Future Internet, (Vol. 10, No. 2), 2018.
- [57] Coin Idol: Italy Delves into E-Voting Using Blockchain Tech; [On-line]; <https://coinidol.com/italy-delves-voting/>; accessed March 14, 2019.
- [58] A. Cohn, T. West, C. Parker: Smart after all: Blockchain, smart contracts, parametric insurance, and smart energy grids; Georgetown Law Technology Review, (Vol. 1, No. 2), 2017, pp. 273-304.
- [59] Market Research Future: Smart Contracts Market is estimated to grow at a CAGR of 32% by Forecast to 2023; [On-line]; <https://www.marketresearchfuture.com/reports/smart-contracts-market-4588>; accessed May 26, 2019.
- [60] M. Yavuz: Fighting Crypto Hacks: Company Tackles Security Issues in Ethereum Smart Contracts; [On-line]; <https://cointelegraph.com/news/fighting-crypto-hacks-company-tackles-security-issues-in-ethereum-smart-contracts>; accessed March 14, 2019.

- [61] The European Parliament And The Council Of The European Union: REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016; [On-line]; <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679>; accessed May 20, 2019.
- [62] A. Van Humbeeck: The blockchain-GDPR paradox; *Data Protection & Privacy*, (Vol. 2, No. 3), pp. 208-212.
- [63] G. Zyskind, O. Nathan: Decentralizing privacy: Using blockchain to protect personal data; 2015 IEEE Security and Privacy Workshops (IEEE), May, 2015, pp. 180-184.
- [64] A. Savelyev: Contract law 2.0: Smart contracts as the beginning of the end of classic contract law; *Information & Communications Technology Law*, (Vol. 26, No. 2), 2017, pp. 116-134.
- [65] B. Marino, A. Juels: Setting standards for altering and undoing smart contracts; in *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, Springer, 2016, pp. 151-166.
- [66] R. O'Shields: Smart Contracts: Legal Agreements for the Blockchain; North Carolina Banking Institute, March, 2017.
- [67] J. M. Sklaroff: Smart contracts and the cost of inflexibility; *U. Pa. L. Rev.*, 2017.
- [68] G. Governatori, F. Idelberger, Z. Milosevic, R. Riveret, G. Sartor, X. Xu: On legal contracts, imperative and declarative smart contracts, and blockchain systems; *Artificial Intelligence and Law*, (Vol. 26, No. 4), 2018, pp. 377-409.
- [69] R. Herian: Legal Recognition of Blockchain Registries and Smart Contracts; EU Blockchain Observatory and Forum, 2018.
- [70] D. L. Hofman: Legally Speaking: Smart Contracts, Archival Bonds, and Linked Data in the Blockchain; 26th International Conference on Computer Communication and Networks (ICCCN 2017), (Vancouver, BC, Canada), July, 2017, pp. 1-4.
- [71] R. Beck, J. Stenum Czepluch, N. Lollike, S. Malone: Blockchain-the gateway to trust-free cryptographic transactions, 2016.
- [72] P. Ryan: Smart Contract Relations in e-Commerce: Legal Implications of Exchanges Conducted on the Blockchain; *Technology Innovation Management Review*, (Vol. 7, No. 10), October, 2017.
- [73] D. Magazzeni, P. McBurney, W. Nash: Validation and verification of smart contracts: A research agenda; *Computer*, (Vol. 50, No. 9), 2017, 50-57.

Chapter 5

Cooperation in Competitive Environments

Han-Mi Nguyen, Dominik Buenzli

With the ever-increasing density of devices connected to the Internet, the risks and problems of a networked world are also increasing. In most cases, such problems can no longer be solved by a single person or company, but require the cooperative cooperation of several players, who often compete with each other. However, it is precisely in such an environment, in which cooperation is a central element, that the necessary willingness is often missing because of the lack of incentives. In addition to economic interests, this problem is made more difficult by the general behaviour of individual actors, who contribute a part to the non-functioning through selfish behaviour. Various schemes have been proposed in the scientific community to address and mitigate these problems. In this paper, we try to shed some light on the different solutions, provide examples and discuss the approaches.

Contents

5.1	Introduction	129
5.1.1	Motivation	129
5.1.2	Research Questions	130
5.2	Background	131
5.2.1	Peer-To-Peer Systems	131
5.2.2	Cooperation in Society And The Tragedy of Commons	131
5.2.3	Where do we Have to Rely on Cooperation?	132
5.2.4	DDoS And The Incentive Chain	133
5.2.5	The Problem of Free Riding	134
5.2.6	Introduction to Game Theory	135
5.2.7	Network Security Games	136
5.3	Incentive Schemes	136
5.3.1	Prize Based Approach	137
5.3.2	Reward Based Approach	138
5.3.3	Game Theoretic Approach	139
5.3.4	Combination of Introduced Approaches: BloSS	140
5.4	Discussion	142
5.5	Conclusions	143

5.1 Introduction

Due to the ever increasing density of devices connected to the Internet, the risks and problems associated with a networked world are also increasing. In most cases, such issues can no longer be solved by a single person or company, but require the cooperative cooperation of several players who are often in a competitive relation with each other.

If we take a Distributed Denial of Service (DDoS) attack on a large online department store as an introductory example, where the provided service is disrupted for several hours, customers nowadays will not hesitate to consider the direct competitors, given that the range of products on offer is approximately the same. Given this scenario, isn't there a purely economic interest for the rival to keep the department store of the competitor offline? Why should they even consider to cooperate and help the target of the DDoS attack? Or as another example, imagine a scenario in which all actors work together in order to achieve a superior goal. In the course of this, the participating actors must adhere to certain rules of the game that guarantee the coexistence and functioning of society. Now, however, some individuals try to enrich themselves with this general generosity and sense of duty, but do not contribute anything or very little to it themselves. Of course, no one would enjoy such beings and the call for regulating mechanisms would quickly become loud. The question that comes up relatively quickly is why there is a lack of cooperation in society, or as in our subsequent case, in the information technology world. Is it only due to human behaviour? Is it due to economic aspects and missing incentives for helping the other participants as stated by [20]? Are there other dimensions to the problem? In science, the problem was studied in depth from different sides and for different problems. In sociology and environmental science, for example, there is the tragedy of commons [16], which deals with the overuse of common goods by individual actors, which reduces the quality of the free good for all if the costs are not adequately addressed. In the IT world, for example in peer-to-peer systems, the same problem also exists for a wide variety of applications. Especially free riding, the consumption of a service without corresponding compensation, poses a particular challenge. To solve this dilemma, various schemes have been proposed that promise relief. We have mainly focused on the following three categories. **Price-based** schemes are designed to promote cooperation by providing monetary incentives. **Reputation-based** approaches are intended to exclude actors with harmful behavior from the system or to lower their quality of service to such an extent that they again have a reason to behave according to the rules. **Game Theoretical** approaches, on the other hand, take up the interplay of actors in one space and are intended to show the optimal strategies for all those involved and thus contribute to an improvement of the general welfare. Since trust also plays a very important role in competitive environments, but a central point of contact in a decentralized environment is somewhat contrary to the actual intention, the implementation of a protocol that takes the blockchain into account is certainly a reasonable solution. Thus, such an approach will be considered in the last example, which is a combination of all aforementioned schemes, introduced by [2]. Subsequently, this paper tries to present some of the proposed solutions found in the literature, compare them with each other and finally show which advantages and disadvantages they offer. In addition, it is to be found out whether a generally valid method exists to solve these kind of issues and to ensure the greatest possible benefit for all.

5.1.1 Motivation

The motivation for stimulating cooperation in competitive environments is very diverse. On the one hand, by exploiting the generosity of individuals, the quality of a service can be greatly reduced. This can go so far that the service is completely brought to a standstill.

Especially in competitive environments, in which individual actors can benefit from the problems of others, schemes that promote the cooperation of individuals and groups are of utmost importance. A good example of this is the alleviation of DDoS attacks. As stated by [4] that by large scale DDoS attacks on Yahoo!, Amazon, CNN and eBay in the year 2000 nearly 2.8% of their market capitalization was lost. It is important to note that this happened almost 20 years ago and the importance of e-commerce grew rapidly. It is therefore easy to imagine that such a far-reaching attack would have a much greater impact today. In [20] it is stated that the solutions in the academic world can mainly be divided into two categories. On the one hand in technical, on the other hand in economic. It is now widely accepted that technical solutions such as cooperative caching or cooperative filtering would help, but the economic aspect has often been ignored. Accordingly, this report aims to investigate why there is a lack of cooperation in such environments and how the cooperation between individual actors can be stimulated.

5.1.2 Research Questions

The following research questions arise from the points mentioned above

RQ1: Why do competitive environments lack cooperation?

RQ2: What schemes do exist to provide an incentive for the different stakeholders to cooperate?

The rest of this work is structured as follows. In section 5.2, the overall context and the terms used in the papers are described. Section 5.3 introduces three different types of schemes that were presented in the literature. In section 5.4, the presented approaches are discussed, as well as compared to each other and evaluated. Section 5.5 finally draws conclusions from the insights gained.

5.2 Background

In the following section, the term Peer-To-Peer (P2P) systems will be introduced in more detail. Afterwards, the analogy of the tragedy of general goods is discussed and the parallels to cooperation in P2P systems are described. The rest of the chapter deals with the introduction of individual sample applications in the P2P environment, which will be consulted for the later explanation of the schemes to promote cooperation. Finally, we describe what an incentive chain is and why it does not work properly in terms of mitigating DDoS attacks.

5.2.1 Peer-To-Peer Systems

Cooperation is a very important part of so-called Peer-To-Peer (P2P) systems, which fundamentally differs from the well-known client-server approach. In a client server approach, as depicted in fig. 5.1, represents a central contact point to which all clients connect. However, this leads to a situation in which the server can be overloaded with an increasing number of requests and thus scaled poorly or not at all. P2P systems on the other hand work without central nodes. Each peer is equal and connected to all other nodes (exceptions exist in Skype, for example, which uses superpeers that work like local servers) and should carry approximately the same workload. Therefore, such a network acts decentralized and allows better scaling. Of course there are not only advantages. As negative aspects, for example, the uncooperative behavior of the nodes (see section 5.2.5), the increased coordination effort due to the omission of a central node as well as connection problems and the omission of individual nodes from the system can be negative. Well, why is this cooperative behavior so important? We will explain this in more detail using an analogy from environmental science in section 5.2.2.

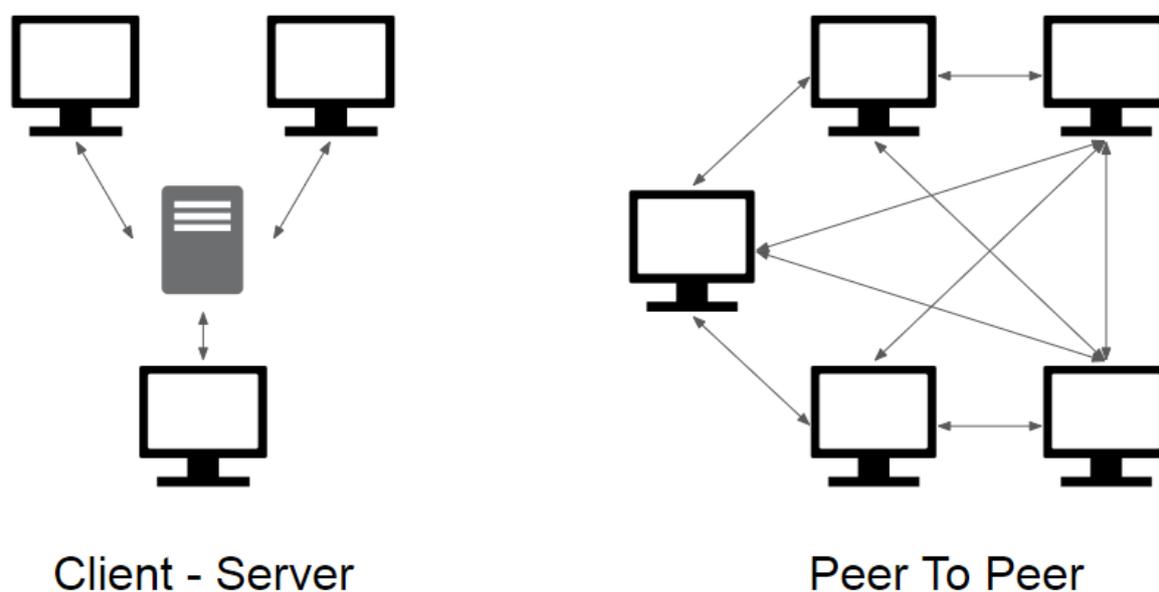


Figure 5.1: Different architectures of the Client-Server and the P2P approach

5.2.2 Cooperation in Society And The Tragedy of Commons

While searching the literature for suitable works, it was immediately noticed that the topic of cooperation is a very broadly spread and treated phenomenon. In particular, many texts were written in psychology and sociology that deal with interpersonal cooperation and

cooperation in society. As stated by [8], a society works best when the individual actors cooperate with each other. Nevertheless, it can be observed that individuals or groups try to take advantage of each others generosity when they are free to do so. This is commonly known as the tragedy of commons. The image depicted in fig. 5.2 and found in [16] describes this phenomenon in a very simple and understandable way. If we take a pasture as a general good and every farmer can let his sheep graze on it, everyone has enough and the ground can recover. However, if a farmer increases the number of sheep, the quality of the pasture will deteriorate and the sheep will have less to eat. If the number of sheep is now increased above the maximum tolerable limit, the soil degenerates and all suffering from it. As long as the environmental costs are not distributed fairly, the pasture will no longer be able to cope with the demand for fodder. This is a parallel to P2P systems, because in both a shared good (e.g. a pasture or a torrent) will only work if no one exploits the generosity of the other, the environmental costs are taken into account and everyone contributes something to the conservation of resources.



Figure 5.2: The Tragedy of Commons illustrated [16] as an easy understandable analogy

5.2.3 Where do we Have to Rely on Cooperation?

The following section will provide a broader insight into different types of P2P networks, which will later be used as examples in different techniques and strategies to increase cooperation. The next section will therefore focus on Ad-Hoc Wireless Networks, Torrents and Distributed Denial of Service (DDoS) attacks.

Ad-Hoc Wireless Networks are described as networks that do not require expensive base stations or wired infrastructure, as the cooperating nodes can communicate directly with each other within the radio range. If a device wants to send information to a device that is too far away to reach it, use nodes other than relays for the messages. It gets even more complicated when you consider that the network topology is constantly changing as devices move. Collaboration is essential for such a network because, for example, a message will never reach a remote device if the nodes do not forward the packets. Incorrect routing information can also cause the network to malfunction. The need for efficient systems that promote collaboration and punish selfish nodes is evident.

Torrents use a P2P network to exchange files. A client connects to multiple other nodes at once and starts receiving data. Once some parts have been collected, it starts distributing the data received so far to other users who have not yet received that specific part. Once the user has collected all the pieces, they are reassembled and the entire file can be used. Afterwards, the user can continue to seed or stop the process, which is not helpful to the system and is considered to be selfish behavior. Additionally, problems arise from a legal perspective, as uploading is considered illegal in most countries. In addition, torrents are known as virus distributors and downloading a file is often considered to be a big risk.

DDoS describes an attack that can usually be divided into two separate steps. Namely the recruitment step, where an attacker tries to attack as many targets as possible and assemble an army of so-called zombies. In the flooding step, the zombies send requests directly or via reflectors to the victim and synchronized IP-Spoofing disables the victim's

services. IP-Spoofing means that an attacker sends a large number of IP packets with a false source address, so there is no point in blocking that address. Only recently there were incidents where blackmailers tried to make money with threatened DDoS attacks and the economic damage is considerable if such an attack is successful, because the service of a provider is usually not available for a long time. Especially in the case of online shops where many alternatives are available, such a failure is serious. The reason why DDoS attacks still pose an unmitigated threat are described in more detail in section 5.2.4

5.2.4 DDoS And The Incentive Chain

Distributed Denial of Service (DDoS) attacks still pose a huge threat to the availability of services in the networked world. Through coordinated attacks of many individual clients and their requests, a service can be brought to a complete standstill. As [5] state, numerous target-resident techniques have been proposed (e.g., packet filtering, anti-spoofing, anomaly detection, protocol analysis). Still, it is a challenging, not to say practically impossible task for a target to defend against a DDoS attack on its own. The problem itself is twofold. Firstly, although the victim is in the best position to detect an attack, it is hard to distinguish between normal and malicious requests. An attacker can even make the attack traffic behave similar to the legitimate one. Second, due to the distributed nature and the scaling of a DDoS attack, a victim may just not have sufficient resources to defend against the attack. As suggested by [14], the defense perimeter for DDoS attacks could easily be moved to intermediate networks. To understand what this means, a highly simplified structure of the Internet must be outlined briefly (fig. 5.3). Since the Internet is subject to the network effect and added value is achieved through universal connectivity, Internet Service Providers (ISPs) have an interest in connecting with other ISPs to offer users the largest possible network. These agreements are called pairing agreements and are concluded between Tier 1 ISPs with comparable network parameters. If we now look at this diagram, it quickly becomes clear that defensive measures of the individual clients do not have the greatest possible efficiency but must take place at a higher level. The approach presented in [14] would work, but is not widely deployed due to lack of economic incentives. This is because intermediate networks are usually not affected by DDoS attacks and are therefore not willing to invest a significant part of their resources for defense. Even if the traffic of a DDoS attack goes through their network, the bandwidth that is available to handle flash crowds can already mitigate such attacks. There exist technical solutions for the aforementioned problem, yet they are not widely implemented. There is for instance **cooperative filtering**, in which ISPs along the attack route filter out malicious traffic. It works in three steps. First, an alarm is triggered when the Intrusion Detection System (IDS) identifies suspicious traffic, then the traffic is traced as far back as possible and finally filtered. However, this would require the prevention of IP spoofing at the edge of the Internet. The respective ISP of the clients could detect IP spoofing by comparing the source address with the route from which the packet is coming and, if necessary, blocking it if the addresses do not match. This makes it much easier to track DDoS attacks. Another approach is **cooperative caching**. Instead of filtering out traffic, DDoS attacks can be prevented by evenly distributing incoming traffic across a large number of cache servers, so each stream is not significant enough to cause congestion. One weakness, however, is that only relatively static content can be cached. However, only the attacked client benefits from the actions taken by the ISPs, and the ISPs are rarely actually compensated for it. Therefore it can be stated, that the mitigation of DDoS attacks is more of an economic problem than a technical one, namely the problem of a broken incentive chain. Still, there exist schemes that try to tackle this issue and try to implement an incentive for the involved actors to cooperate and mitigate

the problem of DDoS attacks together. One promising approach will be discussed in more detail in section 5.3.4.

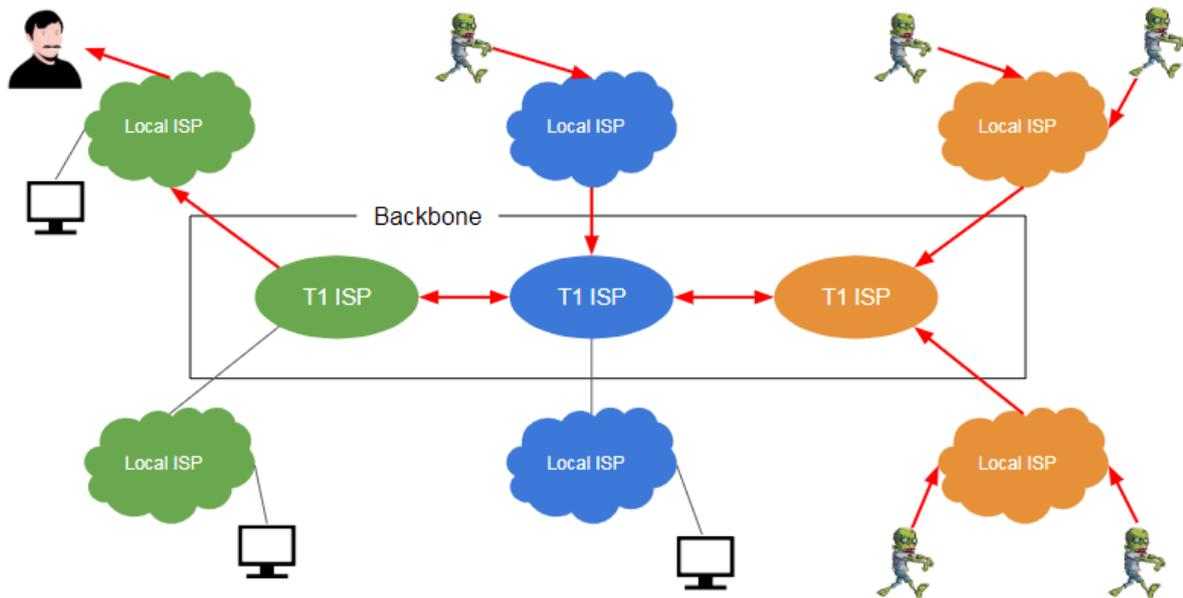


Figure 5.3: Very simplified diagram of the internet with three paired T1 ISPs and zombie nodes starting a DDoS attack against a client

5.2.5 The Problem of Free Riding

Another issue turns out to be free riding in P2P systems. Where some people think that this isn't so bad when they stop seeding after downloading a torrent, they are actually doing a lot of damage to the system. [12] described this phenomenon using Gnutella, a P2P file sharing service, as an example. Studies have shown that almost 70% of all users share nothing, while more than 50% of all sharing is provided by the top 1% of users. Simply put, this means that 70% of all users do nothing good to the system and only benefit while relying totally on the altruistic behavior of the top 1%. The free riders degrade the quality of the service, as they overstretch the search horizon, for example, and therefore requests often do not return results because they exceed the maximum time-to-live. But what exactly do free riders do and what characterizes them? As explained by [11], free riders exploit loop holes in the design, objectives or characteristics of P2P systems, such as self-organization, collaboration and equal standing. In the context of such systems, free riding can be described as the behaviour of individual peers who use the resources of other peers in the network to obtain added value, but without providing adequate compensation. All this has a very strong influence on the performance of the network. To address this problem, the literature [11] describes the identification of users as the first point needed, which may depend on several factors such as network requirements, design, performance requirements and the type of resources shared. Identifying users is very important because there are different types of free riding. For example, a user in a file sharing environment can easily be identified by the use/contribute ratio. But what if a user uploads malicious or useless files? Or if a user is misreporting bandwidth to deceive other users? For all these problems, schemes are needed to punish users who act selfishly or with malicious intent.

5.2.6 Introduction to Game Theory

In the following segment, we explain the basic elements and classifications of games that are later applied to network or security games as per specification of [18]. The basic game elements are the following:

- i. *Players* or agents interacting with each other are assumed to make decisions rationally.
- ii. A set of available *actions* that they can choose from when it's their turn.
- iii. A set of possible *strategies* (consisting of prescribed sequences of *actions* as reaction guideline to all possible scenarios of a game).
- iv. Additionally, players are also presumed to pick the strategy that will maximize their expected *utility*, mathematically denoted as payoff function.

Furthermore, games can be classified into three types of game specifications, namely:

- i. *Number of stages*:
 - static game: a game where players only players choose their actions simultaneously.
 - strategic / dynamic game: multistage game, respectively a sequence (finite or infinite) of static games where players play the same game multiple times.
 - extensive / stochastic game: a dynamic game where the state of a game can change with a transition probability depending on the actions taken and current state.
- ii. *perfectness of information*: depending on whether agents have knowledge about the previous actions of all participants of the game, we distinguish between games of perfect and games of imperfect information.
- iii. *completeness of information*: depending on whether agents have knowledge about all payoff structures of all participants of the game, we distinguish between complete information games and incomplete information games if there exists at least one player who does not know the payoff function of one other player). In addition the Bayesian Games is a special case of the latter, where the payoff structures are known but at least one player does not know the exact type (denoting incomplete information) of the opponent.

It is essential to know the structure of a basic game as other dynamic as well as complex games are extensions of the mentioned base case. However, We will only address static and dynamic incomplete information games as we assume that agents in the network security setting don't encounter the same opposing person again most of the time. Thus, we do not consider the completeness of a game also because it's cumbersome to track back the actions of an anonymous opponent.

The Free-riding problem can be represented as the universally known *Prisoner's dilemma* game. Imagine the following: Two accomplices A and B are to be sentenced to punishment but there is no evidence who the person is that committed the actual crime. They have to choose between *confess* and *lie* simultaneously and independent of each other. An example of such a game is illustrated below, where the payoffs are results of the inverse function of number of years to sit in prison and the utilities are constructed in such a way that it is not profitable to cooperate, i.e. to *lie*:

		Player B		
		confess	lie	
Prisoners Dilemma	Player A	confess	(-5, -5)	(0, -9)
		lie	(-9, 0)	(-1, -1)

The game is not only static but also symmetric due to the fact that players have the same set of available actions. Independent of what the opposing person chooses (confess or lie) the best reaction to the opponent in any case is to *confess*, as it results in the most profitable payoff, because $-5 > -9$ as well as $0 > -1$. Hence, *confess* is a dominant strategy, also called "best response", for both players. This consequence underlines that the *defecting* action is preferred in a game setting where cooperation is not encouraged.

5.2.7 Network Security Games

In the paper [1], the authors suggest three methods in the network security investment setting to determine a possible outcome out of an multiplayer game.

Nash Equilibrium (NE)

The Nash Equilibrium in Game Theory are the combinations of all strategies eventually picked by the players given they act according to the assumption of rationality and profit-maximization. As already deducted from above, the equilibrium of this game in such an competitive environment is the "non-cooperative situation" where all players receive the lowest payoff out of all possible scenarios. The reason why it's called Prisoner's "*Dilemma*" is due to the fact that there is no incentive to cooperate under such circumstances even though there clearly exists another outcome that is more profitable for both of the involved persons, specifically $((lie, lie))$ or $((cooperate, cooperate))$. Not only do concerned individuals suffer due to lack of cooperation but also the overall network security level remains on a low level. On one hand, we can talk about an equilibrium because one out of the four possible outcome indeed realized, but on the other hand, the outcome is sub-optimal because this is the worst-case scenario in comparison to the other possible states.

Nash Bargaining (NB)

In this approach, we take the Nash Equilibrium (non-cooperative situation) as starting point and allow the agents to negotiate among each other. Though the opposing players can defect, this also means that the player in question can do the same as all players in this setting have the same set of strategies (symmetric game). This guarantees higher expected utilities collectively because every individual will want to increase his payoff and can effectively do so. There are no losers as everyone is looking after themselves and investing in network security measurement, thus indirectly improving overall network security.

System-Optimized (SO)

The system-optimized viewpoint proposes the concept that there exists a social optimizer who knows the payoff structures of each player and maximizes the sum of expected utilities. Although this ensures a higher network security level it however cannot prevent that some players might end up as losers. An individual loses if the value of his utility out of the maximization problem is even lower than that in the non-cooperative situation.

5.3 Incentive Schemes

In the following section, the three types of schemes, namely price-based, reputation-based and game-theoretical solutions, are examined and explained in more detail to answer RQ2. Next to these three main approaches to encourage cooperation in a competitive

environment we illustrate the functionality for each type by examples such as Ad Hoc Wireless Network, P2P file-sharing and DDoS Mitigation.

5.3.1 Prize Based Approach

Monetary payment schemes are of course one of the first alternatives that come to mind when thinking of an incentive mechanism. It is basically about compensating people financially for providing services and charging them for consumption. In the following sections we will present two different approaches that are used as a monetary incentive scheme in Ad-Hoc Wireless Networks.

Nuglets

introduced by [3] and [15] is a virtual currency system which implements a mechanism to charge reward the service usage, respectively provision in an Ad-Hoc Wireless Network. Autonomous Systems are represented as nodes that load their data packages to dispatch with enough virtual currency to traverse through the network of nodes. In the system implemented, there are two variants of how the forwarded packages are billed, namely Package Pursue Model (PPM) and Package Trade Model (PTM). In the former, the source node has to pay for the data transfer. This avoids overload within the distributed network because sender nodes are enforced to manage their resources and data outflows efficiently as well as thoughtfully. The distance traveled of a data package is completely dependent on how much Nuglets it is filled with by the sender node. Hence, there is a possibility that the delivery comes to a halt due to insufficient number of Nuglets. However, in Ad-Hoc Wireless Networks, the receivers profit from data exchanges most of the time which makes it unreasonable to expect that a sender has to pay when transferring data because there is no incentive to invest in virtual currencies on his part. By comparison, in the Package Trade Model, the destination node has to pay for the received data transfer and there are intermediary trades allowed in between the traversed nodes. Although this scheme projects the incentives of sender and destination nodes correctly and indeed establishes cooperation, malicious nodes can overload the network without being charged. The virtual currency system persuades with its decentralized and independent property, however tamper proof hardware is needed to undertake for the functioning.

Sprite

In the publications [17] and [15], Sprite - also a virtual currency - is used to settle up credits and charges determined by a central Credit Clearance Service (CCS). Sender nodes obtain receipts for transmission of messages and forward the receipts to the CCS to get paid. The monetary inquiry will only be cashed out only if the receiver of the data package actually forwards it again, where the receipt of the receiver node is examined for that purpose. In this context, compared to Virtual currency system with Nuglets, there is no particularly secure hardware needed but honest reports of single nodes about their actions is still ensured. Of course, a reliable and independent central credit management is required for the system to work.

Problems of Prize Based Approach

Out of all concepts that are mentioned in this paper, Monetary based Schemes are the simplest and most trivial. Money exchanges for services between nodes within a distributed network have to be worked out anyway but a trusted and central intermediary for secure currency transfers is indispensable. In the case of the Package Pursue model, the sender node has to earn a sufficient amount of Nuglets to pay for sending the data not giving him an incentive to invest in virtual currency. In the Package Trade Model, this problem does not appear because sender as well as destination nodes have the incentive to cooperate. Thus, egoistical behaviour can be tackled but the identification of malicious attacks is difficult and gives rise to the possibility that some nodes are endangered to lose all their Nuglets in both introduced Virtual Currency Systems.

5.3.2 Reward Based Approach

Incentive Schemes are incorporated to increase cooperation and sharing but these mechanisms do not necessarily avoid false claims and inauthentic file uploads (just to get benefited by the system for uploading). The basic idea of reputation based schemes is to keep track of the reputation of users to ensure their trustworthiness and block malicious users. We will illustrate three approaches by reference to P2P File sharing.

EigenTrust

The paper [13] describes the EigenTrust algorithm that calculates a global trust value as guiding principle for reputation within the P2P file-sharing network and supplementary reducing the aggregate replication rate of inauthentic files. Each peer records satisfactory or unsatisfactory experiences by other peers which then are used for the calculation of its reputation measure. This means of reputation management helps to improve transparency within the highly anonymous as well as non-cooperative environment. Furthermore, the authors claim that malicious peers can be identified and isolated because users can choose by themselves who they want to interact with next to their credible acquaintances based on these global trust values. Additionally, this reputation management system also offers rewards for cooperative behaviour and not only encourages peers to upload more files but also stimulate malicious nodes to remove inauthentic files in their repertoire that could tarnish their global trust score. Though the aggregation of the score is rather difficult in a distributed network but EigenTrust counteracts this by asking trusted neighbours for suggested peers.

Watchdog & Pathrater

In [15], Watchdog is described as a reactive defense mechanism with monitoring technique that assumes data transfers are broadcasted, meaning that all nodes can receive these messages when being in vicinity of the sender. The task of the Watchdog is to detect malicious nodes. It does so by assigning an error counter to each node and by checking if the receiver node has forwarded the package. In the case where message is not forwarded the node currently acting as a sender accounts an increase of 1 to its counter. The sender node is reported as malicious if the predefined threshold is reached. Afterwards, a Pathrater can supplementally help finding possible routes without the identified malicious nodes. This Watchdog&Pathrater solution might lead to an over-investment in monitoring systems. Moreover, this mechanism does not encourage nodes to actively cooperate out of intrinsic motivation but relies on the correct surveillance and its intimidating effect on peers behaviours. Hence, this functional interaction might not be sustainable and self-containing.

CORE

As summarized in [15], CORE stands for *CO*llaborative *RE*putation Mechanism and makes simultaneous use of monitoring and reputation systems. The watchdog described above represents the first component whereas the reputation system creates a reputation table for each node containing reports of other peer nodes they have indirectly and directly interacted with quantifying the willingness to cooperate. The node in question receives a bad subjective record if it is not eager to directly cooperate with peers and thus will not be listed in the global list of nodes with positive reputations that will be distributed globally as well as periodically.

Problems of Reputation Based Approach

The above-mentioned Reputation based mechanisms confirmed that the perception and judgment of others about one's actions can indeed influence the behavior, especially in a distributed network where one is dependent on others. But they also bear some disadvantages, such as: the assessments of reputation might be biased or inappropriate due to easily modifiable information when distributed. Next, the ratings can be self-fulfilling

prophecies and must be assessed, updated and shared continuously as reputation changes over time. Furthermore, the state of network topology must be checked permanently.

5.3.3 Game Theoretic Approach

As of yet, price or reputation based schemes were elaborated which are settings where individuals are more or less directly influenced by other agents. However, the game-theoretic approach specifically takes interacting dynamics into account.

Repeated Games

The Free-riding problem of Gnutella compared to BitTorrent was that past events were not trackable (with reference to *imperfect information game*) and peers encountered anonymous players, i.e. peers did not meet the same player enough or could not know if they're dealing with the same player (with reference to *incomplete information game*). BitTorrent solves this by introducing an interaction in their P2P file-sharing network that is similar to a repeated game on piece-level. A user can only download a piece of another peer if they upload a piece at the same time. Ultimately, a whole file on BitTorrent can be acquired if all pieces necessary are collected. This condition guarantees cooperation in the highly uncooperative file-sharing setting because one is forced to contribute in order to consume. Furthermore, in the long term it is more beneficial to stay in cooperation than defaulting. Defaulting indeed delivers a large payoff (in comparison to small but consistent payoff in the cooperative situation) at first but - if players are bound to deal with each other again - this past negative behaviour might lead to the opponent not willing to continue participating in the game. As mentioned before, this constellation ensures cooperation but this equilibrium is not efficient because there exists a dominant strategy, namely "strategic piece revelation" (illustrated in fig. 5.4), that slows down the process of acquiring all pieces. A peer i remains popular if peers j and k currently do not play this game on piece-level with each other for a specific file plus both are dependent of peer i (i.e. peers j and k do not have the piece peer i possesses). The best reaction to other peers is to stay interesting, i.e. to be able to play the game on piece-level for as long as possible and continuously collecting pieces of files. This strategy consists of revealing or uploading the least rare piece first before supplying a rare piece that is hard to acquire. By untruthfully communicating about its possession of pieces the peer remains popular for other peers and can keep on playing the file-sharing game.

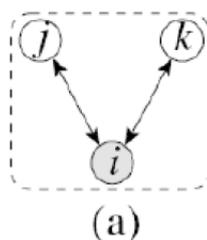


Figure 5.4: "Strategic Piece Revelation" by ©Sven Seuken, Department of Informatics, University of Zurich

Client Puzzle

Distributed Denial of Service flooding attacks consume resources of victims not allowing them processing their users' requests. The attacker sends a plethora of junk requests to its victims who cannot distinguish the malicious nodes from the other autonomous systems and starts to handle the inquiries anyways. In game-theoretic notation, this situation can be translated to a an *incomplete information game*, where the type of the requester (malicious or harmless) is unknown. The preventive defensive mechanism proposed in [10]

propounds the idea that IPS first should send out a puzzle to the requester to solve before dealing with the received request. If the requester is indeed harmless, he is willing to invest resources in solving the task. consequently, he sends the signal of him being harmless to the counterpart giving him the green light to process the inquiry. This mechanism warrants that both malicious and legitimate nodes have the correct incentive to indicate their actual type and therefore allowing other nodes to distinguish them. Sharing personal information of one's own volition results in the reduction of other players belief of one being malicious as well as overall incomplete information in the network.

Problems of Game Theoretic Approach

A constellation of a game can be self-containing and stable if players are induced to act in a predetermined manner that corresponds to their best option. But most of the time, game-theoretic models are computationally complex. Because finding even one equilibrium out of the possible scenarios is difficult. Sometimes, if an equilibrium could be deducted, the Nash Equilibrium may not be dominant enough to be considered as the best prediction of an outcome. Furthermore, the underlying assumption that humans are rational is erroneous as people make mistakes accidentally or on purpose. Human beings also have inconsistent time preferences, meaning that their actions don't make sense over time. For instance: students might want to create study plans in the beginning of a semester but also want to procrastinate as time passes.

5.3.4 Combination of Introduced Approaches: BloSS

All proposed incentive schemes in this paper do not preclude each other, rather we recommended to combine some features for a more self-contained and predictable outcome. This was achieved in the script [2], where the mechanism of BloSS is elucidated. BloSS stands for Blockchain Signalling System which combines game-theoretic as well as Monetary based aspects in a Reputation Based Scheme using the Blockchain. Briefly, it is hardware mainly developed to encourage cooperation among autonomous systems to fight against DDoS attacks as it would be impossible to counter them alone. It particularly supports and facilitates the exchange of information about potential DDoS attacks, adopting our game-theoretic approach. Furthermore, this increase in transparency gives rise to better fundamentals concerning the financial communication within this closed system - and thus more confidential setting, that eventually can be translated into monetary incentives. Therefore, incomplete information or its complexity must be reduced such that agents are motivated to effectively implement the existing solutions. This can be done on an aggregate level through signalling the facts about offensive actions in the distributed network, as in [2], or actively and voluntarily signalling one's type to reduce the asymmetry of information on one's side. Even though cooperation can be established in such a setting, malicious behaviour cannot be fully tackled yet because security measures are not undertaken explicitly according to the authors. The (fig. 5.5) describes the possible interactions between a Target of a DDoS and a Mitigator in the rather closed Blockchain system. The service provided by the Mitigator gets rated by the Target and receives a payment according to the assessment.

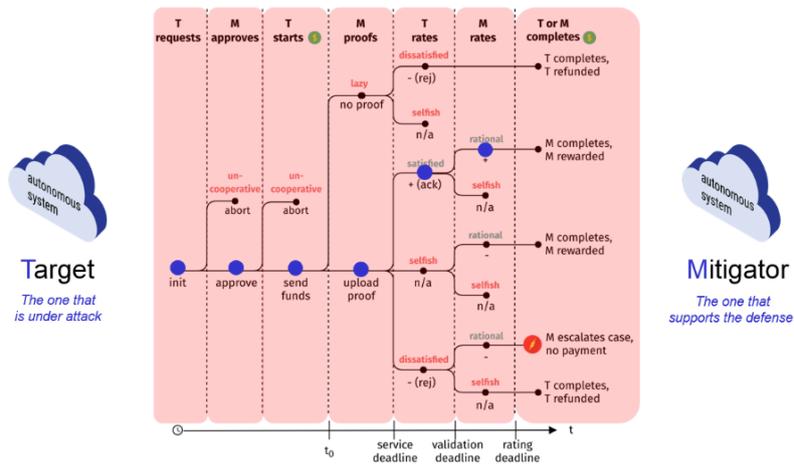


Figure 5.5: BloSS for DDoS Mitigation by © Communication Systems Group (CSG), Department of Informatics (IfI), University of Zurich (UZH)

5.4 Discussion

The section 5.3 about Incentive Schemes already addresses our second research question **RQ2** and listed some existing and promising schemes that more or less generate the correct incentive for participants engaged in service exchange in the distributed network. Though, the interesting question is why individuals in competitive environments still do not cooperate to some extent or at all, referring to our first research question **RQ1**. Our initial position is the fact that autonomous systems do not cooperate in competitive environments because they don't have an incentive to do so. In game-theoretic notation, the *non-cooperative situation* is the result of chosen strategies of the agents assuming they picked their best and profit-maximizing strategy given their knowledge about the other players. It just so happened, that the set-up of the whole game induces the players to choose the sub-optimal outcome out of all scenarios. Further on, we presumed four factors that could restrain agents from implementing the already existing solutions to cooperation, specifically: Decentralization, Selfishness, Human Behaviour and Schemes.

Decentralization

If autonomous systems are allowed to act on their own, they will most probably end up in the non-cooperative situation as explained before. A centralized instance might indeed guarantee a higher overall network security as elaborated in the *System-Optimized Model*, most notably a better outcome than in the non-cooperative case. But the existence of a centralized instance is not the main issue, which can be confirmed by the *Nash Bargaining* model, where players are guaranteed to obtain a higher payoff out of the network security game. The reason behind is that they are allowed to deal among each other and everyone has an incentive to improve their utilities. They know there exists an outcome that can be reached by negotiation which reinforces cooperative behaviour.

Selfishness

Egoistic behaviour is a huge problem concerning cooperation but this doesn't mean that it cannot be tackled. If incentives are shifted in a way that individuals can maximize their profit as well as increase social welfare, the whole mechanism not only remains self-contained but also sustainable. Additionally, the selfish essence of individuals can be a clear indicator that actions are going to be taken as estimated helping the system developer to create a setting with representative payoffs that leads players to the desired outcome without intervening with their nature.

Behaviour

Humans are not in a position to make perfect rational choices. Thus, it is difficult to predict their actions when underlying a system where they are interacting with other beings. In contrast, computer systems are more apt to make rational choices than humans because they have to follow a protocol such that actions at a later point in time cannot be devious from the moment the decision was made, forcing the decision-making and compliance of a human to be more predictable.

Schemes

In fact, the effective implementation of those schemes depends on the belief about communicated behaviour of other agents. Hence, the reduction of incomplete information is essential and can be achieved through signalling mechanisms. Autonomous systems are also hesitant to work collaboratively if there underlies an asymmetry of information. Additionally, by seeing the efforts made by other players, one is more willing to engage in an interaction. Thus, the investment in sending a signal to a favored counterpart for potential cooperation is a strategy that is recommended to be offered in an incentive scheme.

5.5 Conclusions

In this paper we revised three main approaches to encourage cooperation in a competitive environment by examples such as Ad-Hoc Wireless Network, P2P filesharing and DDoS Mitigation. Monetary payment schemes are the most trivial method that tackle financial incentives in the pay system, obviously influencing the actions of participants within the network. Reputation based schemes affect their behaviour less directly than price based schemes and especially report reliability of users to detect malicious individuals. Additionally, the game-theoretic approach especially takes the interplay between players' actions in account that forms the outcome. Some of the introduced models can solve the problem of cooperation but unfortunately either do not guarantee optimal social welfare or punish harmful nodes at the same time. However, a combination of the proposed approaches can cover the weaknesses of the other methods as in [2]. Nevertheless, it must be noted at this point that a more detailed analysis of the available approaches cannot give a generally valid formula for presupposing cooperation between individual actors in any environment. Different approaches are needed for different applications such as Ad-Hoc Wireless Networks and DDoS Mitigation, as these also contain completely different problems and cooperation does not always take place for the same reasons. Where in torrents, for example, sharing is in the foreground and a reputation system could be sufficient to exclude players who behave incorrectly, it is not enough to rely on reputation alone in a DDoS mitigation scenario, since an economic aspect also plays a role here. Thus, even a monetary aspect will not be negligible. Finally, it should be noted that trust also plays a very important role. In many schemes, a central point is required, e.g. to handle the clearing of a payment. However, this contradicts the decentralised approach. As presented by [2] blockchains could be used, which offer a decentralized alternative to establish trust. By reading the literature it became clear that a single approach alone could not cover all aspects, but a combination of several such schemes would certainly be advantageous. Here it would certainly be advisable to do more research and the method of [2] certainly goes in the right direction.

Bibliography

- [1] Anna Nagurney, Shivani Shukla. *Multifirm models of cybersecurity investment competition vs. cooperation and network vulnerability*. 2013, European Journal of Operational Research, pp. 588-600.
- [2] Bruno Rodrigues, Thomas Bocek and Burkhard Stiller. *Enabling a Cooperative, Multi-domain DDoS Defense by a Blockchain Signaling System (BloSS)*. Communication Systems Group (CSG), Department of Informatics (IfI). UZH.
- [3] Buttyan Levente and Hubaux Jean-Pierre. *Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks* . 2001, [On-line], <http://infoscience.epfl.ch/record/52377>, last visit April 14, 2019.
- [4] Cavusoglu, Huseyin, Birendra Mishra, and Srinivasan Raghunathan. *The Effect of Internet Security Breach Announcements on Market Value Capital Market Reactions for Breached Firms and Internet Security Developers*. 2004, International Journal of Electronic Commerce 9, no. 1, pp. 69-104.
- [5] Guanhua Yan, Stephan Eidenbenz. *DDoS Mitigation in Non-cooperative Environments*. 2008, [On-line] <http://www.cs.binghamton.edu/~ghyan/papers/networking08.pdf>, last visit Feb 14, 2019.
- [6] Harikrishna Narasimhan, Venkatanathan Varadarajan, C. Rangan. *Towards a Cooperative Defense Model Against Network Security Attacks*. 2010, [On-line] https://www.researchgate.net/publication/268177490_Towards_a_Cooperative_Defense_Model_Against_Network_Security_Attacks, last visit, Feb 14, 2019.
- [7] Jens Grossklags, Nicolas Christin, and John Chuang. *Secure or insure? A gametheoretic analysis of information security games*. 2008, In Proc. 2008 World Wide Web Conference (WWW08), pp. 209-218.
- [8] Kaune Sebastian, Pussep Konstantin, Tyson Gareth, Mauthe Andreas, Steinmetz Ralf. *Cooperation in P2P Systems through Sociological Incentive Patterns*. 2008, [On-line] https://www.researchgate.net/publication/221035927_Cooperation_in_P2P_Systems_through_Sociological_Incentive_Patterns, last visit Feb 14, 2019.
- [9] Margit A. Vanberg. *Competition and Cooperation in Internet Backbone Services*. 2009, [On-line] https://www.researchgate.net/publication/226259088_Competition_and_Cooperation_in_Internet_Backbone_Services, 10.1007/978-3-7908-2082-9_2, last visit Feb 14, 2019.
- [10] Mehran S. Fallah. *A Puzzle-based defense strategy against flooding attacks using game theory*, IEEE Transactions on Dependable and Secure Computing, Vol. 7, No. 1, pp. 5-19, 2010.

- [11] Mohammed Onimisi Yahaya. *Free Riding in Peer-to-Peer Networks: Review and Analysis*. 2015, African Journal of Computing & ICT. Vol 8, No. 1. Pp 53-60.
- [12] Rahul Mishra, Anirban Mondal. *Incentive Schemes for Mobile Peer to Peer Systems and Free Riding Problem: A Survey*. 2016, [On-line] <https://arxiv.org/ftp/arxiv/papers/1606/1606.07785.pdf>, last visit Feb 14, 2019.
- [13] Sepandar D. Kamvar, Mario T. Schlosser, Hector Garcia-Molina. *The EigenTrust Algorithm for Reputation Management in P2P Networks*. 2003, Proceeding WWW '03 Proceedings of the 12th international conference on World Wide Web, pp. 640-651.
- [14] Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson. *Practical Network Support for IP Traceback*. In Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. ACM, New York, NY, USA, 295-306.
- [15] Sudip Misra, Isaac Woungang and Subhas Chandra Misra. *Guide to Wireless Ad-Hoc Networks*. 2009, 5th edn. London: Springer.
- [16] Sustainable Environment Org. [On-line] <https://www.sustainable-environment.org.uk/Earth/Commons.php>, last visit April 14, 2019.
- [17] S. Zhong J. Chen Y.R. Yang *Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks*. 2003, IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), San Francisco, CA, pp. 1987-1997 vol.3.
- [18] Xiannuan Liang, Yang Xiao. *Game Theory for Network Security*. 2012, [On-line] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.708.113&rep=rep1&type=pdf>, last visit Feb 14, 2019.
- [19] Yu Liu, Cristina Comaniciu, and Hong Man. *A bayesian game approach for intrusion detection in wireless ad hoc networks*. 2006, In Proc. 2006 workshop on Game theory for communications and networks.
- [20] Yun Huang, Xianjun Geng, Andrew B. Whinston. *Defeating DDoS Attacks by Fixing the Incentive Chain*. 2007, [On-line] <https://dl.acm.org/citation.cfm?doid=1189740.1189745>, last visit Feb 14, 2019.

Chapter 6

Network Functions Virtualization (NFV) in Smart Cities

Lawand Muhamad, Can Inan, Atif Ghulam Nabi

In the last few years, many researchers and industrialists have been showing an interest for developing cities in the future. They reach the conclusion that technology will play a significant role in transforming the existing infrastructures into smart cities. Due to the rapid increase in the urbanization process cities are facing new challenges across the world. It draws the attention of city planners and researchers to solve these challenges in an optimal way using Information and Communications Technology (ICT). One technology that has been growing in popularity is Network Functions Virtualization (NFV) to run the network services in a virtual environment. This report provides a comprehensive scientific overview on smart cities and the role of NFV in combination with Software Defined Networks (SDN) to create the cities in the future. We mainly focus on the benefits, challenges and opportunities for implementing NFV in smart cities.

Contents

6.1	Introduction	149
6.2	Background	149
6.2.1	Smart Cities	149
6.2.2	Network Function Virtualization	158
6.3	Opportunities for NFV in Smart Cities	162
6.3.1	NFV and SDN the key Enablers	162
6.3.2	Benefits	163
6.3.3	Opportunities	163
6.3.4	Real World Use Cases	165
6.3.5	Challenges and Discussion	167
6.4	Conclusion and Future Directions	169

6.1 Introduction

The world is moving towards smart cities due to continuous increase in the urbanization process. It's expected that more than 70% of the population will live in the cities by 2050. The motivation for people to move into the cities is the economic welfare to fulfill their needs and enjoy the life with state-of-the-art facilities in urban areas [1]. It brings new challenges for the city governments and administrators to provide the optimal services and improve the quality of life for their citizens.

These problems can be solved by realizing the concept of smart cities using smart technologies such as sensor networks, Internet-of-Things (IoT), Software-Defined Networks (SDN), Network Functions Virtualization (NFV) and 5G networks. It is required to embrace the virtualization in the networks to build the scalable infrastructures for communication using smart technologies to solve the emerging challenges for government institutions such as real-time monitoring and low latency communication among the interconnected departments [17].

To this end, NFV technology has paramount opportunities in the context of smart cities that can help to run the network services in commercial off-the-shelf environments, which can significantly reduce the costs for providing the services [11]. The main goal of this report is to discuss the opportunities and benefits for using NFV technology to provide optimal services in the smart cities. It also provides a scientific overview on the theoretical models and frameworks for the smart cities along with real world use cases.

6.2 Background

In this section, the main concepts and definitions of Smart Cities and NFV is introduced. Also, the related fields that enable great opportunities for the city of future is discussed.

6.2.1 Smart Cities

A city is considered a conglomeration of human settlement. There is a long history of towns and cities, but there are different opinions that which ancient settlements are considered as cities. The urbanization process is the movement of people from small villages and towns towards cities, it mainly a process due to which villages are becoming towns, towns are shifting into urban areas and urban areas are growing into cities. The population division of United Nations highlights the processes that are contribution in the rapid growth of urbanization as shown in the Figure 6.1.

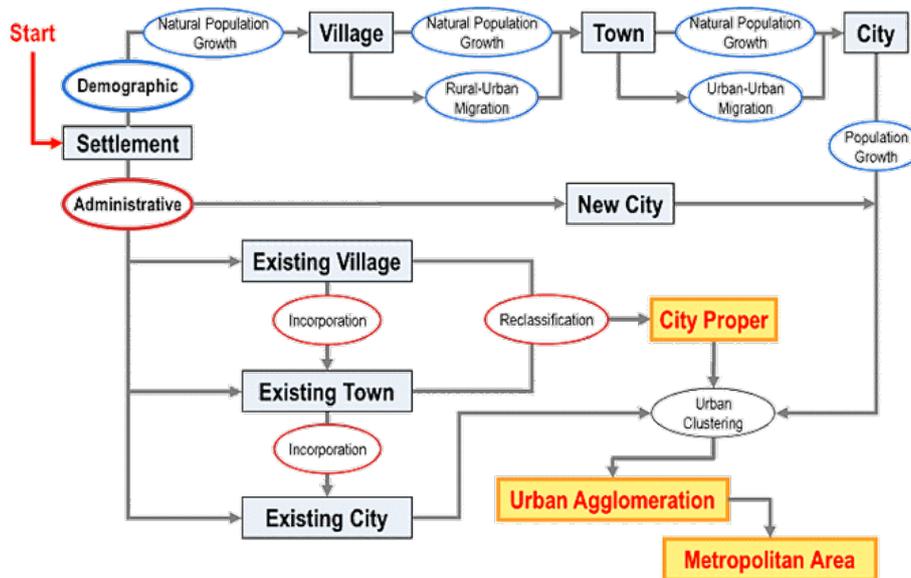


Figure 6.1: Urbanization Process [41]

Nowadays, cities generally have comprehensive systems and infrastructure for housing, transportation, waste management, public utilities, land use, and communication. The continuous increase in the percentage of people who are living in the cities, and the ways in which they adapt the changes in urban areas are bringing new challenges for the cities. The fast growth in the urbanization has introduced many challenges for citizens such as higher costs of living, higher mortality rates, worse pollution, traffic and high commuting time. The growing list of challenges has focused the attention of city planners and researchers to solve these problems in an optimal way, with the aim to improve the quality of life in urban areas. It has introduced smart cities as a new concept to reach the sustainable goals using Information and Communications Technology (ICT). The government institutions and private industries are focusing to provide optimal services and facilities their citizens leveraging the potential of advanced ICT in order to make the infrastructure more sustainable.

The use of ICT adds two basic features to implement smart objects: the sensing and the automation. In this direction, there are many devices in the market for sensing and collecting information such as noise, temperature, pollution. It is required to consider additional features when we talk about a system such as Identity which allows to detect objects in cyber space and connectivity that enables communication for data transmission [1]. The smart phones in these days are manufactured with build-in sensors to collect and share the information in order to improve their decisions.

6.2.1.1 Definition

There are more than forty definitions for smart cities in scientific literature. There is not any unified acceptable definition for smart cities among the researchers because of multi-dimensional attributes of a city such as geography, environment, social and economic aspects.

OpenLearn [34] defines smart cities as a term used to describe the use of smart technologies and data to solve cities sustainability challenges. It describes the term smart cities in a simple way using two components, smart technologies and data, that aims to solve the sustainability challenges. It includes new Internet technologies, the Internet of Things, smart phones and smart meters, networks of sensors and RFIDs. These smart technologies help to collect useful data, which is measured and analyzed using state-of-the-art tools to generate insights for public benefits.

There are some other perspectives on the smart cities such as Taewoo Nam [32] describes that a city becomes smart if the investments in human, social capital, modern transport, and communication enables the sustainable economic growth with a high quality of life. Giffinger et al [18] defines smart cities as "Regional competitiveness, transport and Information and Communication Technologies economics, natural resources, human and social capital, quality of life, and participation of citizens in the governance of cities."

In the same direction, Geller, A. L [15] defines that as "A city well performing in a forward-looking way in economy, people, governance, mobility, environment, and living, built on the smart combination of endowments and activities of self-decisive, independent and aware citizens". He puts stress on the smart combination of endowments and activities of independent citizens. The list of definitions goes on with diversified views and opinions that gives a robust understanding on smart cities involving different disciplines.

6.2.1.2 Dimensions

In the context of a city, the term smart has different dimensions (Fig. 6.2) which include technology, people and community. The cities can be called smart by using the smart technologies when we also consider the people and the communities living in the cities.

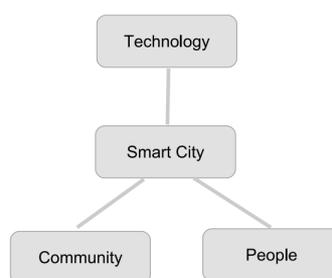


Figure 6.2: Repository axis[1]

The first dimension relies on the quality of information systems specifically integrates computing, telecommunication and modern infrastructure to gather the needs of public. The city is considered smart in real terms when we specify is not just the technology, but also the people and the communities.

People dimension in Figure 6.2 include several aspects such as "Creative City, Learning City, Human City and Knowledge City. [35]." For the third dimension, "Smart Community is a strategy that aims at involving the most significant number of users in IT [22], ranging from a small to a wide community, to improve their quality of life [29]."

There are also some other dimensions covered by other researchers such as *Virtual City* where all functions are implemented in a cyberspace and *Cognitive Smart City* that expands the concept of the smart city by referring to the convergence of the emerging Internet of Things (IoT) and smart city technologies, their generated big data, and artificial intelligence techniques.

6.2.1.3 Frameworks

Many concepts of the smart city depend on the combination of technological infrastructure. Jung Hoon Lee[26] describes the following four technological aspects of a smart city for technology framework.

Digital City: This kind of city allows to create an environment using the broadband communication infrastructure where the citizens are connected with each other and they can share the information easily to anyone in the city.

Intelligent City: As the name represents, it includes research to create an atmosphere for creating new skills and innovation in the urban areas. One of the most important characteristic is that technology infrastructure remains up to date using the latest technology.

Ubiquitous City: The ubiquitous city allows the citizens to communicate with each other and any services in the city using any smart device, this kind of city is build based on the digital city where the underlying infrastructure supports the distributed network of devices in the city.

Information City: It gather the information from the network of IoT devices, this information is made available on public platforms for the benefit of citizes in order to solve the problems and challenges. Such a city enables the citizens to access the real time data which they can use to perform analysis using the latest data analysis techniques and allow them to contribute to solve the open challenges in the cities.

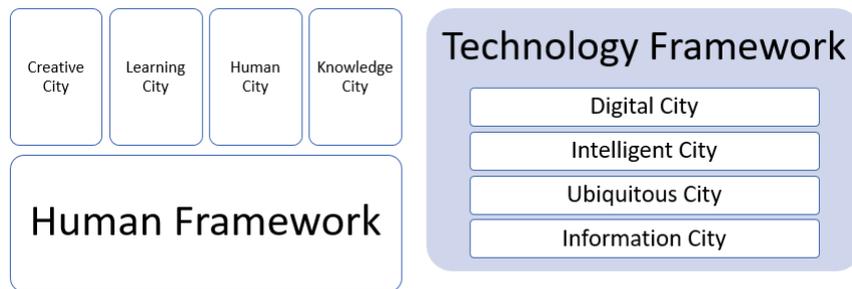


Figure 6.3: Frameworks

There are also other frameworks such as *Institutional framework* [10] that allows to work in a strategic partnership with government and other institutional organizations to enable the use of IT for improving the quality of life, *Energy framework* [9] where the city has a smarter energy infrastructure and *Data Management framework* [17] where smart city employs a combination of data collection, processing, and disseminating technologies in conjunction with networking and computing technologies to promote the overall quality of life for its citizens.

6.2.1.4 Models

In order to achieve the goals for smart cities, many leading companies, city planners and researcher have been working over the last year to develop frameworks. There are many examples available in the developed countries around the world that have made a considerable progress to make the concept of smart cities a reality.

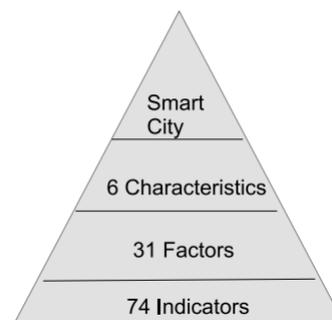


Figure 6.4: R. Giffinger's Model Structure [18]

Nevertheless, the new cities are still facing challenges and problems in adopting the successful models for making the cities really smart. Some of the important models are

described in this section that can be applied to the cities with several dimensions and demographic structure.

The industry and academia are working together with mutual cooperation of government institutions for making smart cities a reality. They have developed several approaches and models that are applicable for urban areas considering various dimensions such as social, economic and geographical structures. The Table 6.1 presents Smart Cities Models that can be experimented to cities having various dimensions. These models are also described below.

Giffingers et al. [18]	Smart cities indicators Cohen
Cohen [16]	Smart Cities Wheel Model IBM
IBM [38]	Smart Cities Nine Pillar Model

Table 6.1: Smart City Models [1]

The purpose of smart city is to create smartness across the industries, educational institutions, government sector by applying the latest technologies in the daily life. In [18], the author categorizes Smart City into six characteristics "Smart Economy, Smart Environment, Smart Governance, Smart Living, Smart Mobility, and Smart People" as described in the work previously mentioned. The author also presents the smart city model in an hierarchical triangle to highlight different aspects of the smart city along with categorizing each characteristic by some factors as it's highlighted in the following Figure 6.5.

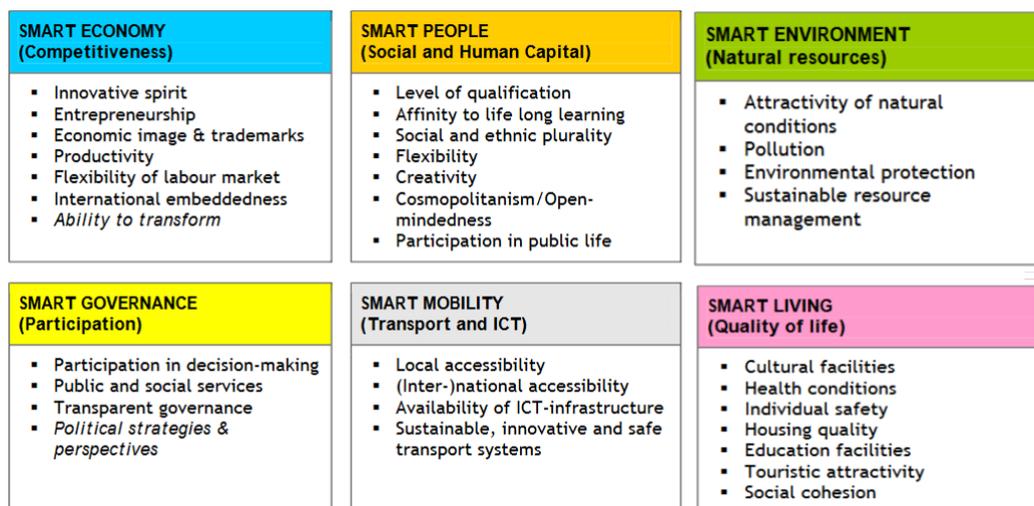


Figure 6.5: Characteristics and factors [18]

Each characteristic is defined by various factors (e.g. Smart Economy is defined by innovative spirit, entrepreneurship, economy image & trademarks, productivity etc.) and each factor is highlighted by different indicators. These factors were defined in various settings with smart city development in mind. At the end, 31 factors were selected to explain the 6 characteristics in the model. In order to measure the performance and evaluation 1-4 indicators were selected in each factor. The characteristics, factors and indicators for each factor design the framework for the indicators and the following assessment a city's performance as smart city.

Boyd Cohen [16] finds six important keys, in his Wheel Model. Each of the qualities, every segment has its own components. In addition, he proposes some aspects to adapt the framework for smart cities. International Business Machines (IBM) provides a concept to create smart cities. It highlights key aspects for smart cities (e.g. people, infrastructure and operations) that establishes three service dimensions in the cities such as human services that includes health, social programs for citizens, infrastructure services that

comprises of water, planning, transportation and management that allows to manage the city and make the governance better in planning the natural resources as shown in the Figure 6.6.

According to the IBM's model [16], a city is mainly based on three pillars, where each of the pillar is divided into three sub-pillars and sum of these pillars makes the foundations of a city. The world is moving rapidly towards cities. In this highly competitive environment, the cities are changing and evolving continuously to achieve the sustainable environment. We are standing at an important junction in the evolution where new forces are enabling new ways for the urban areas to work in a smart environment.

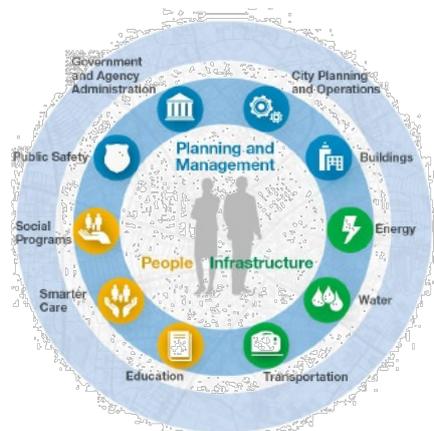


Figure 6.6: IBM's Nine Pillars Model [21]

6.2.1.5 Paradigm

Many researchers have highlighted different characteristics that shows the potential to achieve sustainable goals for smart cities. The EU project describes the relation of these characteristics as shown in the Figure 6.5. The models described in the above section describes the concept of smart cities with different attributes but most of the models have similar characteristics. The six dimensions are overlapping in the majority of models e.g. Smart HL, Smart Governance, Smart Economy, Smart Mobility, Smart Environment and Smart Living [1]. The Figure 6.7 shows six paradigms as described below:

Smart Economy includes factors surrounding the economic competitiveness such as innovation, competitiveness, entrepreneurship, trademarks, productivity and flexibility of the labour and socially responsible.

Smart People explains the level of qualification of the citizens and the quality of social interactions for integration and public life and the openness. It also includes the aspects of creativity, open-mindedness to the external environment and participation of citizens in the public life.

Smart Governance consists of several aspects of political participation such as participation of public in the decision making processes, public and social services for citizens as well as the functioning of the administration in a transparent way. The emergence of ICTs is promoting the cities to achieve smartness in the governance.

Smart Mobility highlights the local and international aspects and the availability of information and communication technologies infrastructure using the modern and sustainable transport systems.

Smart Environment is described by natural conditions such climate, green spaces, public parks in the cities with no pollution and efficient resource management to protect the environment.

Smart Living consists of several aspects of quality of life such as culture, health, safety, housing, tourism etc. Nowadays, smart homes and buildings connected with IoT paradigm in the smart cities are progressing to achieve smart living.

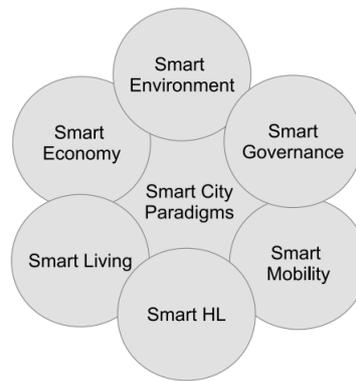


Figure 6.7: Smart City Paradigms [1]

6.2.1.6 Benefits of Smart Cities

The technologies used in the smart cities have untapped potential to improve the quality of life in the urban areas. The smart cities model and paradigms emphasise to utilize the data and technology in a meaningful way for making better decisions in the cities and enhance the quality of people's life. Apart from the benefits in terms of security, hygienic environment, time, connected society, job opportunities and significant improvements can be done in the environmental sector. The gradual increase of awareness around the world, in developed countries, is creating benefits across the several sectors to improve the quality of life and make the environment more sustainable. The main benefits of smart cities are listed in the Figure 6.8.



Figure 6.8: Smart City Benefits [24]

Smart cities are enabling municipal governance by providing efficient social security to maintain the living standards. With the network of smart sensors, city governments are now able to effectively rapid respond in case of emergency situations in the towns or cities. It has the potential to manage the utilities in a highly optimized way through the use of smart technologies. City planners are able to create environment using the effective approaches such as the participation of citizens in the decision process. Smart cities are also bringing benefits for the citizens such as efficient transportation system to reduce the commute times and providing the optimized routes to citizen for reaching their destinations in a convenient way.

Citizens are able to get the high quality of education, governments are empowering and encouraging their citizens to meet gap of skilled workforce in a new competitive environment. Healthcare facilities are being introduced to meet the basic needs of citizens. The citizens are now able to access the government services in an easy and fast way using their smart phones. On one side, smart cities are improving the quality and lifestyle of their citizens by providing quality public services and enabling government institutions to work

in a more efficient manner. On the other side, smart cities are contributing to create a sustainable economic development. It creating new business opportunities and avenues for the investors. A green environment is attracting tourists across the world. Highly optimized and efficient technology in the agriculture sector is producing hygienic food for citizens. The list of benefits goes long with every sector in the smart city.

6.2.1.7 Sustainable Goals

The United Nations's General Assembly set 17 global Sustainable Development Goals in 2015 for the year 2030. Each of the 17 Sustainable Development Goal's have a list of 169 targets which is further measured with 1-3 indicators. There are 232 indicators that will measure the goals. All of these Sustainable Development Goals (Fig.6.9) can be achieved by smart cities. However, how much smart cities can contribute to the global sustainability is still an open question [45]. It requires more research, comparative studies and evaluations.



Figure 6.9: Sustainable Development Goals [40]

The smart cities has the potential to meet these sustainable goals. How much smart cities can contribute to achieve the global sustainability? is an open question. It requires more research, comparative studies and evaluations.

6.2.2 Network Function Virtualization

6.2.2.1 Introduction

Service providers provide more than only a network connectivity for their business customers. They also offer additional services and network functions like encryption, firewalls, Network Address Translation (NAT), Domain Name Service (DNS), caching and others. Traditionally these network functions were deployed by using proprietary hardware at the customer locations. This approach is costly and makes upgrades difficult. Every time a new network function is added to a service, new hardware is needed to install at the customer premises. Because of that, service providers began exploring ways to speed up deployments and reduce cost through Network Function Virtualization (NFV) [3]. NFV is a big subject in the the telecommunication industry and becomes a forward driver for considerable transformations in the network industry.

6.2.2.2 Definition

NFV separates network equipment like an encryption or firewall from dedicated hardware and moves them to virtual environments [11]. Alternatively of installing expensive proprietary hardware, service providers can simply buy inexpensive switches, storage and servers to run virtual machines that performs network functions. This brings multiple functions together into a single physical server.

By using NFV, if a user wants to add a new network function, the service provider can simply install a generic server at the customer premises that uses a standard IT virtualization platform like OpenStack or VMware to start up virtual machines for each

network function[3]. As illustrated in Figure 1.3, physical CPE is going to be replaced by virtual CPE. The server that runs the virtual machine is called a compute node. The compute node includes software called a hypervisor that manages the virtual machines and the resources for the compute node.

In the service providers data center, a virtualized infrastructure manager or VIM, manages multiple compute nodes. The service provider also needs to run the EMS software. EMS is a software that manages the network functions. The EMS servers are also virtualized and run on a compute node in the data center. Network providers have in reality multiple data centers that serve hundreds or thousands of customers. When a network function is deployed for a customer, it can be done anywhere in the network, at the edge, at the closest data center or in a centralized data center. NFV can be distributed to multiple locations in a provider network, including a mix of data centers and customer locations. Some functions like a firewall or encryption are best at the customer location other function like DNS can be done in a data center.

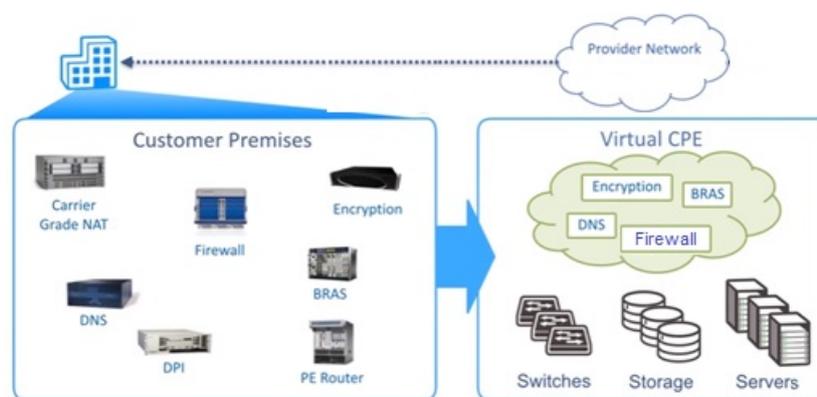


Figure 6.10: Virtual CPE approach [3]

6.2.2.3 History of NFV

In October 2012 a number of the worlds biggest Telecommunication Service Providers (TSP) started a collaborative work on NFV and wrote a white paper which is calling for research and industrial action. Seven of the these operators(BT,AT&T, Orange, Deutsche Telekom, Telecom Italia, Telefonica and Verizon) selected in November 2012 the European Telecommunications Standard Institute (ETSI) to be the home of the Industry Specification Group for NFV (ETSI ISG NFV)[30].

In the first two years between 2013 and 2014 the ETSI ISG published the first five ETSI Group Specification(GSs) documents which four of them were design to get an understanding about NFV in the industry. They included NFV virtualization requirements, use cases, architectural framework and terminology. Between 2015 and 2016 they published documents which prioritized the key capabilities for NFV and defined them to requirements, interfaces and information models. Currently the ETSI are working on their third NFV release which will be soon published and is going to include policy management, security management, charging billing and accounting and several more [13].

The implementation of NFV has been more difficult than anticipated because of the complexity of the technology and the deficiency of interoperability between different platforms and vendor solutions. However things are changing, vendors are figuring out possibilities to work together to encourage interoperability and service providers start to adjust their team to be more efficient along with retraining its workforce to be make them more skilled at virtualization [14].

6.2.2.4 NFV Architecture

According to the ETSI the NFV architecture contains different components like Virtual Network Functions (VNF), NFV Infrastructure (NFVI) and the NFV MANO [37]. In this section these components are defined.

A) *NFVI*

The NFVI provides the environment to deploy VNFs. It's a combination between software and hardware resource. The physical resources are hardware storage, commercial-off-shelf (COTS), network which is made up of nodes and links. and they offer the connectivity between VNFs.

Virtual resource is the approach of storage, computing and network resource together. A virtualization layer, which is based on a hypervisor, uncouples the virtual resources from the underneath located physical resources. The storage and computing resources are represented in the data center with on or more Virtual Machines (VMs). Virtual Networks include virtual nodes which has either a hosting or routing function and virtual links that provide a logical interconnection between two virtual nodes.

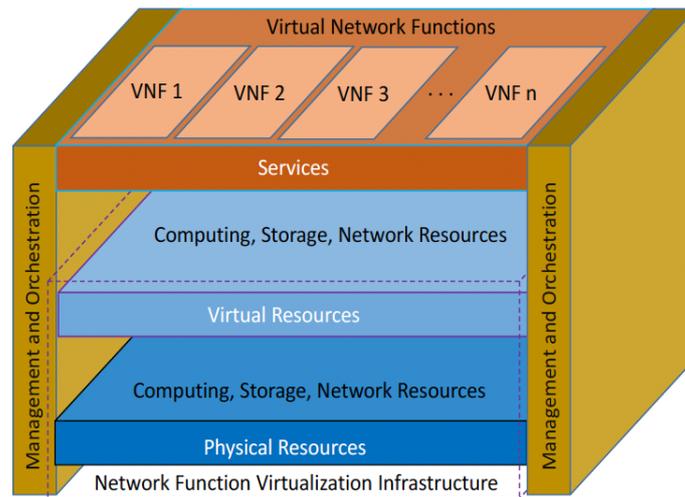


Figure 6.11: NFV Architecture [37]

B) *Virtual Network Functions (VNFs)*

VNFs perform network functions such as firewalls, Dynamic Host Configuration Protocols (DHCP) servers, and Residential Gateway (RGW), which are deployed on a virtual server like a VM. An individual VNF can consist of several internal components and can also be deployed over various VMs, where a VM owns a single piece of the VNF [30]. In this direction, Telecom Service Providers (TSP) can offer a service which consist of one or several network functions. In order of NFV, the network functions are deployed and virtualized on virtual resources. For the user there shouldn't be any performance differences whether the function is running on dedicated hardware at the users premise or VMs.

C) *NFV Management and Orchestration (NFV MANO)*

As reported by the ETSI's MANO framework [12]. NFV MANO brings the functionality necessary to provide the VNFs and the associated operations, like the configuration of the VNFs and the framework these functions run on. It consists of the lifecycle management and orchestration of software and physical resources which assist the infrastructure virtualization and the lifecycle management of VNFs. Furthermore it incorporates databases that are applied to store the information and data models which determine lifecycle properties and deployment of services, functions and resources.

NFV MANO is responsible for all the required virtualization-specific management functions needed in the NFV framework. The framework characterizes interfaces which can be used for the interaction between different parts of the NFV MANO and also coordinates with conventional network management systems like Business Support Systems (BSS) and Operations Support System (OSS) to make management of traditional equipment and VNFs possible.

6.2.2.5 NFV as key Enabler for 5G networks

NFV is a key enabler for the future 5G network. It makes the 5G network slicing possible. Network slicing is a virtual network architecture that enables multiple virtual networks to be built atop a shared physical infrastructure [39]. Network slicing will allow to split a traditional physical network into several virtual networks which can serve different customer segments such as illustrated in Figure 1.11 or individual Radio Access Networks (RANs). The different network slices are going to be separated from each other in the user and control plane.

The benefits of 5G NFV network slicing will be the lower latency, increased broadband, bandwidth, mobility, security, availability and resiliency. Operators will have the flexibility to distribute speed, coverage, and capacity to logical slices dependent on the different use cases [39].

6.3 Opportunities for NFV in Smart Cities

In this section we are going to show the impact and opportunities that NFV will have on cities in the future and how NFV contribute to Smart Cities. Especially the opportunities of NFV in segments like transportation, health care , green environment and take a closer look on real world use cases like Singapore and Bristol which already took a major step in the direction of Smart Cities. Of course there will also be challenges that get along with the implementation of NFV and the realization of Smart Cities which also have to be discussed.

6.3.1 NFV and SDN the key Enablers

If we want to understand how cities are becoming smarter we need, besides of NFV, to have an understanding of Software Defined Network (SDN) and the Internet-of-Things (IoT) [36].

SDN is a network architecture which makes it possible for the network to be intelligently, centrally controlled or programmed by using software applications and operators can regardless of the underlying technology, manage the entire network consistently [8]. SDN makes real time responses to network needs possible. If for example an explosion occurs in the city, SDN enables administrators to reroute bandwidth to give greater throughput of live video, audio, photo etc.

NFV such as discussed in previous chapters is going to virtualize network functions which are currently being implemented by dedicated network hardware. In a Smart City environment, NFV allows standard hardware to run on virtualized environment. This fact in combination with SDN makes it possible to direct network resources more precise between data centers/cloud servers and IoT devices.

IoT is the technology that connects daily used devices like street lights, parking meters, refrigerators etc. to the internet. Thereby its going to be possible for the devices to communicate with real time data inside the network. In regards to Smart Cities, the IoT devices are going to supply the real time feedback and the SDN controllers make the real time direction of resources possible.

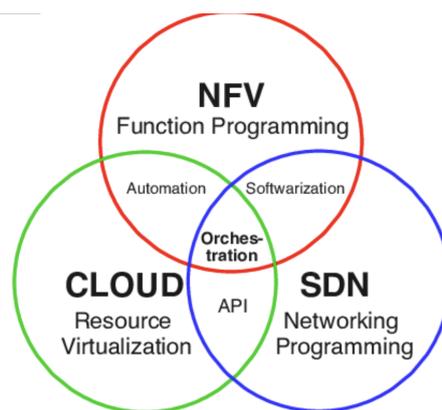


Figure 6.12: Relationship between NFV, SDN and Cloud [31]

NFV and SDN as key enabler together with cloud and Internet of Things are going to change how cities function in the future. In times of a crisis SDN makes redirecting of resources possible and NFV provides the scalability of hardware required to redirect resources. In a city without smart infrastructure, when something happens like for example an explosion in the city, a broad range of data points will stream into a central network area. To transport and receive this huge amount of data, standard computing hardware

will even withstand because of dedicated resources or will break down because those resources are overcharged with peak load. Dedicated hardware can only handle a certain amount until it fails.

In a city with a smart infrastructure, NFV enables actual time scalability of resources through virtualization. If critical compute hardware is virtualized, SDN controllers are capable of immediately increase needed resources to guarantee that ways of communication, under high bandwidth stress loads, do not break down [36]. This increase of bandwidth is especially necessary in regard of the increasing IoT devices, illustrated in Figure 1.6, which would in case of an emergency send information to critical access points. Because of the SDN controllers which can increase the redirect bandwidth needs of the city network, information about the event can be accessed faster and clearer. To make the best out of a smart infrastructure, SDN and NFV are dependent on high quality IoT devices distributed on the ground [36].

6.3.2 Benefits

The usage of NFV in combination with related fields like SDN can have great benefits for cities to reach the sustainable city goals. These are a few of the most important:

- Through consolidation of network appliances, sharing computer resources, reduction of energy consumption, and use of lower hardware it will be possible to reduce the the capital expenditure and operational expenditure among a host of other things. Service Providers are able to save a lot on costs and time through a more efficient network process [11].
- NFV reduces time to market of new services by changing the innovation cycle of operators through software based deployment and through swift introduction of tailored services based on customer needs. Reduction in time to market does also comes with gains on operational efficiency for both providers and customers. This enables new vendors to enter the market faster and create a competitive environment under the service providers, which will lead to lower prices for the consumer [3]
- NFV in combination with cloud technologies can adopt tools to automate operations and management. Service providers stand to gain by implementing NFV as they meet the needs of communications market through automated scaling of resources, faster service introduction, and optimum utilization of the allocated resources [30].

6.3.3 Opportunities

Transportation, environment pollution and providing sufficient health care are some of the biggest issues smart cities are facing. NFV can be a solution for some of these issues in larger cities. We are going to have a look at three opportunities that are being enabled by NFV in smart cities and see how it can help resolving and innovate.

6.3.3.1 Transportation

Traffic has always been a major issue in larger cities. There has been made huge efforts to road transportation and public transportation but with urbanization process it is still a growing issue since city administrations can not keep up with the pace. In Zurich for example the average citizen was stuck in traffic jam for about 51 hours [33]. This does not only reduce the life quality of the citizens dramatically but it also has a big economical impact, due to the time that could have been used more productive.

Through NFV and the usage of lasers, GPS, advanced cameras, wireless communication and more navigation in cities can be made easier and accidents can be avoided.[37] Using

algorithms and functions on the collected data could lead to scenarios where tools predict traffic jam before it ever happens and would lead drivers to specifically calculated routes to avoid the creation of traffic jam. One other interesting applications is the use of the shared data in conjunction with autonomous driving. The current implementation of autonomous driving in cars like Tesla, Mercedes or Volvo relies only on data which is being collected by the sensors of the car itself and some data provided by navigation system. This implementation is certainly not secure, since the biggest threat on the road is the unpredictability of road users. Autonomous driving where data is shared between all road users and cars could reduce this threat almost completely and make driving a lot safer.

These technological advances would unquestionably make private cars more attractive over public transport or car sharing alternatives. But contradictory to this many cities are trying to reduce the attractiveness of private cars and make public transportation the number one way of transportation in cities. For the urban transportation of the future reducing congestion and pollution through the use of smart technologies is key, but there has to be found the right balance between the usage of private cars and other alternatives.

6.3.3.2 Healthcare

The technological advances in healthcare has been huge in the past few decades. Life expectancy has risen over 20 years in the past century. But all these advances come at a cost. The healthcare cost have exploded in many countries. In fact, in Switzerland the rapidly rising cost of healthcare is one of the biggest worries many citizens have and many middle class families need financial aid from the government to pay the premium. The use of NFV could potentially reduce these costs.

One solution could be E-health. E-health itself is a solution, which aims to use new technology in healthcare. It contains several solutions for hospital challenges, from the point of view of healthcare professionals, doctors and politicians e-health will be a way to promoting healthcare industry especially for those living with chronic conditions [1]. Many solutions and projects are under development in healthcare in several axes like: telecare, telemedicine or even telesurgery: it is a set of services carried out off location. Services typically include teleconsultation and teliagnosis, which lets doctors perform diagnostics with instruments without the presence of the patient or even let doctor perform surgery without their physical presence. M-Health is a form of e-health that improves the communication, the sensing, the monitoring part of health data, in order to provide real time information and results to patients, researchers and doctors, which can react if any signs of potential diseases are shown [7]. Technology trends can also be used to store, research and analyse data of patients with unexplored diseases. Through Artificial Intelligence or other tools the historical data of those patients could be analyzed and evaluated.

Through these types of electronic health solutions it will not only be possible to improve the health conditions of many people but also to potentially save many lives. At the same time, these technological advances rely on data being both collected and shared. Privacy issues and the medical mystery are in fact that are most often headlined in discussions about barriers to incorporate technology into various areas in healthcare.[2] It is necessary that the legislation adapts to these technological changes and opens up the way for these solutions to be deployed and adopted.

6.3.3.3 Green Environment

There has been an increased awareness of the negative effects of CO₂ in the past few months, especially also in Switzerland where many young students have participated in

many protests against the current climate policy and to increase awareness of the disastrous effects climate change will have in the next centuries.

NFV could potentially open new ways of distributing and saving energy. One the one hand to generate renewable energy based on system like wind power generation and solar power generation. On the other hand by connecting things and using Artificial Intelligence techniques, other possibilities show up like making the energy generation, storage, consumption more smarter, efficient and better. For creating an intelligent energy management, it is necessary to create a smart grid capable by routing energy to end users in an efficient way using the already existent ways of communication. For a smart grid real-time and reliable information become key factor for delivery of power from the generating units to the end users. The impact of capacity constraints, hardware failures, and natural catastrophes, which can cause power disturbances and outages, could be avoided by online power system condition monitoring, diagnostics and protection. Till today the intelligent controlling and monitoring systems by modern information and communication systems have become essential to realize the envisioned smart grid [25].

6.3.4 Real World Use Cases

Many cities are trying to keep up with the rapid technological advances made. But only a few of them are able to keep up with the pace and stay on the top. Reason being laws, lack of resources or social barriers. NFV in conjunction with SDN however could provide many benefits to Smart Cities, which may result in wider digitalization in cities.

Although the technology being in rather new, we see first efforts being made to use NFV in Smart cities and profit from the benefit it provides. We are going to discuss two use cases of leading smart cities, where in one case NFV is and in the other one could potentially change the way we interact and connect with people in smart cities.

6.3.4.1 Singapore

Singapore as one of the most developed countries in Asia often is called the smartest nation in the world. The nation has been using a wide variety of technologies to improve the life of its citizens. Chia, Eng Seng shows several examples on how Singapore is using ICT in smart ways [5].

- In many nations especially in leading industrial nations owning a private car. In Singapore, however, owning a car is a privilege only the richest can afford. Even a normal Volkswagen Golf can cost more than 100'000 \$, reason being the high taxes on cars. Therefor public transport or car sharing services are very popular among citizens, which reduce the overall traffic problematic drastically.
- Singapore is considered one of the safest nation in the world. There more than 50'000 police surveillance cameras, which make sure that violence and burglary is almost nonexistent in Singapore.
- The city is providing healthcare service for elderly citizens through a range of technologies. These are digital service platforms as well as remote monitoring devices, which send data directly to doctors or hospitals.
- With the use of ICT many interactions between the government and citizens have been automated or digitized. Therefore, the processes are often simple, fast and efficient.

Despite being called the smartest nation in the world, we were not able to find any literature or examples of NFV being used in Singapore. But knowing the benefits and

the opportunities NFV provides and the role of NFV as the enabler of smart cities of the future. A short search on glassdoor for NFV roles in Singapore shows more than 20 open positions. There are certainly efforts being made and we may see NFV being deployed and used in Singapore as an early adopter very soon.

6.3.4.2 Bristol Is Open

Bristol Is Open is a joint venture between that English city's council and the University of Bristol, and more specifically its Electronic Engineering Department. Bristol itself is a city and county in Southwest England with a population of about 500'000. As far as we know, it is the only example of network functions virtualization and Software defined Networks being applied to a smart city. It has been recognized by the UK Smart Cities Index 2017 as the U.K.'s leading smart city. Most of the information provided about the project can be found under the projects website [44].

The Electrical Engineering Department of the University has a high-performance lab with an NFV and SDN implementation based on open source, with more than 40 researchers dedicated to the project. The infrastructure is provided by an external startup called InterDigital [4]. Bristol Is Open is creating a set of platforms on which to run smart city applications and rolling out nodes around the city. One is already connected to a super computer while others are linked to LTE equipment and running multiple input, multiple output-based Wireless Local Area Networks [42].

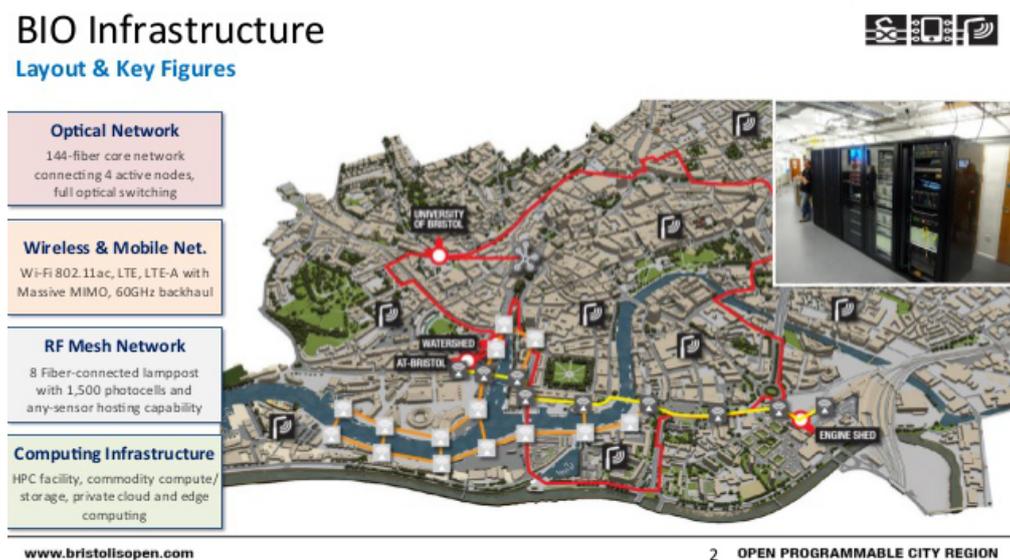


Figure 6.13: Bristol Is Open Infrastructure[44]

This experimental node is colocated with Bristol Is Opens operational platform for the smart city, which is based on the fiber optic infrastructure and ducts bought by Bristol Is Open from a cable company which has went out of business. This in combination with two new wireless technologies used to virtualize the networks. As seen on figure 1.7 they have located different access points in the city on which the devices can connect to. The devices then are able to share data in much faster, more efficient and simpler way. In a first test run, they were able to bring live streaming to another level with [4]. With current live streaming technologies those results are not able to be reproduced.

The first successful milestones of implementing NFV and SDN in smart city like Bristol proves that the technology can be used with all its benefits. These type of projects will certainly lay the foundation for new projects regarding NFV and SDN in smart cities, where it may go even a step further and replace existing physical network installations.

6.3.5 Challenges and Discussion

Even though NFV enables many opportunities and provides great benefits for future smart cities, it still has to overcome the many challenges it faces in its early age. To reach the goals several issues have to be taken into account. In this section we discuss the most important challenges researchers all over the world are facing and trying to solve to

6.3.5.1 Scalability

One of the benefits of NFV is the scalability it provides. Nevertheless, if we consider the huge amount of data relevant to cities like Berlin, London or New York, where an enormous amount of devices will be expected to simultaneously connect in a limited coverage the huge load relevant to smart cities, then scalability becomes one the major concerns to avoid network overloading and congestion [6]. In addition, when focusing on areas where sensors and all actuators need to communicate under strict latency requirements like in telesurgery, in which any disruption or latency spikes could lead to fatal consequences, then it is necessary to make sure, that the networks availability is granted at any time.

6.3.5.2 Energy Efficiency

Energy bills represent more than 10% of TSPs OPEX [28], therefore reduced energy consumption is one the key benefits of NFV. The argument is that with the flexibility and ability to scale resource allocations up and down as traffic demands increase and decrease, providers could potentially reduce the number of devices operating at any point in time and ergo reduce their energy bills. On the other hand, NFV will likely make data centers an integral part of telecommunication networks. According to an analysis in the SMARTer 2020 report [19], if the cloud were a country then it would rank 6th in the world in terms of its energy demand and still this demand is to be increasing by 63% by 2020 [20]. Therefore, it is necessary to study whether NFV will meet its energy savings expectations or whether the NFVs energy consumption will just be transferred to the cloud.

6.3.5.3 Security and Privacy

Despite the enormous potential of NFV, concerns regarding issues of privacy, security do present a major issue for the deployment of NFV. Therefore, cloud privacy issues will be among the key concerns for network service providers, if they have to move to public clouds. Because the functions to be virtualized represent subscriber services, confidential personal information may be transferred to the cloud. This creates challenges since the functions are distributed, making it hard to know where this data is located and who has access to it. In the case where the functions are deployed in third party clouds, users and Telecom service providers would not have access to the physical security system of the data centers. Even if the service providers do specify their security requirements, it will most likely be hard that they are respected. A virtual appliance should be as secure as a physical appliance if the infrastructure is secure. Network operators will be seeking tools to control and verify hypervisor configurations. Several types of vulnerabilities compose a security threat in this context [27].

- **Isolation Failure Risk:** This is the case when an attacker manages to break into a hypervisor by compromising some VNFs running over it. This attack can impose risks if carried out successfully. In this attack scenerio, the attacker first compromises VNF by gaining access to the operating system. Using different tools and VNF network connectivity with the cloud management network, the attacker gains access to the hypervisor management API and then the attacker breaks into

the hypervisor to cause great damage. These are possible due to the insufficient isolation between hypervisors and VNFs.

- **Network Topology Validation and Implementation Failure:** Using NFV, virtual networking components like virtual routers or virtual networks can be created. Quick decisions can result in human error when a virtual router is created and used to interconnect virtual networks without the use of any firewall, which then creates a surface for attacks.
- **Denial of Service Protection Failure:** DoS attacks could be directed to virtual networks or VNFs public interfaces to impact service availability and exhaust the network. A enormous amount of traffic from a compromised VNF can be generated and sent to other VNFs that are running on the same hypervisor.
- **Malicious Insider:** These are internal security risks by damaging actions of internal administrators. One attack scenario could be a malicious administrator takes the memory dump of a users VM. Since the malicious administrator has the root access to the hypervisor and by using a search operation, they can extract confidential user data.

Security & Privacy regarding NFV in Smart Cities is not only a topic that is widely covered in academic literature due to its risks and potential danger it creates. But also in the discussion during our presentation of the usage of NFV in Smart cities at the University of Zurich Communication Systems Seminar, we found out that the topic is highly controversial and many people have a different take on this. One opinion which was represented by many students was that even today a lot of our private data is stored on third party services. Those providers are being targeted by many hacker groups, some of these attacks even success full as events in the past have shown. The outcry overall does not last very long and vanishes in the matter of weeks. People generally take convenience over security in the context of private data. This could also be the case for NFV. People may take the benefits and convenience it provides over its the security and privacy issues as shown above. But there were also a few students which did not agree with this opinion. Privacy represents one of the key elements of a humans integrity. Giving up on his own privacy could be seen as giving up ones own integrity. With digitization privacy and security will surely be one of the most highly controversial topics in the future with no right answer to it.

6.3.5.4 Management and Ochestration

The deployment of NFV will challenge current management systems and requires significant changes on how networks are operated, deployed and managed. Such changes are required to provide network and service solutions, but also to exploit the flexibility and dynamism made possible through NFV [23].

This will likely lead to scenarios where functions that provide a service to a given customer are placed on different server pools. The challenge then will be to have an acceptable level of orchestration to make sure that on a per service level, all the required functions are instantiated in a coherent and on-demand basis, so that the solution remains manageable.

6.3.5.5 Modeling of Resources, Functions and Services

The current implementation of network rely on many predefined standards, which all providers The potential of NFV is based on its ability to deliver high levels of flexibility and automation. However, the resources and functions in NFV will be provided by different entities. Therefore, the availability of open, standardized and well understood descriptors

for these resources, functions will be key enablers for larger NFV deployments. Models should consider not only initial deployment but also life cycle management reconfiguration [30].

6.4 Conclusion and Future Directions

People generally move to urban areas in order to satisfy their necessities following their occupations, they like the environmental change and state-of-the-art facilities in urban areas. This urbanization process is also adding new challenges for cities. Technology is playing a vital role in the infrastructure of a city that has a positive impact to improve the quality of life, and it introduces the concept of smart cities. The cities across the world are looking for optimal facilities for their citizens through telecommunication and intelligent solutions to make smart cities a reality. The goal of smart city is to optimize the quality of human-life. [1]

Network Functions Virtualization in conjunction with related fields like SDN are one of the key enablers of smart cities. Through decoupling network functions from the hardware means that architectures can be a lot more easily changed since the requirements can change in the fast paced world of IoT. We have seen areas in this paper where NFV has contributed to but there are still many challenges and problem open to solve till a widespread adoption of smart cities is possible.

Cities are on the edge of evolution. The rapidly changing demographics of cities are bringing new challenges. The growing list of challenges demands the optimized solution of these problems, with the aim to improve the quality of life in urban areas. It has introduced smart cities as a new concept to reach the sustainable goals using smart technologies such as sensor networks, IoT, SDN, NFV and 5G. In the context of a city, the term smart has different dimensions such as technology, people, and community. We have discussed three models for the smart cities, dimensions, paradigms and benefits to create smart cities in the future. Smart cities has the potential to contribute to sustainable development goals. There is a need to implement the virtualization in the networks to build the scalable infrastructures for communication using smart technologies. We have discussed the introduction and definitions of NFV. Some background aspects how a number of the worlds biggest Telecommunication Service Providers started a collaborative work on NFV and wrote a white paper, but the implementation of NFV has been more difficult than anticipated because of the complexity of the technology and the deficiency of interoperability between different platforms and vendor solutions. We have described the architecture components of NFV like VNFs, NFVI and NFV MANO. Also, NFV is a key enabler for the future 5G network that makes the 5G network slicing possible. The opportunities for NFV in the smart cities are highlighted in the report e.g., how the NFV and SDN as key enabler together with cloud and Internet of Things are going to change how cities function in the future. It will be possible to reduce the the capital expenditure and operational expenditure through consolidation of network appliances, sharing computer resources, reduction of energy consumption, and use of lower hardware. NFV will reduces time to market of new services by changing the innovation cycle of operators and NFV in combination with cloud technologies can adopt tools to automate operations and management. The opportunities for green environment, transportation and healthcare are discussed in the report. Some real world use cases such as Singapore as one of the most developed countries in Asia often is called the smartest nation in the world. Although NFV enables many opportunities and provides great benefits for future smart cities, it still has to overcome the many challenges it faces in its early age such as such as scalability, energy efficiency, security and privacy. In the end, we conclude our work with future directions.

Bibliography

- [1] A. Arroub, B. Zahi, E. Sabir and M. Sadik. *A literature review on Smart Cities: Paradigms, opportunities and open problems*. International Conference on Wireless Networks and Mobile Communications (WINCOM), 2016, pp. 180-186.
- [2] Abelson J. S., Kaufman E., Symer M., Peters A., Charlson M., and Yeo H. *Barriers and benefits to using mobile health technology after operation: A qualitative study* . Surgery, vol. 162, no. 3, 2017, pp. 605-611
- [3] Bo Gowan ,2016, *What is Network Function Virtualization (NFV)?* <https://www.ciena.com/insights/articles/What-is-NFV-prx.html>, last visit April 15, 2019
- [4] Bristol is Open ,2018, *Bristol Is Open and InterDigital - A Smart City partnership*. https://www.youtube.com/watch?v=ZPfEk_UShrY&t=100s, last visit April 15, 2019
- [5] Chia, Seng E. *Singapore's smart nation program - Enablers and challenges*., Conference: 11th System of Systems Engineering Conference, 2016, 1-5.
- [6] Condoluci M., Sardis F., Mahmoodi T. *Softwarization and Virtualization in 5G Networks for Smart Cities*. Mediterranean University of Reggio Calabria, 2015, Italy, Department of Informatics
- [7] Cook D. , Duncan G., Sprint G., Fritz R. *Using Smart City Technology to Make Healthcare Smarter*. Proceedings of the IEEE. 2018, PP. 1-15.
- [8] Ciena. *What is SDN*, 2017, from <https://www.ciena.com/insights/what-is/What-Is-SDN.html>, last visit April 16, 2019
- [9] Daily Energy Insider. *EEI's new board chairman cites smart-city opportunities as convention gets under way*, from <https://dailyenergyinsider.com/featured/5732-eeis-new-board-chairman-cites-smart-city-opportunities-convention-gets-way/> , last visit April 18, 2019
- [10] Eger J.M.. *Smart Growth, Smart Cities, and the Crisis at the Pump A Worldwide Phenomenon*. ACM The Journal of E-Government Policy and Regulation, 2019, pp. 47-53
- [11] ETSI NFV ISG. *Network Functions Virtualization*. from https://portal.etsi.org/nfv/nfv_white_paper.pdf, last visit March 12, 2019
- [12] ETSI, N.. *Network Functions Virtualization (NFV)*. Management and Orchestration. NFV-MAN, 2014, 1, v0.
- [13] ETSI *Network Functions Virtualization (NFV)* , from <https://www.etsi.org/technologies/nfv>, last visit April 15, 2019
- [14] ETSI *NFV Reality Check - How is NFV Being Implemented Today?*, from <https://www.sdxcentral.com/resources/sponsored/ebriefs/att-nfv-reality-check/>, last visit April 15, 2019

- [15] Geller, A. L. . *Smart growth: a prescription for livable cities* , American Journal of Public Health, 2003, 93(9)
- [16] Cohen B. *Key Components for Smart Cities* from <http://www.smartbrantford.ca/TheSixComponents.aspx>, last visit January 15, 2014
- [17] Gharaibeh A., Salahuddin M. A., Hussini S. J., Khreishah A. *Smart Cities: A Survey on Data Management, Security, and Enabling Technologies*. IEEE Communications Surveys Tutorials, 2017, pp. 2456 - 2501
- [18] Giffinger, R., Fertner, C., Kramar, H., Meijers, E. *Smart cities: Ranking of European medium-sized cities*. 2017, Vienna University of Technology
- [19] Global e-Sustainability Initiative (GeSI) (2015). *SMARTer2020.*, from <http://gesi.org/SMARTer2020>, last visit April 17, 2019
- [20] GREENPEACE (2014). *Clicking clean: How companies are creating the green internet.*, Washington, DC, USA
- [21] IBM, *Smart cities* ,from https://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/, last visit April 18,2019
- [22] Industry Canada.*Report of the Panel on Smart Communities*, Ottawa, Canada: Government of Canada 1998.
- [23] Keeney J., Meer S. v. d., Fallon L. *Towards real-time management of virtualized telecommunication networks*, in Network and Service Management (CNSM).10th International Conferenc, 2014, pp. 388-393
- [24] Kyoto Smart City Expo (2017). *IoT practices and Evolution in the Smart City* last visit April 18, 2019 https://expo.smartcity.kyoto/doc/ksce2017doc_2-51.pdf
- [25] Lakshminarayana S. and Anjul. *Smart grid technology and applications*, 2014 POWER AND ENERGY SYSTEMS: TOWARDS SUSTAINABLE ENERGY, 2014, pp. 1-6.
- [26] Lee J. H., Hancock M. G., Hu M-C, (2013). *Towards an effective framework for building smart cities: Lessons from Seoul and San Francisco*, Technological Forecasting and Social Change
- [27] Lal S., Taleb T. and Dutta A. *NFV: Security Threats and Best Practices*, in IEEE Communications Magazine, vol. 55, no. 8, 2017, pp. 211-217
- [28] Lucent A., Hill M., NJ, *Global What if Analyzer of NeTwork Energy ConsumpTion (GWATT)*. Bell labs application able to measure the impact of technologies like SDN & NFV on network energy consumption, Bell Labs, Alcatel Lucent, 2015, White Paper
- [29] Lynn, L. E., Heinrich, C. J., and Hill, C. J. *Studying governance and public management: Challenges and prospects*, *Studying governance and public management: Challenges and prospects*. Journal of Public Administration Research and Theory, 10(2), 2000, 233-262
- [30] Mijumbi R.,Serrat J.,Gorricho J., Bouten N., De Turck F. and Boutaba R. *Network Function Virtualization: State-of-the-Art and Research Challenges*, in IEEE Communications Surveys and Tutorials, vol. 18, no. 1, 2016, pp. 236-262,

- [31] Nathan F., *Illustration of relationships among NSO, NFV, SDN, and Cloud*, from https://www.researchgate.net/figure/Illustration-of-relationships-among-NSO-NFV-SDN-and-Cloud_fig5_323867726, last visit April 15, 2019
- [32] Nam T., Pardo T.A. *Conceptualizing Smart City with Dimensions of Technology, People, and Institutions*, Proceedings of the 12th Annual Digital Government Research Conference, 2011, pp. 282-291
- [33] NZZ (2018). *In Zuerich stand man 2017 nur noch 51 Stunden im Stau*, from <https://www.nzz.ch/mobilitaet/auto-mobil/in-zuerich-stand-man-2017-nur-noch-51-stunden-im-stau-ld.1355136>, last visit April 16, 2019
- [34] OpenLearn, Smart Cities from <https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=67877>, last visit April 18, 2019
- [35] Park H.-Y., Park J.-S, Yang J.-I, Lee J.-H. *A proposed methodology for ubiquitous city service development: service reusability perspective*, Serv. Sci. 2 ,2009 111126.
- [36] QuoteColo (2016) *Smart Cities: How SDN and NFV are changing the Way We Live*, from <https://www.quotecolo.com/smart-cities-how-sdn-and-nfv-are-changing-the-way-we-live/>, last visit April 16, 2019
- [37] Sandeep M.K. & Dr. Prabhu.J. *Analysis of Network Function Virtualization and Software Defined Virtualization*, Internation Journal of Informatics Visualization, 1(4),2017, 122-126,
- [38] Sderstrm O., Paasche T., and Klauser F. *Smart cities as corporate storytelling*, City: analysis of urban trends, culture, theory, policy, action, 18:3,2014, 307-320
- [39] SDxCentral *How 5G NFV Will Enable the 5G Future* Retrieved April 18, 2019, from <https://www.sdxcentral.com/5g/definitions/5g-nfv/>, last visit April 18, 2019
- [40] United Nations *Sustainable Development Goals*, from <https://www.un.org/sustainabledevelopment/sustainable-development-goals/>, last visit April 18, 2019
- [41] UN Population Division *World Urbanization Prospects 2018*, from <https://population.un.org/wup/General/DefinitionIssues.aspx>, last visit April 18, 2019
- [42] Wilson P. *a project to make Bristol an open, programmable city*, from <https://www.youtube.com/watch?v=YyUAXWn6A2E>, last visit April 16,2019
- [43] Yaqoob I., Hashem I. A. T., Mehmood Y., Gani A., Mokhtar S. and Guizani S. *Enabling Communication Technologies for Smart Cities*. IEEE Communications Magazine, vol. 55, no. 1, 2017, pp. 112-120
- [44] Bristol is Open <https://www.bristolisopen.com/>, last visit April 13,2019
- [45] Role Of Smart Cities In Sustainable Development, from https://www.ripublication.com/irph/ijert_spl17/ijertv10n1spl_09.pdf, last visit May 30, 2019

Chapter 7

Investigating the Blockchain Technology in the Context of Cybersecurity

Lenz Baumann, Roland Schlaefli, Silas Weber, Pascal Zehnder

The ability of the Blockchain technology to handle tasks in a decentralized manner and its property to leave an audit trail of all its history makes it a perfect match when addressing cyber security related issues. Whether it directly mitigates them or it is used to create more secure applications, the Blockchain has a lot to offer. This paper starts with a brief introduction to Blockchains in general and continues with an overview of the various approaches and ideas that exist for the integration of the Blockchain technology with cybersecurity to increase overall security of applications or a whole scenario.

Contents

7.1	Introduction	175
7.2	Background	176
7.2.1	Cybersecurity	176
7.2.2	Blockchain	177
7.2.3	Smart Contracts	179
7.3	Blockchain Applications in Cybersecurity	182
7.3.1	Distributed Denial of Service (DDoS)	182
7.3.2	Public Key Infrastructure (PKI)	187
7.3.3	Internet of Things (IoT)	191
7.3.4	Specific Blockchain Use Cases	193
7.4	Final Considerations	199
7.4.1	Summary	199
7.4.2	Discussion	199

7.1 Introduction

Since the initial development of computing and communication systems, their security has been an ever-growing concern. More and more data is being collected and shared between entities and third-parties [?]. Protecting this data and the underlying systems is of utmost importance for our internet-based economy, as companies are bound by increasingly strict laws and need their customers to be confident in data privacy.

The evolution of Blockchain systems over the past decade has led to research on a broad range of application areas in computer science as well as other contexts. Amongst many others, the applications of Blockchain technologies to secure systems and data has been an area of very recent research [?]. The primary goal of this work is, therefore, to provide an overview of key areas where such technologies could be applied in future systems.

To establish a solid foundation, Section 7.2 provides a summary of the most important basic concepts and technical background. More specifically, Section 7.2.1 goes into the traditional concepts of cybersecurity and the threat models that have been developed in the area (*e.g.*, the CIA triangle [1]). Section 7.2.2 then presents the basics of Blockchain on the example of Bitcoin, while Section 7.2.3 summarizes the concept of smart contracts, an evolution based on the core infrastructure of the Blockchain.

On a lower, more technical level, this work focuses on the applications of Blockchain for the prevention of Distributed Denial of Service (DDoS) attacks, as well as the improvement of critical parts of the internet infrastructure. DDoS attacks are a possibility for attackers to flood the service of a company. Such attacks can lead to availability issues (*e.g.*, causing service downtimes) and thus result in significant business forfeits. Approaches to protecting networks against such DDoS attacks using Blockchain technologies are summarized in Section 7.3.1.

A critical part of the internet and communications security in today's environment is the existence of communication methods that participants trust to deliver messages securely and confidently. Public Key Infrastructures (PKI) provide the backbone for many of these methods. Existing PKI architectures often rely on a centralized infrastructure, which enables many kinds of attacks. We provide an overview of possible improvements to PKI using Blockchain technologies in Section 7.3.2.

The emergence of the Internet of Things (IoT) and a growing market for interconnected devices brings exciting challenges for Blockchain use. Section 7.3.3 reviews some of the most critical security and privacy challenges, as well as how they can be mitigated with Blockchain technology. It provides an overview of how Blockchain can help with proving the integrity of IoT data (*e.g.*, smart meters) and goes into how Blockchain can strengthen the security of IoT in the current state of research (*e.g.*, in smart homes).

The goal of this report is not only to cover theoretical concepts or frameworks but also to shed some light on existing real-world applications or proposed system architectures that are currently being developed (presented in Section 7.3.4). These applications show us whether the theoretical concepts apply to the real world as well as what types of problems and issues arise.

7.2 Background

In this section, we build a foundation with regards to the theoretical concepts and background that are needed to follow the concepts presented in Section 7.3. First Section 7.2.1 presents the traditional ideas of cybersecurity as they have been part of research for the past decades. Subsequently, Section 7.2.2 elaborates on the core concepts of the Blockchain which are necessary to understand the main research topic of this work. Finally, Section 7.2.3 reviews the concept of smart contracts and how they have developed as an evolution of the existing Blockchain architecture.

7.2.1 Cybersecurity

Cybersecurity has the potential to be one of the most global interests for governmental or private institutions, as well as for individuals. Political, financial, corporate, military, and medical organizations collect an enormous amount of data on computers and other devices. All those organizations need to make sure that unauthorized individuals or organizations cannot access, modify, or remove this sensitive information. When the media publishes that an unauthorized person has attacked a company, they suddenly lose reputation, which probably leads to the loss of many customers. All in all, the business result of an attacked institution decreases by a significant margin. Therefore, an investment in purposes of cybersecurity provides a significant reduction in business risk [2].

Factors in IT security can be human users (social engineering or insufficient security knowledge), the complexity of the software, and the speed or time pressure to bring software to the market [2]. Generally, there is no optimal solution for security, as every system is eventually subject to attacks [2].

Various fields shaped the history of cybersecurity, such as cryptography which was used by military and diplomacy before World War 2 [2]. Multiple cryptosystems evolved, such as the Enigma machine, which helped the Allies to decipher German radio transmissions [3].

7.2.1.1 CIA Triangle

Data, software, and hardware all over the world need to be protected in a way that guarantees confidentiality, integrity, and availability [4]. Those three security objectives are considered the most important components of security and are modeled into the rear triangle, also called the CIA-triangle. By combining these three aspects, a system becomes valuable for a user. However, there are three possible ways to harm this value by breaching one of the above components, as seen in [5].

- **Confidentiality:** As the use of computers in sensitive fields such as industry, military, or government has grown, keeping information secret has become of utmost importance. Data such as student grades, medical records, and many more are sensitive information that needs to be controlled carefully. Ensuring confidentiality is a problematic operation throughout all computer systems. There always has to be an authorized officer to distribute authorization rights and allow individuals to access data within a company or institution.

Definition: Only authorized users can **view** assets (*e.g.*, a thief gains access to user data).

- **Integrity:** The trustworthiness of resources or data and the prevention of unauthorized or unwanted changes is the base of this component. It usually includes data integrity (information content) and origin integrity (source of data). Mechanisms of integrity may be either prevented or detected. A prevention mechanism blocks

any unauthorized try to modify data. Detection mechanisms, on the other hand, do not prevent violations of integrity. They report that the underlying integrity was compromised. As the source and the trust of data usually relies on assumptions, evaluating the integrity of data is often difficult.

Definition: An asset is **modified** only by authorized users (*e.g.*, a thief gains access to data and modifies its content).

- **Availability:** A resource or information must be accessible to a user at any time. An unavailable system is as useful as having no such system. By guaranteeing a running system, those resources may be utilized to generate revenue for a company or to gain information from data. Denial of service attacks (DoS), in general, are efforts to block the availability of a system and are difficult to detect because unusual access patterns have to be detected. In Section 7.3.1, those types of attacks are further described.

Definition: Any authorized party may **use** an asset (*e.g.*, a thief steals a computer and the user has no longer access).

7.2.1.2 Attack vectors

The main goal of computer security is to protect valuable assets, such as hardware, software, and data. A threat is a potential violation of having security within a system. There are many possibilities to attack a computer system which can be categorized as either direct attacks (sometimes technically difficult to mount) or indirect attacks (*e.g.*, social engineering attacks) [2]. Typical examples include the following: Hoaxes, Bugs, Trojan horses, worm, virus, rootkit, or bots.

7.2.1.3 Terminology

In the literature, the terms cybersecurity and information security are used interchangeably [6]. Security, in general, is hard to define as no universally agreed definition exists. A possible definition of security may be the state in which there is no relevant threat or security breach. As it is not possible to enumerate all possible threats or to verify their nonexistence, security cannot be measured in a meaningful way. Security, in general, is very individual depending on the person and her willingness to take risks, which can be perceived differently by other people. All in all, security refers to protection against **intended** incidents and attacks [5, 4]. Additionally, security has to be disassociated from the term safety, as the latter refers to **unintended** incidents and attacks [5].

The terminology of “Computer Security”, “Information Security” and “IT Security” have been used interchangeably over many years. Going with time, the term “Cyber Security” became more popular, when the former US President Barack Obama proclaimed in 2009, as stated in [7], “I call upon the people of the United States to recognize the importance of cybersecurity and to observe this month with appropriate activities, events, and training to enhance our national security and resilience”.

7.2.2 Blockchain

Connected to the growing popularity of the Bitcoin Cryptocurrency that conquered the markets of the world, the interest for its underlying technology - the Blockchain - grew similarly. The idea of having a fully decentralized, tamper-proofed distributed online ledger, fascinated not only the digital world but also other non-digital areas. Moreover, while Bitcoin gained a bad reputation due to its price fluctuation [8], the Blockchain continued to receive attention from the industry and academia. However, questions arise,

such as if the Blockchain technology is the solution for existing problems or it is a problem itself [9, 10, 11].

In the following subsections, the Blockchain technology shortly explained and discussed. It is not the goal of this section to give full technical insights into the various mechanisms but to provide a base of knowledge for the sections to come.

7.2.2.1 The Blockchain in Detail

The Blockchain is a shared, publicly available online ledger containing a chain of inter-linked blocks that themselves contain transactions between users. [12]. No single user controls, updates, or maintains the Blockchain. Instead, every action performed on the Blockchain requires consensus amongst all users (nodes) [8]. Because its blocks are inter-linked by a cryptographic hash and therefore contain information about the previously included blocks, no changes can be made to the Blockchain without being traceable in hindsight.

The Blockchain is a data structure that allows humans and computers to interact with each other without having to rely upon trusted third parties. Instead, it is the controlling power of the data structure itself or put differently, the consensus amongst all users, that assures the compliance of each party with their part of a deal. As a result, it allows "people who have no particular confidence in each other [to] collaborate without having to go through a neutral central authority" and becomes "a machine that creates trust". [13] [8]

7.2.2.2 Properties of the Blockchain

The most important properties of the Blockchain are compiled in the following list [12]:

- **Public verifiability:** The Blockchain has the significant advantage over other data structures that every observer can at all times verify that it is in a valid state and that all the changes made to its blocks are according to the protocol.
- **Transparency & Privacy:** Everyone with access to the Blockchain can view all transactions over time. The more transparent the data is, the less private it becomes.
- **Redundancy:** The Blockchain reduces the chances of losing data with its distributed nature. Each peer locally stores parts of the whole Blockchain. The chance that data gets lost is extremely small.
- **Anonymity:** The Blockchain per se does not know users or real people. Instead, it uses a public-private key-pair that binds every transaction to a specific public key. Committing changes on the Blockchain, therefore, leaves no link at all between a real person and the corresponding public key.

7.2.2.3 Blockchain Application Scenarios

Due to the Blockchains various properties, multiple use-cases emerged. Even though the below list is neither exhaustive nor its items disjunct, it shows the most common scenarios for the use of the Blockchain [12] that are not further discussed in this paper:

- **Crypto Currencies:** From the moment when the paper by Satoshi Nakamoto [14] was published, Blockchains and cryptocurrencies were interlinked. Because a cryptocurrency is per definition a digital asset, the Blockchain suits this use case particularly well.

- **Supply Chain Management:** In Supply Chain Management, the flow of goods includes "various intermediate storage and production cycles" [12] and the Blockchain presents a suitable way to store and verify all processes in a tamperproof manner.
- **Payment and Money Transactions:** Transactions between different banks and result in temporary debts between them. This issue can be solved using *Distributed Ledger Technologies*, that make use of a Blockchain to settle debts.
- **Decentralized Autonomous Organizations (DAO):** "A DAO is an organization that is run autonomously through a set of smart contracts" [12]. A DAO is therefore fundamentally dependent on the Blockchain, as it relies on decentralized governance of funds enforced by smart contracts.
- **Proof of Ownership:** The idea of proofing intellectual or physical property is one of the most straightforward use cases for the Blockchain. In this scenario, the user registers his or her property (*e.g.*, image, text, or another object like land) through the use of an identity function in the Blockchain. "While this does not fully prove ownership, it does provide evidence of ownership if no one else can show that the object was previously published" [12].

7.2.2.4 The Blockchain and Cybersecurity

In most cases, the Blockchain technology is used within different types of applications to make them more secure. More secure databases, transaction systems for banks, the replacement of certificate authorities, or the security of critical physical infrastructures fall into this category. Another field of applications uses the Blockchain directly to mitigate cyber attacks. Examples, therefore, are the mitigation of DDoS Attacks or the handling of PKI related cyber threats. Applications that fall into the second category use the Blockchain more direct and immediate, while the first category uses them on a higher level, to help to reduce the cybersecurity threats in the set up of the application itself. Both of those categories are covered in subsequent sections.

7.2.3 Smart Contracts

This section looks at the concept of smart contracts on the example of Ethereum. Ethereum is a Blockchain "with built-in programming language", or in other words, a "consensus-based globally executed virtual machine". The Ethereum project started in 2015 and has gained popularity since then. It is second place in terms of market capitalization according to coinmarketcap.com, right after Bitcoin and Ripple [15].

Nodes in the Ethereum network make up the Ethereum Virtual Machine (EVM). Smart contracts are written in the programming language solidity. They get compiled to bytecode that the EVM can understand. Once deployed on the Blockchain, the code cannot be changed. Deploying smart contracts means mining them into the Blockchain, thus deploying smart contracts costs a fee like every other transaction. Once they are there, they are part of the Blockchain history. Every smart contract on the Blockchain lives at an address where the contracts exposed functions and variables can interact with. Smart contracts can inherit from smart contracts and can interact with other smart contracts. Users are able to transfer cryptocurrencies (*e.g.* Ether) to smart contracts. However, sending Ether to a smart contract that does not provide the functionality to retrieve those Ethers means they are lost forever in the contract. This is crucial, especially when designing contracts that implement business logic or any contracts for that matter. Smart contracts should be reviewed and audited carefully and tested for vulnerabilities else its weaknesses can be exploited like in the infamous "DAO" attack [16].

Every transaction on the Ethereum Blockchain costs a fee called gas. Gas is expressed in Ether. The miner that mines a transaction collects the gas associated with it; thus, gas serves as an incentive for miners to mine and contribute to the network. There is a defined set of operations alongside their gas costs in the Ethereum yellow paper [17]. Figure 7.1 depicts an excerpt of those costs.

APPENDIX G. FEE SCHEDULE		
The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.		
Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero} .
G_{base}	2	Amount of gas to pay for operations of the set W_{base} .
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low} .
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid} .
G_{high}	10	Amount of gas to pay for operations of the set W_{high} .
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.

Figure 7.1: Smart contract fees as shown in the Ethereum yellow paper

Miners generally prioritize transactions with a higher potential gain (gas). The total gas fee to be paid is calculated by the gas price times the amount of gas to use. If the gas price for a transaction is set high, the transaction is mined to the Blockchain faster. If on the other hand the price is set low, miners choose to mine other transactions first before mining transactions with a lower price. How high the gas price is is determined by the market. During peak network traffic, gas prices are high because as more people want their transaction to go through. Contrarily, prices drop when the network is less utilized. A transaction has a gas limit that can be set. This is to indicate the maximum amount of gas a transaction can use up before it is aborted. The gas limit times the gas price is the maximum amount the issuer is willing to pay for a transaction. In case of an endless loop, the execution only lasts until the gas limit is used up. If the gas limit is too little, the computation runs out of gas before it is finished, gets reverted, and the gas paid is lost. If the gas limit is higher than the amount of gas used, the remaining gas is refunded to the sender of the transaction.

Miners can only include so many transactions in a block that block gas limit is not exceeded. This is also not a fixed value as miners can vote with each block to increase or decrease the block gas limit by a certain amount. Block gas limits serve to keep propagation of blocks and transaction time low by fundamentally limiting the size of the blocks.

The following overview sums up the key terms as mentioned above [18]:

- **gas:** How much gas a transaction needs
- **gas prize:** How much Ether a unit of gas costs, determined by the market
- **gas limit:** Maximum amount of gas to use up for a transaction
- **transaction fee:** Gas used times gas price
- **block gas limit:** Maximum sum of gas of all transactions in a block

Not all smart contract interactions cost gas. Functions that do not modify the state of a contract, and hence the Blockchain, do not need to be mined by a miner. Calling those functions still uses gas because every operation in the EVM uses gas, but that gas is refunded immediately as calling those functions does not result in a transaction. In Solidity, those functions have the modifier “pure” or “view”. Functions denoted with

“pure” do not even read state variables. Both those functions can be run on a single node in the EVM and nothing has to be propagated to the network. Thus, no transaction has to be mined into the Blockchain, and no gas fee has to be paid [18].

```
1  pragma solidity 0.5.0;
2
3  contract MyContract {
4
5      string private myString = "foo";
6
7      function getString() view returns (string) {
8          return myString;
9      }
10
11     function setString (string _string) {
12         myString = _string;
13     }
14 }
```

Listing 7.1: Smart contract example

In the smart contract example, presented in Listing 7.1, the functions “setString” modifies the state, namely the variable “myString”, thus a transaction needs to be recorded in the Blockchain and gas has to be paid. The function “getString” on the other hand is denoted with the keyword “view”, no state variables are modified so no transactions are required and no gas fee has to be paid.

Smart contracts allow programming on the Blockchain which opens up a multitude of possibilities also in the context of cybersecurity for which upcoming sections discuss some solutions.

7.3 Blockchain Applications in Cybersecurity

Having built the theoretical foundation in Section 7.2, we now focus on the main research question of this work and summarize recent research in the intersection of Blockchain and cybersecurity. The subsequent sections can broadly be separated into a low-level, more technical part (Sections 7.3.1 and 7.3.2), and a high-level, more application-oriented part (Sections 7.3.3 and 7.3.4).

Section 7.3.1 first presents some of the possible Blockchain-based solutions that have been developed as a countermeasure to DDoS attacks, alongside a comparison to currently existing countermeasures and the potential motivations for attackers in a DDoS context. Subsequently, Section 7.3.2 goes into the potential application of Blockchain to rebuild a vital part of the internet security infrastructure, precisely the current idea of PKI's. On a higher level, Section 7.3.3 then presents some of the critical issues in securing IoT devices, as well as how they could be approached with Blockchain technologies. Finally, Section 7.3.4 summarizes the currently productive applications of the Blockchain in a broad cybersecurity context, as well as some other application areas that are still part of current research.

7.3.1 Distributed Denial of Service (DDoS)

A DDoS attack has become a popular way to cripple servers of an institution or a private person. The current internet design has the purpose of moving packets from a source to a known destination. The network itself forwards all packages at minimal cost and generally outsources the complexity, including security, transport reliability and quality of service to the sender and receiver of the transported packages. This concept has been called the **end-to-end paradigm**. As no intermediary entity intervenes, a party (either sender or receiver) can damage its opposition by using attack possibilities such as IP Spoofing or the aforementioned DDoS attacks [19]. By faking the source address in the header of a packet, the sender can hide its identity. This security issue is called IP Spoofing [20]. As described in [19], the following security issues raise opportunities of attacks:

- **Limited internet resources:** Every host or service has hardware limitations that users may consume.
- **Highly interdependent internet security:** No matter how secure a host may be configured, DDoS attacks always depend on the security of others within the Internet.
- **No collocation of intelligence and resources:** The intermediate network has plenty of resources, as they forward packages at minimal costs. In contrast, the end networks only invest in as much bandwidth as they planned to use for their services.
- **No enforcement of accountability:** As described above, attackers can escape from accountability by using IP Spoofing mechanisms.
- **Distribution of control:** In a world of a distributed network, such as the Internet, multiple clients participate in the network. Each one of them has different security mechanisms. No global control entity can define a security policy or standard.

Since the number of connected devices has increased due to IoT devices, such as connected cameras, or smart fridges, attackers have a growing capacity to take control of unsecured devices [21]. By operating multiple such attacking entities called computer bots (clients that have been taken over by malware) remotely, attackers try to overflow a website, a network or a server and prevent rightful users from accessing the application. The service

is either responding slowly or shut down entirely. More precisely, attackers send a stream of packets to the victims, which consume all capacity of hardware resources and therefore make it unavailable for legitimate clients to access the service provider.

Another popular way for an attacker is to send malformed packets to impact the availability of the application service negatively. Those packets confuse the web application or some protocols on the victims' hardware and force the server to reboot. There are probably additional possibilities to attack services on the internet. Such attack possibilities are mostly discovered first once they have been exploited in a significant attack and servers have been down for a particular time [19].

The procedure of a DDoS attack is split into the following phases: An attacker recruits multiple agents (clients) into which the attacker injects malicious code. Attackers often hide the identity of infected clients by using IP Spoofing mechanisms [19].

Multiple incentives exist to attack clients using DDoS attacks. Unfortunately, the primary goal of such an attack is to damage the selected victim. Motives may be found in prestige (gaining respect within the hacker community when attacking popular websites or services), personal hatred, material gain (damaging competitors by attacking them), any political reasons, or blackmailing others [22, 19].

Many companies currently offer DDoS protection services, such as Cloudflare or Akamai, and their number is increasing [23]. Those solutions serve as a proxy and manage routing, load balance, and drop traffic when a DDoS attack occurs. For all solutions, a third party DDoS Protection Service (DPS) provider is required, resulting in extra cost, as the analyses are performed in the cloud [21]. Those cloud-based defense services could become a communication bottleneck because the traffic (download and processing) is dependent on a single provider. By utilizing resources from other companies, the workload of mitigating DDoS attacks can be shared [21].

The Internet Engineering Task Force (IETF) additionally is proposing a protocol called DDoS Open Threat Signaling (DOTS) that requires both clients and servers. A new protocol is required, which has to be maintained. DOTS clients have to register to a DOTS server and use this protocol among the agents to organize the DDoS protection [21]. In the following sections, various approaches to mitigating DDoS attacks without the need to deploy a new protocol are illustrated.

7.3.1.1 DDoS Mitigation with Smart Contracts

This concept investigates a possibility to mitigate a DDoS attack in a fully decentralized manner using smart contracts and their underlying Blockchain. These technologies allow sharing information (detection and mitigation mechanisms as well as IP addresses) about attacks in an automated and decentralized system [21].

As described in Section 7.2.3, a smart contract is a software that is made to help contracts being able to execute and verify on their own. To do so, there has to be an infrastructure that implements, verifies, and enforces the negotiation of those smart contracts by using particular computer protocols and that runs fully decentralized. As known from Section 7.2.2, a Blockchain ensures permanent storage and provides obstacles to manipulation of content and is thus an ideal infrastructure for smart contracts. Nodes participating in the Blockchain run a smart contract by executing and validating a script and after that storing the contract and the results in a new block [21].

As presented in Section 7.2, the architecture is composed of three components. The customers report IP addresses to the Blockchain via smart contracts. The Autonomous Systems (AS) retrieve lists containing these addresses and implement DDoS mitigation techniques. All participants interact with the underlying Blockchain [21].

The web server of one AS is a victim of a DDoS attack. Participants that have proven ownership of their IP then create a smart contract that stores all IP addresses of attack-

ers. Subscribed systems receive updated lists of IP addresses every 14 seconds, as the underlying Blockchain, Ethereum, creates new blocks within that timeframe. As soon as all other autonomous systems receive the list of attackers, various mitigation strategies may be triggered tailored to the specific domain [21].

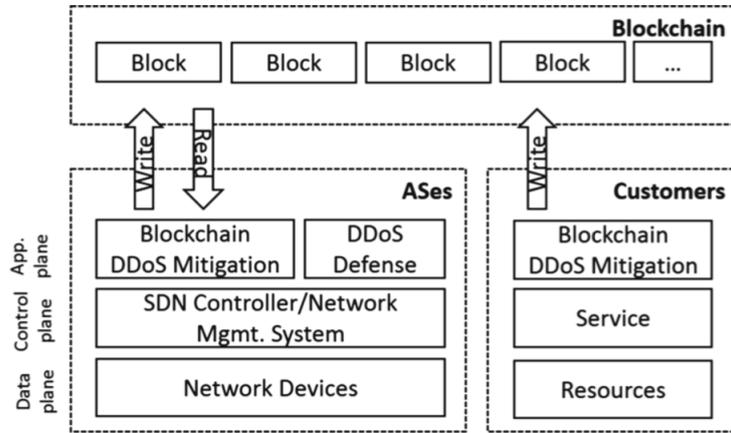


Figure 7.2: System Architecture

This approach uses an already existing and publicly available distributed infrastructure to adumbrate IP addresses that are either white- or blacklisted. This approach serves well as an additional security mechanism to existing DDoS defense systems across multiple domains by using an Ethereum Blockchain without transferring the control of their internal network to a third party. For large-scale attacks, this approach currently is not supported well, but this issue is to be addressed in future work [21].

7.3.1.2 Mitigation-as-a-service in Cooperative Network Defenses

With cooperation between multiple domains, various collaborating AS can alleviate DDoS attacks by redirecting excessive traffic to other domains that filter the traffic. Incentives ensure the use of mitigation-as-a-service for cooperative network defenses. By paying fees, enterprises or individuals can subscribe to such a cooperative defense network and protect themselves from future attacks [24].

As soon as a target detects an attack, it sends a request to all participating autonomous systems to mitigate the current attack. Subsequently, a mitigator that is responsible for the range being attacked then either accepts or declines the mitigation request. By using a **proof-of-mitigation**, the completion of the mitigation has to be confirmed, and the target can pay the mitigator. As this proof-of-mitigation has to satisfy time constraints, tamper-evidence, and reproducibility, this proof has to be executed automatically during the current time window. Additionally, any user interaction has to be excluded to ensure efficiency [24]. Various approaches creating such proof are described and discussed in this section and have been tested on different metrics related to security and practicability.

Marketplace of Mitigation VNFs Virtualized Network Functions (VNF) can be deployed on any hardware without additional configurations. The concept of Network Function Virtualization (NFV) can, therefore, provide an efficient solution by virtualizing a single function in the network, as seen in Section 7.3. All autonomous systems involved in the cooperative network could load the VNF image directly from the marketplace, which then provides the mitigation service hosted on virtual devices. By comparing the hashed checksum of the VNF image to a known value from the marketplace, the integrity of the VNF image is checked. Additionally, local caching of all VNFs avoids large load on

the marketplace. A big advantage of this approach is the high degree of isolation, as the VNFs consist of the minimal code necessary to handle the mitigation. Unfortunately, the target still needs to trust that the mitigator only runs untampered VNFs directly from the marketplace [24].

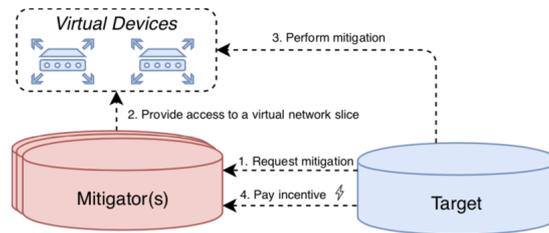


Figure 7.3: Marketplace of Mitigation VNFs

Trusted Computing In contrast to the previous approach, as illustrated in 7.3, the mitigator directly initiates the mitigation. So-called Trusted Platform Modules (TPM) enable secure storage of hashes. This allows that only the code that is approved by the cooperative defense is run on the system. VNFs create a defense network where trusted and known VNFs always handle requests for mitigation without trusting the responsible operator that provides the mitigation service to the autonomous system. A significant disadvantage of trusted computing is the strict hardware requirement, as the TPM is only available as a standalone chip or integrated into the motherboard [24].

Secure logging As mitigators can use full network infrastructure, they are allowed to proof the mitigation using the available traffic data. By having a detailed log of network activities, they could prove the successful mitigation by identifying a reduction in the traffic from the attack source to the DDoS target. A mitigation-proof based on the log files decreases the complexity of the proof. The aforementioned log file has been created on an isolated system (which requires no additional trust) and therefore has to be checked about its integrity by a third party. A disadvantage comes with high-volume attacks which create large log files. These large files introduce additional delays as the log files have to be transferred for remote auditing [24].

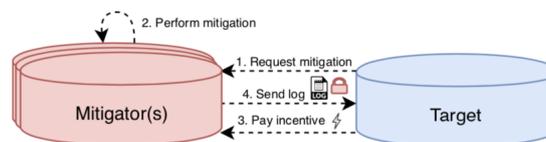


Figure 7.4: Secure logging

Network Slicing Since network virtualization technologies and Software-Defined Networks (SDN) have advanced in recent times, they both serve as a basis for network slicing as a service. When the attack target requests mitigation services, it gains access to the virtualized network slice of the mitigators autonomous system. The slice is then configured to provide access for all attacking IP addresses that have been requested within the mitigation request [24].

As a final consideration, the authors remarked that none of the preceding methods could on its own be used to address a trade-off between practicability and security when qualitatively comparing all four approaches. However, combining some of these methods could lead to a practical as well as secure solution [24].

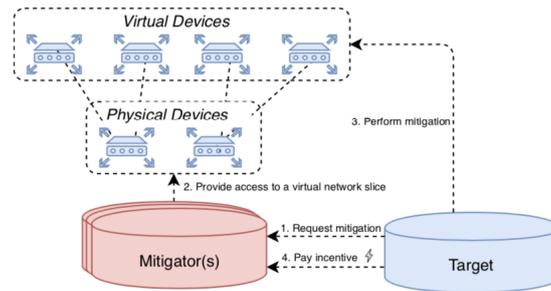


Figure 7.5: Network Slicing

7.3.1.3 Multi domain DDoS Mitigation based on Blockchains

Another approach repeatedly uses smart contracts as a means of advertising information across multiple domains. The architecture as seen in figure 7.6 involves the following entities: **SDN** facilitate the development of customizable security management; **NFV** that are provisioned in generic hardware strengthen security policies through virtualized functions; an Ethereum-based **Blockchain** is a base for all participants of the cooperative network to advertise DDoS attacks within a timeframe of 14s, in which a new block is mined; and a **smart contract** that stores black or whitelisted IP addresses of customers [21].

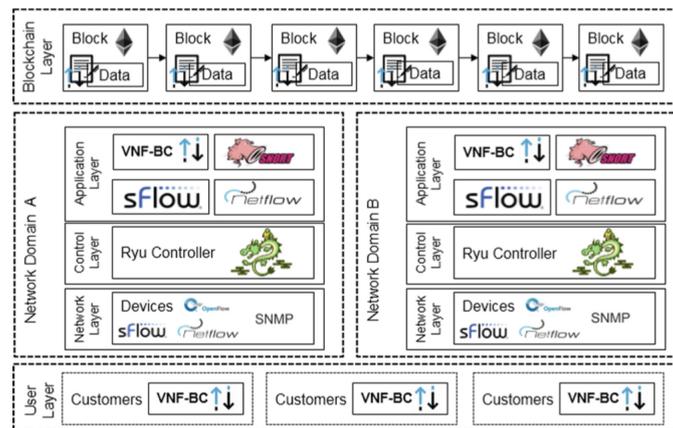


Figure 7.6: Multi-domain architecture

This architecture is based on key technologies such as SDN and NFV. Blockchain and smart contracts advertise DDoS attack information to broadcast white or blacklisted addresses without building any distribution mechanisms or protocols [21].

7.3.1.4 Blockchain Signaling System (BloSS)

Similarly to [21], this approach deals with cooperative, multi-domain DDoS defense systems, but it presents a Blockchain Signaling System (BloSS) which is a somewhat technical approach of deploying hardware simplifying signals of DDoS attacks. Principal components for a BloSS are smart contracts that describe how information has to be transferred between AS and decentralized applications that include parameters defining the interaction of AS. Additionally, a consortium based Blockchain, differentiating from the public and private Blockchains, serves as an intermediary level of confidence.

An AS creates a wallet and notifies a smart contract that stores some information. The smart contract then implements a method to reclaim addresses published by other systems by querying a central smart contract. This central contract is used to configure all involved smart contracts with updated information on their managed IP networks, as the addresses

of the involved wallets are immutable. To prevent free-riding peers (parties that only consume without any contribution), an incentive mechanism needs to be set by defining tokens [25].

7.3.2 Public Key Infrastructure (PKI)

Public Key Cryptography serves as one of the cornerstones of modern communication systems and technologies, also including the Blockchain. Entities can secure their communication asynchronously using a pair of keys (public and private). After the distribution of its public key, an entity can be contacted by anyone via securely encrypted channels. What makes this type of cryptography so convenient is that its security does not depend on any shared hidden secret (*e.g.*, a passphrase). An entity encrypts its message using the public key of its counterparty, after which only the supposed recipient can decrypt the message with its private key. While the public key and private key always belong to the same communication channel, it is practically not possible to infer the private key from the public key. This means that the public key can be distributed without considerations about the trustworthiness of potential recipients (*e.g.*, publicly on online platforms) [26]. For a private exchange between two to a few counterparties, the simple approach of distributing public keys out-of-band (*e.g.*, in person) might suffice to establish sufficient confidence and trust. As soon as we consider communication networks on a larger scale, an in-person exchange between all parties in the network would no longer be feasible. In such a case, one or more trusted, often centralized entities need to support the process of establishing trust [26].

PKI is the summary term for an infrastructure that enables secure communication through such trusted entities. As described in [26], a PKI "lets participants communicate with each other securely using public key cryptography and allows them to verify each other's keys with the help of certificates and trust relations."

7.3.2.1 Existing PKI Systems

Traditional PKI systems are most visibly used for communication over the web. Before two entities (*e.g.*, a server and a client) can securely communicate over HTTPS/TLS, the domain name of the server must have been validated and certified by a certificate authority (CA). A CA is an entity that has the authority over issuing certificates for a given PKI namespace (*e.g.*, domain names or a subset thereof).

A PKI certificate verifies that the given name-value (*e.g.*, domain name and public key) binding is authentic and is itself signed with the key of a CA. The trustworthiness of a CA is signed by a higher-level CA, building a chain up to a root-level CA that ultimately needs to be trusted without verification. When establishing communication with a server over HTTPS, a client verifies the identity of the server according to its certificate, and client and server confidentially decide on a data encryption secret by employing asymmetric encryption. All further message and data exchanges are secured with symmetric encryption [26]. The procedures of PKI management as employed in HTTPS/TLS, as well as many other systems and private infrastructures, are standardized by the IETF in the X.509 standard [27].

Contrary to the centralized approach, the Pretty Good Privacy (PGP) architecture is based on a decentralized web-of-trust where all participants manually define their trust anchors (*i.e.*, which entities to trust as the anchor for a certificate chain) instead of relying on a predefined trust store as included in operating systems or browsers [26]. While PGP is in relatively widespread use for secured email communication, this work focuses mainly on centralized infrastructures for their approachability with Blockchain architectures.

7.3.2.2 Threat Model for Public Key Infrastructures

PKI systems allow for an establishment of trust and secure exchange between parties that do not necessarily know each other. However, both participating parties need to trust the centralized CA to issue valid certificates. The most apparent threat to a PKI system is, therefore, an attack on this centralized trusted party.

Any such kind of attack could, if successful, allow an intruder to generate fraudulent certificates, which could then be used to impersonate one or more of the parties and perform a proxied man-in-the-middle attack. In such an attack, the attacker acts as a proxy that intercepts messages between the parties, decrypts and extracts their contents, and redirects them to the original recipient, making it very difficult for the participants to figure out they are being intercepted [28].

In the concrete example of a web certification authority, domains are most often validated by merely sending an email to an address on the domain to be certified (*e.g.*, `webmaster@xyz`) or by verification of DNS records. This indirect verification creates vulnerabilities on several levels, as emails and DNS settings are managed by separate entities that could be attacked. CAs have also been known to issue fraudulent certificates on government request (*e.g.*, for censorship), or through simple human error [28, 29].

As centralization in PKI systems leads to a large attack surface, developing approaches that can function without the need for any such central entities (*e.g.*, CAs) have been a topic of several publications in recent years. The following sections provide a more detailed overview of decentralized PKI in general as well as more specific applications in the area of web certification. While there are many other types of attacks on PKI (*e.g.*, obtaining a private key through social engineering or directly attacking the encryption protocols), we focus on the critical aspect of centralization in this work.

7.3.2.3 Decentralized Public Key Infrastructures

Namecoin A foundational work that introduced the idea of applying Blockchain to the decentralization of naming systems was initially created to rebuild the Domain Name System (DNS) such that it can work without centralized DNS servers. Much like central PKI entities, DNS servers can be exploited and, if hacked, can be a threat to large parts of the internet user base.

The main idea of Namecoin is to store all DNS-related naming information in a Blockchain ledger that enforces state updates to occur via appended transactions. All entities in the network share a copy of this ledger and can independently lookup the records associated with a name. Records are owned, and can only be modified, by a given cryptographic identity [30, 31].

While this approach to naming provides the benefit of decentralization, it comes with challenges regarding security, performance, and scalability. The Namecoin Blockchain is a fork of the Bitcoin Blockchain and shares most of its technicalities and limitations but progresses a separate ledger. This means that it has a separate and much smaller set of miners, making it very vulnerable to 51% attacks (where an attacker controls a majority part of the computing power and can rewrite the chain) or DDoS. Even though Namecoin implements merged-mining (*i.e.*, it allows bitcoin miners to participate in Namecoin mining), its computation is mainly concentrated on one big pool of miners with >50% of the computational power [30, 31].

In addition to the limitations on a security level, Namecoin also struggles with regards to performance (*e.g.*, transactions per second). As the block size in bitcoin and its forks (*i.e.*, Namecoin) is relatively small, there is a limited amount of data that can be stored in transactions and processed in a block, which severely limits the throughput of transactions and can lead to issues in a DNS use case.

Blockstack The Blockstack system was initially based on the Namecoin Blockchain using a dedicated namespace. Using the initial system called Blockstack ID, users could link their username with their public key, as well as some additional data. To work around the issues of limited storage within a block, Blockstack ID would connect related key-value pairs using a linked list data structure mapped to the Blockchain [31].

Mainly due to the security implications of Namecoin also applying to Blockstack ID, its maintainers have intermittently developed a new approach that replaces Namecoin with the Bitcoin Blockchain. Contrary to its predecessor, Blockstack does not store its data directly in the Blockchain. Instead, the Blockchain only stores hashes of the data, while the data itself is stored in external systems [31].

Blockstack introduces a novel approach that is called the virtualchain. The virtualchain is a logical structure built on top of a Blockchain that processes transactions relevant to the Blockstack system and implements a state-machine over these transactions. When state updates occur, new transactions are encoded and written to the underlying Blockchain (subject to consensus). The virtualchain is used to compute a global state of who owns what names and what is associated with these names. These components form the control plane of the Blockstack architecture [31].

Instead of storing the production data in Blockchain productions, Blockstack only appends hashes and pointers to the real data to the Blockchain. The production data is managed by a separate data plane that consists of routing as well as actual persistence capabilities. Similar to the approach taken by traditional DNS, the routing layer is based on zone files that contain all routes/records for any given name. The actual persistence layer then stores whatever data is to be associated with a record in a database or another storage system [31].

Separating the architecture into a control and data plane allows for efficient lookups of data against the data plane, while still allowing for verification of the data stored therein using the control plane. As a hash of the zone file is stored alongside its name in the Blockchain, a low-level transaction with consensus is necessary to modify the zone file that is associated with a name. Furthermore, if hashes of record contents are put inside the zone file, these contents also become immutable without consensus [31].

The Blockstack system architecture also offers possible solutions for several fundamental problems regarding the storage of associated data. The production data is stored outside of the Blockchain, which works around the limited amount of storage per block. Additionally, the data that needs to be replicated across participants in the network is limited, as transactions only contain hashed and encoded information [31].

Bitforest Approaches like Blockstack are possible solutions for the deployment of a fully decentralized PKI system that cannot be controlled or censored by any one entity. However, while this might be advantageous for some cases, organizations that deploy a PKI, even if it is decentralized, more often want to keep some control over their namespace. Without any such means of control, PKI systems are vulnerable to many kinds of abuse. An exemplary kind of abuse is called name squatting and refers to the preemptive reservation of names under the expectation of future profit (*e.g.*, through auctioning off the name) [32].

Similarly to the Blockstack architecture, the approach taken by Bitforest is based on a data structure on top of a Blockchain. However, Bitforest allows an organization to keep control over who participates in their namespaces. A namespace administrator can define mappings of names to indices in a binary search tree, where for each index, a list of all current and preceding values is stored [32].

However, even though an administrator defines the mapping, only the owner of the name can update the associated value by appending to the log of the corresponding name. This

maintains the critical property of identity retention, meaning that only the owner and no centralized entity can ever change a value that has been associated with a name [32].

When clients need to look up the current value that corresponds to a name, they search for the corresponding index in the directory maintained by the namespace administrator. Querying the search tree for the given index yields the current value, while verifying the previous log entries ensures the integrity of the name [32].

7.3.2.4 Web Certifications using Blockchain Technologies

One of the most critical application areas of PKI systems is the certification of the integrity of domains in the world wide web. Through its exposure to a substantial number of people, the web certification system is one of the most widely exposed and has been subject to several critical vulnerabilities and attacks. Recent research tries to prevent some of these attacks by employing a Blockchain-based web certification infrastructure that does not need to rely on trusted central entities.

Ghazal In the traditional web certification model, domains and their owners are validated by trusted certification authorities. The most popular means of validation is domain-validation, meaning that only the association between the public key and domain name is certified. This approach suffers from the many critical issues due to indirections, as described in Section 7.3.2.2.

Ghazal proposes a new uni-authoritative paradigm for web certification and domain infrastructure. Concretely, no single authority should be needed to certify entities or register domains, as the Blockchain itself could serve as a sort of authoritative entity. This approach has the potential to solve many of the vulnerabilities occurring due to indirections and centralization in general [33].

The foundation of the Ghazal prototype is built on a smart contract on the Ethereum Blockchain. The contract specifies all procedures regarding the registration, transfer, and expiration of domains, as well as the associated values and certificates. Upon registration of a domain name in the *.ghazal* namespace, the owner can assign arbitrary values to the name (including public keys/certificates). Registrations and other operations incur a fee that is paid to a block-hole account on the Ethereum network, meaning that nobody can spend the fee again [33].

While a uni-authoritative paradigm could solve some of the issues in today's web certification, browsers and operating systems would need to be extended to support such functionality. For example, to support the current Ghazal proposal, *.ghazal* domains would need to be recognized, processed, and verified differently from the remainder of sites on the web [33].

It is also still a topic of research if and how a decentralized system could support domain lookups efficiently without the need for storing the entire Blockchain on each client device. Furthermore, the approach taken by Ghazal is plagued by new issues that arise through the removal of all central entities. For example, if a domain owner's private key were to be lost, the corresponding name could never be reassigned [33].

Certificate Revocation and Transparency Certificate Transparency (CT) is a recent improvement to the traditional web certification model. When CT is enforced by a browser (*e.g.*, modern versions of Chrome), the browser only accepts certificates without warning if they have been published to a public append-only log of certificate issuances. This measure has been introduced to counteract the malicious issuance of certificates by hacked CAs or through compulsion or malfeasance [29].

While CT in traditional web certification can already improve web security significantly, it is only a reactive measure, meaning that fraud can still occur but are probably recognized

more quickly. [29] proposes that certificate issuance as well as handling the revocation of certificates could be moved to a Blockchain-based system, preventing the issuing of malicious certificates entirely without the approval of the domain owner.

In the solution proposed in [29], certificate authorities would not hold the solemn authority over certification. Instead, a certificate would only ever be valid if the associated web server cooperates in their publication by publishing the CA-issued certificate to a public certificate issuance Blockchain. Browsers would then only accept certificates that have been both issued by a valid CA and published to the Blockchain. By additionally enforcing a short expiration time on certificates, certificates that need to be revoked could just not be renewed, causing the CA to issue a revocation. Similarly to the approach taken in Ghazal, a key challenge of such an approach is that clients need to maintain a copy of the certificate Blockchain to verify the integrity of certificates.

7.3.2.5 Keyless Signature Infrastructures and Blockchain

A Keyless Signature Infrastructure (KSI) is the basis for an alternative approach to verification of the integrity of documents that does not depend on the secrecy of private keys. A document that is submitted to a KSI is first hashed and then added to a Merkle hash-tree containing all documents processed in the same round (*i.e.*, with other documents that arrived around the same time). The signature token received upon document submission can then be used to check whether a document was contained in the tree at the given time. By following the path encoded in the signature token, it must be possible to reconstruct the root-hash of the entire tree [34, 35].

The KSI architecture processes incoming signature requests in rounds (*e.g.*, one round per second) and, after completion of a round, appends the root-hash of the generated hash tree to a global tree (“hash calendar”). The root of this global tree is then periodically published as a trust anchor that can be used to verify the presence of a document at a point before its publication [34]. Contrary to the approach of signing documents with one’s private key, this approach depends on the infrastructure (*i.e.*, signing servers) to sign the document into the tree. This is what makes possible keyless signing but also incurs a dependency on the availability of the KSI to sign a document [35].

The authors of [34] suggest that the root-hash could be published to broad-reach media like newspapers in monthly iterations. This would allow verification of a document’s integrity even without the KSI infrastructure, as one can compute the root-hash and compare it to the one published. However, this approach is inefficient in that it needs manual verification against an external medium and can only be performed with a reasonable publishing frequency [35].

The Blockchain-based approach presented in [35] improves the efficiency by publishing to the Blockchain in an increased frequency such that one can more granularly verify when a document was signed (*i.e.*, added to the tree). Additionally, the break between media is no longer necessary, as signatures can be automatically verified based on the Blockchain contents. When publishing to the Blockchain, the frequency can be increased to several times per hour, as it is only limited by the throughput of blocks being mined [35].

7.3.3 Internet of Things (IoT)

Figure 7.7 shows the predicted market impact of IoT in billion U.S. dollars in each category. Not only is IoT relevant for Industry 4.0 with smart factories or the move toward smart cities. IoT also impacts everyday life with smart homes, vehicles, and healthcare. This section aims to present collected solutions on how Blockchain technology can help with security with regards to IoT and its applications.

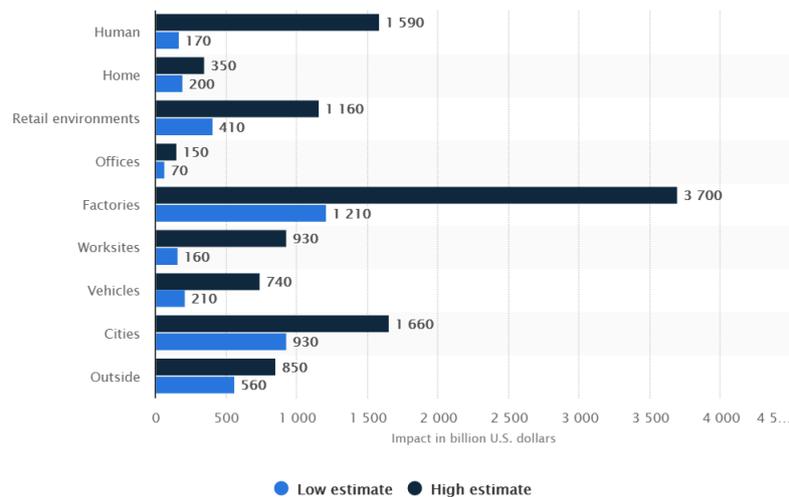


Figure 7.7: Iot market impact, Source: <https://www.statista.com/statistics/580778/worldwide-internet-of-things-economic-impact-forecast/> [36]

Banerjee et al. in [37] state that given the potentially sensitive nature of IoT datasets, there is a need to develop a standard for sharing IoT datasets among the research and practitioner communities and other relevant stakeholders. They propose two examples to use Blockchain for IoT security. One for IoT dataset sharing using Blockchain and another one for Blockchain-based firmware detection and self-healing of devices.

First, to ensure the integrity of the IoT datasets, they propose a Reference Integrity Metric (RIM) maintained by a Blockchain. Whenever a dataset is downloaded, its integrity can be checked using the RIM. A central hub maintains the references to the repositories of the datasets. Information of the members is also stored in a Blockchain. The lifetime of the shared datasets is in the hand of the repository owner. As only the RIM is stored in the Blockchain and not the datasets themselves, members can still choose to stop sharing their datasets by removing the repositories. To preserve privacy, they emphasize the use of an automated tool that anonymizes the datasets before they are published.

Second, to ensure firmware integrity, they also propose a system with RIM checking and Blockchain. Firmware is considered the root of trust. Starting with the bootloader, it checks whether the next software (*e.g.*, the operating system) can be loaded. However, firmware must allow for updates, which opens up a window to compromise the RIM of the firmware. By using Blockchain, the firmware history can be tracked, and compromised devices can be detected and forced to rollback their version to the last valid entry.

Khan et al. in [38] review and classify security issues for IoT. They discuss various Blockchain solutions for IoT:

- **Address Space:** With Blockchain, more addresses can be used than with IPv6. Blockchain has a 160-bit address space, while IPv6 offers a 120-bit address space. Also, many IoT devices are constrained in memory and computation capacity, and therefore are unfit to run an IPv6 stack.
- **Identity of Things (IDoT) and Governance:** IoT devices often change ownership along the supply chain or when resold. Also, IoT devices have many relationships. Those relationships can be *device-to-human*, *device-to-device* or *device-to-service*. Further relationships could be *deployed by*, *shipped by*, *used by*. With Blockchain, such challenges can be solved efficiently and securely.
- **Data authentication and Integrity:** Data transmission of IoT devices connected to a Blockchain is always cryptographically proofed and signed by the true sender.

- **Authentication, Authorization, and Privacy:** Smart contracts offer the ability for more effective authentication and authorization rules than traditional methods like OAuth 2.0. Programming access rules can enforce data privacy into smart contracts. These rules then determine who has the right to update, patch, or reset the IoT devices. Also, service and repair requests and change of ownership can be initiated that way.
- **Secure Communications:** IoT communication protocols like HTTP and MQTT are not secure by design. They have to be wrapped by security protocols to make them secure, like with TLS for HTTPS. With Blockchain, the complexity of PKI and its management is simplified. Every IoT device connected to the Blockchain has its unique GUID and asymmetric key pair.

Dorri et al. in [39] present a use case for Blockchain in a smart home. In their use case, each smart home is equipped with an always online, high resource device, called the “miner”. It is responsible for handling all communication within and external to the home. The miner mines a private and secure Blockchain used for controlling and auditing communications. To address the challenges regarding power consumption in Proof-of-Work (POW), they propose a framework based on trust to limit energy consumption and make their solution more suitable in the context of IoT.

The design consists of three core components: the smart homes, a cloud storage, and an overlay. Smart homes build and overlay with the Service Provider (SP). Smart homes are clustered and, in each cluster, one home is selected as Cluster Head (CH). They maintain a public Blockchain and two key lists. One list contains the private keys of overlay users that are allowed to access data for the smart homes connected to this cluster. The other list holds the public keys of smart homes that are allowed to be accessed. The cloud storage stores and shares data of the smart home devices.

The local Blockchain in the smart homes keeps track of transactions and has a policy header to enforce users’ policy for incoming and outgoing transactions. The policy header consists of four parameters. The first is the requesters public key; the second is the requested action (*e.g.*, to store or access data in the cloud storage); the third is the ID of the device inside the smart home; and the last parameter is to indicate the action that should be done for the transaction that matches with the previous properties (*e.g.*, deny or allow the transaction). The miner in each smart homes authenticates, authorizes, and audits transactions.

7.3.4 Specific Blockchain Use Cases

In this section we take a closer look at applications that are linked to cybersecurity differently: Here the Blockchain is not per se used to mitigate direct cybersecurity threats but is used to make specific applications more robust, error-prone and less vulnerable.

7.3.4.1 Assessment Criteria

Below we want to introduce different scenarios of applications for the Blockchain technology and answer the following questions:

- How does the application make use of Blockchain Technology?
- Is the Blockchain needed or could the problems be solved without it?

To answer these questions, we are going to make use of the schema presented by [12], that helps to decide when to use a Blockchain and when not.

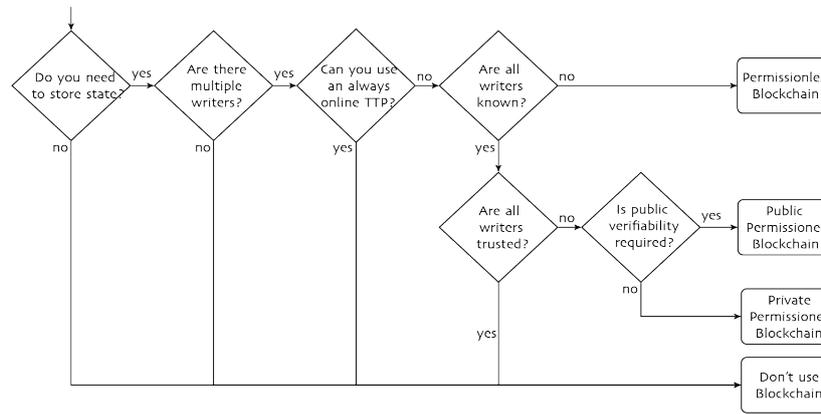


Figure 7.8: "Do you need a Blockchain?" Schema [12]

7.3.4.2 E-Voting

The ability to vote is the very foundation of every successful democracy and must be accessible for all eligible citizens. Most common systems today are paper-based voting systems that do not scale well and rely on the procedural security of officials conducting their jobs. Moreover, while various e-Voting systems exist, all come with multiple security vulnerabilities, that pose an enormous risk of election rigging and fraud. The Blockchain, on the other hand, offers a transparent and incorruptible database that does not have a single point of failure and cannot be controlled by a single entity. Meanwhile, new concerns arise, such as a lack of privacy and authentication issues.

To date, several e-Voting systems exist, differing mostly in the amount of integration and the role of the Blockchain. A more exhaustive overview of such applications can be found in [40].

Most existing systems such as Votebook [41] or Vote Watchers [42] do not fully integrate the Blockchain, but instead use it as an underlying database system. In both cases, people have to authenticate by showing up at a real voting booth using paper ballots. Both use the Blockchain as a specific type of database that is tamperproof and can create a history of all its entries. Additionally, the results of each booth can be aggregated with all other stations to generate a final result[40]. According to [43], these systems follow the only "doable" approach, as they do not deny the fact, that voters have to identify themselves somewhere.

FollowMyVote [44] follows a fundamentally different approach: It not only integrates the Blockchain but uses its ability to act as a fully distributed ledger and self-organized entity. Each voter installs a local "Voting Box" on his computer, requests ballots online and votes directly from within the application. The authentication process is done via the organization that holds the election by uploading official documents also from within the application. Nevertheless, according to [43], this idea is fundamentally flawed as authentication mechanisms are still an issue, and the technique could scare off remote voters.

Within the three proposed systems, two fundamental questions seem to remain unanswered:

1. **Authenticating users and the existence of a trusted third party:** To authenticate the voters, an organization that has all the necessary data is inevitable. In most cases, this leaves only the government as an option. However, according to the diagram by [12] the use of Blockchain technology only makes sense if no trusted third party exists. This causes a dilemma: If the non-existing trust in the government is the justification for the use of the Blockchain, the question arises whether the government can be trusted regarding the authentication of the voters. Even

though a systematic rigging is harder to achieve with this approach, the problem remains at hand.

2. **The user as a single point of failure:** While the Blockchain does seem to be perfect for voting situations, it still leaves the most important point of failure unprotected: The user himself. Even the most tamper-proofed, Blockchain-based voting system does not prevent hackers from a compromising an end-device.

Up to date, no proposed system has been shown to be secure, verifiable, and private at the same time. The question of authentication comes into play as an additional point of failure of the concept [43]. Hence, if a trusted third party (online) exists, the use of the Blockchain is not necessary. If the government is trusted as far as the authentication of the voters goes, a public permissioned Blockchain can be a good fit. However, the security of the system then relies on the integrity of the validators, leads us to where we started. If the government is not trusted at all, there exists no solution that overcomes that systematic flaw in a countries governance.

The Blockchain can be a solution if the question of authentication can be answered satisfactorily. Otherwise, a traditional paper-based voting system is as good as any voting system, including Blockchain based systems or systems that are based on a trusted third party.

7.3.4.3 Autonomous Vehicles, Smart Cities & IoT

Smart vehicles are increasingly connected to surrounding infrastructure via the Internet, thus making them part of the IoT. This development brings apparent advantages, but also many challenges, especially in the area of cybersecurity. Malicious attacks on a vehicle endanger not only the vehicles data and passengers but also other road users. Traditional approaches meanwhile tend to be ineffective due to centralization, unscalability, and unsecured communication architectures [45]. Because the Blockchain offers solutions to many of these problems, various approaches exist to connect self-driving vehicles with a Blockchain-based architecture.

Ali et al. propose a smart vehicle ecosystem based on a Blockchain architecture [45]. It consists of nodes that are clustered and connected via Overlay Block Managers (OBMs) that form the Blockchain overlay. Each vehicle presents a node that is connected to to the system through the closest OBMs. All Transactions are broadcast to and verified by the OBMs. To ensure the user's privacy, each vehicle is equipped with internal vehicle storage to store sensitive data. Each owner can decide which data is provided to third parties and which data is not. To make specific public keys identifiable in the real world, a trusted third party is involved, and a centralized mechanism is used. This is needed in the case of service centers and software providers. The ecosystem makes remote software updates, vehicle insurance, smart charging, and vehicle sharing possible.

Also on a large scale, but slightly different is the hierarchical system proposed by [46]. Two trusted third parties that register the vehicles (Department of Motor Vehicles) and that categorize the vehicles into minor nodes and ordinary nodes (Revocation Authority) exist at the top layer. At the same layer, controller nodes constitute the Blockchain overlay and hold the information of the vehicle network. The lower layer consists of minor nodes. These are selected vehicles that store data generated by sensors and applications. The hierarchy between the higher-order nodes and the lower-order nodes allows to set different rights for different levels of the overlay: the system nodes are allowed to make specific data requests to the miner nodes, to gather information from sensors, while the miner nodes are only allowed to get general information from the higher-order nodes.

The authors of [47] look at the problem from a different angle: They focus on the communication between two road members through side-channels such as visible light and

acoustic signals. To solve the problem of limited data throughput while trying to validate the other communication partner securely, centralized approaches were discarded due to the high number of existing manufacturers and standards. Instead, a Blockchain based Domain Name System (DNS) is proposed. When two vehicles meet, their communication needs to be extremely fast and work in the absence of any internet connection or any centralized infrastructure. To make this possible, Blockchain technology is used as a way to replace the need for a central authority when authenticating another vehicle. By creating a decentralized DNS that binds the license plate value and identity together using a certificate, it can create a good base for a secure session between two vehicles.

In the presented ideas, the Blockchain takes on different roles. Moreover, while the approach by [47] uses it as a replacement for a DNS service, the approaches by [45] and [46] think further and propose actual ecosystems including other IoT devices in the use case of a smart city. The thought of having multiple layers of nodes that, even though all incorporate the Blockchain, have different rights to write and access the Blockchain, seems very promising and suits well to the hierarchical structures that are necessary to organize a smart city network. In some way, it can be thought of as a system of checks and balances, where some nodes have more rights than others, but their actions can always be seen and verified from everywhere at any time.

Applying the schema represented by the flowchart in figure 7.8, we identify the main issue again in the existence of a trusted third party. Whether it exists explicitly such as in the systems of [46] or implicitly such as in the case of [47] or [45], we have a third party that needs to come into play in all three scenarios. According to [12], this can only lead to two possible outcomes: Either, the use of the Blockchain is not necessary, or a private permissioned Blockchain should be chosen [12], reducing the Blockchain to a decentral database system. Nevertheless, the IoT setting fits the Blockchain particularly well, because the data generated by sensors can be incorporated right away without any trusted third parties.

7.3.4.4 Personal Data Protection and Sharing in the medical sector

The medical sector has two concerns regarding data, the sharing of data, and the protection of data. On one hand, sharing private data is necessary to ensure safe medication and even save lives, on the other hand, the data is extremely sensitive and can inflict big problems on individuals if made public. As an additional and particularly compelling aspect, the combination of IoT devices and smart contracts needs to be mentioned. With the use of smart contracts, IoT devices carry out autonomous transactions on the Blockchain, while guaranteeing access control and data validity at the same time.

In the health-care sector three traditional models exist to deal with the facility of interoperability of medical data: The push model, where medical information is sent from one provider to another, the pull model, where one provider asks another provider for information and the view model, where one provider looks at the record of another provider [48]. A significant drawback to all of these models is that the data is not audited or tracked in a standardized way. For example, if a patient is transferred to a different hospital, the new hospital may not be able to access data that was not "pushed." This means that there is no guarantee that the data's integrity is maintained over time. It is argued by [48] that the Blockchain offers a fourth model, which makes it possible to share medical records securely across providers without needing to include a trusted third party into the process. It leaves the data owner in full control about what is shared and leaves a verifiable audit trail of the data behind [48].

The MedRec system by [49] is a Blockchain-based network to share and protect clinical data. Via smart contracts on an Ethereum Blockchain, it logs patient-provider relationships, that associate a medical record with viewing permissions, and data retrieval instruc-

tions for the execution on external databases. The systems nodes consist of providers, that require a full server and database infrastructure, and patients, that can use any mobile device or web-interface and use only a local database with their data. Providers can add new records associated with a particular patient, and patients can authorize the sharing of records between providers. The involved parties are notified by automated notifications when changes occur or new data is added. To avoid unintended or abusive use of data, different policies are put in place by the owner and carried out by smart contracts. To create incentives to run a Blockchain node, MedRec offers block rewards and anonymized patient data.

The authors of [50] focus on a different aspect and point out that existing schemes cannot ensure the correctness and integrity of outsourced medical records. To address these problems, [50] propose a secure cloud-based system that runs without introducing any trusted third party, in the following way: Each patient makes an appointment with the hospital and obtains a treatment key for diagnosis. With the treatment key, a secure channel between the patient, hospital, and the doctors is established. The patient generates warrants to delegate to doctors, that indicate the identities of the treating doctors for specific treatment time and the needed auxiliary medical information [50]. During the treatment time, doctors generate medical records for the patient and store them on the Blockchain [50]. The system is protected against medical record modification attacks, impersonation attacks, and a change of timeline whether by a hacker or an insider, like a doctor.

While the approaches by [50] and [49] both offer a different level of extensiveness, they both make good use of the Blockchains ability to generate an audit trail and run user-configurable smart contracts autonomously, providing at the same time proper security mechanisms against the misuse of and the falsification of patient data. The use of the Blockchain is challenged though by the question "Are all writers trusted?" as asked by [12]. It shows that the real problem lies in the "transitioning"-phase of the data: The moment when the real world gets projected by a human being onto the digital world, imposes the vulnerability. We must, therefore, ask the question: If we do not trust the doctors to create proper medical records, why do we consult them? Moreover, if we do trust them, why do we need a mechanism that prevents the doctors from forging medical records?

The question whether such a system must be run on the Blockchain can therefore not be answered finally. The Blockchain is used to make the system more secure. However, the Blockchain cannot take over the responsibility for human or institutional malevolence.

The whole field of IoT devices also includes the field of Wireless Body Area Networks (WBANs). In a WBAN, a patient is equipped with one or multiple wearables or implanted medical devices, that take real-time measurements of vital indicators, such as heart rate or glucose levels. All devices report to a master device that collects and aggregates the data and offers a user-friendly interface.

To address the security concerns connected to such a setup, [51] propose integrating WBAN systems with smart contracts on a consortium-managed Blockchain, creating an immutable log of the transactions between WBAN devices and the health care providers. [51] In the presented system, the data of all devices get collected and aggregated on a master device (*e.g.*, a smartphone) and is then, together with customized threshold variables, sent to a specific smart contract on the Blockchain. The smart contract evaluates the data and issues alerts to both patient and healthcare provider, as well as automated treatment instructions to the actuator nodes, if necessary. That way, no medical information is stored on the Blockchain or in the smart contract. Instead, only the success (or failure) of the transactions gets stored. The data itself is forwarded to a designated storage database. To authenticate the data later on, the Blockchain transactions can be linked to the storage base where the patient's medical history lies. The systems nodes

consist of healthcare companies. This allows only designated members to read the blocks, execute smart contracts, or verify new blocks.

Compared with traditional systems, the system offers higher performance regarding the availability, the immutability, privacy as well as the transparency of the data. Regarding confidentiality and speed, the proposed system performs worse or equal than traditional systems [51]. Nevertheless, because the traffic of the master device is routed via the user's internet connection, the data is possibly transferred over an open channel. Security issues connected to this problem, are not yet addressed by the system [51].

This system by [51] fundamentally differs from the others in one aspect: The initially generated data is generated by an IoT device and is per se, digital. Therefore no translation step happens between the generation and the incorporation of the data. Consequential, the system does not just use the Blockchain as a replacement for any database, but - especially in combination with the smart contract - as a real entity that supports the interconnection of IoT devices with the service provider in a distributed, secure and tamper-proof manner. The fact that it does not use the Blockchain to actually store data but only to log the transitions emphasizes this aspect.

An uncovered topic in all the work on the Blockchain in the medical sector is the "right to forget," as granted by the GDPR [52]. At least from a theoretical point of view, a traditional system allows the deletion of a record. Meanwhile, a Blockchain based system does per design not make it possible to delete anything at all, as it is stored within the chain, in an unmodifiable way. In this regard, the papers of [50] and [49], that store data on the Blockchain itself, have to be treated as approaches that violate the GDPR.

7.3.4.5 Concluding remarks

The covered topics show a broad use of the Blockchain to increase systems security. Most of the time, the Blockchain is chosen as a way to store data in a tamperproof manner and to be able to verify the integrity of the data at all times. Most approaches, however, try to reduce the Blockchain to a distributed database. The result is always a private permissioned Blockchain that has no real solution for the initial problem: The incorporation of the data itself into the Blockchain. The human as the first actor and the problem that arises with this fact, namely that we cannot outsource trust to a machine, is neglected.

Promising, on the other hand, are use cases where such a human translation-factor does not exist per se. Use-cases that already have digital assets at hand, suit a Blockchain scenario well. Such approaches can exploit the advantages and properties of the Blockchain technology at its fullest, without twisting it to fit a specific purpose.

7.4 Final Considerations

The goal of this section is to summarize the key points of this paper as well as to provide a discussion in the context of the research shown in this work.

7.4.1 Summary

After providing some foundational concepts in the areas of cybersecurity, Blockchain, and smart contracts, the potential use of Blockchain technology for combating DDoS attacks, for decentralizing PKI, for securing the IoT, as well as for some specific applications is elaborated. In conclusion, Blockchain technology is found to be a good choice whenever there is a need for integrity and decentralization.

For dealing with DDoS attacks, Blockchains and smart contracts can be used to serve as a decentralized database to store white- and blacklisted IP-addresses. Blockchains and smart contract also facilitate the sharing of information. Various mitigation techniques such as SDN and NFV can be combined and facilitated with Blockchain technology and smart contracts.

In PKI, a significant danger is a certificate authority being hacked. Decentralizing PKI averts this danger. Namecoin is a Blockchain with the goal to store all DNS-related naming information in a Blockchain. There are Blockchain-based PKI solutions that build upon Namecoin to decentralize the PKI architecture. With Gazal, the traditional web certification model is moved to the Ethereum Blockchain with a smart contract specifying all procedures regarding the registration, transfer, and expiration of domains, as well as the associated values and certificates.

In IoT, Blockchain technology is proposed to help with the integrity of IoT datasets. IoT devices like sensors typically generate lots of data. Blockchain preserves the integrity of IoT generated datasets. Further, IoT and Blockchain combinations are a useful combination in managing IoT devices. This can be along a supply chain where IoT devices change ownership or have to record transactions, or in a smart home where access roles between all smart devices are defined in smart contracts, securing the IoT network from malicious tampering.

Furthermore, Blockchain technology can be used in various concrete applications, enhancing their security. In e-voting systems, Blockchains can be used as a transparent and incorruptible database. In smart cities, autonomous vehicle Blockchain technology can be applied to enhance cybersecurity. In this area, Blockchain helps to solve the problem of centralization, unscalability, and unsecured communication architectures by its very nature. In the healthcare sector, Blockchain can be used to share patient data between hospitals based on an auditable trail. In one of the proposed systems, the MedRec systems, patients can authorize what data to share and with whom.

The question remains whether Blockchains are really needed and if they solve a problem or shift the problem somewhere else.

7.4.2 Discussion

As shown in this paper, there are various ways in which Blockchain technology can enhance cybersecurity. However, Blockchains have their limitations. They use much power and don't scale well. This is a problem, especially for IoT applications where devices are designed to use little power and have little storage. Therefore the Blockchains are sometimes adapted to fit their use case better and, for example, to use trust instead of proof of work.

Before trying to apply Blockchain technology to solve a problem, it is important to first reflect on the question of whether a Blockchain is really needed, applying a process as

shown in the flowchart in Figure 7.8. Fundamentally, Blockchains cannot solve problems of human nature. In e-voting, if the government itself is not trusted, how can one trust the Blockchain set up by the government? Nevertheless, there is a place for Blockchain technology for helping with cybersecurity as this paper has shown by collecting examples thereof.

Bibliography

- [1] M. E. Whitman and H. J. Mattord, *Principles of information security*. Cengage Learning, 2011.
- [2] M. Bishop, “What is computer security?,” *IEEE Security and Privacy*, vol. 1, no. 1, pp. 67–69, 2003.
- [3] D. Kahn, *Seizing the Enigma: The Race to Break the German U-Boats Codes, 1939-1943*. 1991.
- [4] C. P. Pfleeger, S. L. Pfleeger, and J. Margulies, *Security in Computing*. 2014.
- [5] M. Bishop, *Introduction to Computer Security*. 2004.
- [6] R. Von Solms and J. Van Niekerk, “From information security to cyber security,” *Computers and Security*, vol. 38, pp. 97–102, 2013.
- [7] The White House, “National Cybersecurity Awareness Month,” 2009.
- [8] S. Shackelford and S. Myers, “Block-by-Block: Leveraging the Power of Blockchain Technology to Build Trust and Promote Cyber Peace,” *Yale Journal of Law and Technology (2017 Forthcoming)*, vol. Kelley Sch, pp. 16–85, 2016.
- [9] K. Stinchcombe, “Ten years in, nobody has come up with a use for blockchain,” 2017.
- [10] R. Nielsen, “The Blockchain Is A Solution In Search Of A Problem Whistling In The Wind,” 2018.
- [11] B. Lunn, “Bitcoin Blockchain could solve the cyber security challenge for Banks,” 2015.
- [12] K. Wüst and A. Gervais, “Do you need a Blockchain?,” *IACR Cryptology ePrint Archive 2017*, no. i, p. 375, 2017.
- [13] “The Trust Machine,” 2015. <https://www.economist.com/leaders/2015/10/31/the-trust-machine>, last visit 2019-04-06.
- [14] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
- [15] “Coinmarketcap.” <https://coinmarketcap.com/>, last visit 2019-04-14.
- [16] D. Siegel, “Understanding the dao attack,” 2016. <https://www.coindesk.com/understanding-dao-hack-journalists>, last visit 2019-05-16.
- [17] “Ethereum yellow paper.” <https://ethereum.github.io/yellowpaper/paper.pdf/>, last visit 2019-04-14.
- [18] “Blockgeeks ethereum gas.” <https://blockgeeks.com/guides/ethereum-gas/>, last visit 2019-04-14.

- [19] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, p. 39, 2004.
- [20] Cloudflare, “IP Spoofing,” 2019. <https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/>, last visit 2019-03-29.
- [21] B. Rodrigues, T. Bocek, and B. Stiller, “Multi-domain DDoS Mitigation Based on Blockchains,” vol. 10356, pp. 185–190, 2017.
- [22] S. Mansfield-Devine, “The growth and evolution of DDoS,” *Network Security*, vol. 2015, no. 10, pp. 13–20, 2015.
- [23] A. Pras, R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and R. Sadre, “Measuring the Adoption of DDoS Protection Services,” pp. 279–285, 2016.
- [24] S. Mannhart, B. Rodrigues, E. Scheid, S. S. Kanhere, and B. Stiller, “Toward Mitigation-As-A-Service in cooperative network defenses,” *Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and IEEE 3*, pp. 362–367, 2018.
- [25] B. Rodrigues, B. Stiller, E. Scheid, S. S. Kanhere, and J. Gresch, “The Proposal of a Blockchain-Based Architecture for Transparent Certificate Handling,” pp. 185–196, 2019.
- [26] T. Straub, *Usability Challenges of PKI*. PhD thesis, Technische Universität, Darmstadt, 2006.
- [27] C. Adams, S. Farrell, T. Kaese, and T. Mononen, “Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP),” 2015.
- [28] M. Zusman, “Criminal Charges are not pursued: Hacking PKI,” 2008.
- [29] Z. Wang, J. Lin, Q. Cai, Q. Wang, J. Jing, and D. Zha, “Blockchain-Based Certificate Transparency and Revocation Transparency,” in *Financial Cryptography and Data Security* (A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, and M. Sala, eds.), vol. 10958, pp. 144–162, Berlin, Heidelberg: Springer Berlin Heidelberg, 2019.
- [30] “Namecoin.” <https://namecoin.org/>, last visit 2019-04-14.
- [31] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, “Blockstack: A Global Naming and Storage System Secured by Blockchains,” p. 15, 2016.
- [32] Y. Dong, W. Kim, and R. Boutaba, “Bitforest: a Portable and Efficient Blockchain-Based Naming System,” p. 7, 2018.
- [33] S. Moosavi and J. Clark, “Ghazal: Toward Truly Authoritative Web Certificates Using Ethereum,” in *Financial Cryptography and Data Security* (A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, and M. Sala, eds.), vol. 10958, pp. 352–366, Berlin, Heidelberg: Springer Berlin Heidelberg, 2019.
- [34] A. Buldas, A. Kroonmaa, and R. Laanoja, “Keyless Signatures Infrastructure: How to Build Global Distributed Hash-Trees,” in *Secure IT Systems* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y.

- Vardi, G. Weikum, H. Riis Nielson, and D. Gollmann, eds.), vol. 8208, pp. 313–320, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [35] C. Jamthagen and M. Hell, “Blockchain-Based Publishing Layer for the Keyless Signing Infrastructure,” in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld)*, (Toulouse, France), pp. 374–381, IEEE, July 2016.
- [36] Statista, “Forecast economic impact of the internet of things.” <https://www.statista.com/statistics/580778/worldwide-internet-of-things-economic-impact-forecast/>, last visit 2019-05-16.
- [37] M. Banerjee, J. Lee, and K. K. R. Choo, “A blockchain future for internet of things security: a position paper,” *Digital Communications and Networks*, vol. 4, no. 3, pp. 149–160, 2018.
- [38] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.
- [39] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, pp. 618–623, 2017.
- [40] A. Ben Ayed, “A Conceptual Secure Blockchain Based Electronic Voting System,” *International Journal of Network Security & Its Applications*, vol. 9, no. 3, pp. 01–09, 2017.
- [41] K. Kirby, A. Masi, and F. Maymi, “Votebook: A proposal for a blockchain-based electronic voting system,” p. 14, 2016.
- [42] Blockchain Technologies Corporation, “VoteWatcher - The World’s Most Transparent Voting Machine.” <http://votewatcher.com/>, last visit 2019-04-15.
- [43] R. Osgood, “The Future of Democracy,” 2016.
- [44] I. Follow My Vote, “The Online Voting Platform of The Future - Follow My Vote.” <https://followmyvote.com/>, last visit 2019-04-15.
- [45] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, “BlockChain: A Distributed Solution to Automotive Security and Privacy,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.
- [46] P. K. Sharma, S. Y. Moon, and J. H. Park, “Block-VN: A distributed blockchain based vehicular network architecture in smart city,” *Journal of Information Processing Systems*, vol. 13, no. 1, pp. 184–195, 2017.
- [47] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. M. Goldrick, “Securing Vehicle to Vehicle Communications using Blockchain through Visible Light and Acoustic Side-Channels,” 2017.
- [48] N. Kshetri, “Blockchain’s roles in strengthening cybersecurity and protecting privacy,” *Telecommunications Policy*, vol. 41, no. 10, pp. 1027–1038, 2017.

- [49] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “MedRec: Using blockchain for medical data access and permission management,” *Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016*, vol. 13, pp. 25–30, 2016.
- [50] S. Cao, G. Zhang, P. Liu, X. Zhang, and F. Neri, “Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain,” *Information Sciences*, vol. 485, pp. 427–440, 2019.
- [51] A. N. Baccarini, K. N. Griggs, E. A. Howson, O. Ossipova, T. Hayajneh, and C. P. Kohlios, “Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring,” *Journal of Medical Systems*, vol. 42, no. 7, pp. 1–7, 2018.
- [52] European Commission, “General Data Protection Regulation (GDPR),” 2017. <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN{#}d1e2589-1-1>, last visit 2019-04-10.

Chapter 8

The Hyperledger Fabric

Karim Abou el Naga, Danjiel Dordevic, Claude Muller, Lucas Thorbecke

8.1 Abstract

It's been ten years since the inception of Bitcoin and the big blockchain hype is over. Yet we still have to find productive real-world applications that go beyond a proof of concept or speculative trading. The big promise of blockchain technology to build trust and transparency often clashes with privacy, security or performance requirements of businesses. Therefore the open-source Hyperledger Fabric project was started under the care of the Linux Foundation and IBM. Fabric is a modular and extensible framework for deploying and operating permissioned blockchains. Its promise is to make blockchain technology enterprise ready. To do that it introduces a novel blockchain architecture that redefines how the blockchain copes with non-determinism, consensus finding and performance. Fabric allows to write smart contracts (called chaincode in Fabric) in general purpose programming languages such as Java or GO while not having to rely on a native cryptocurrency. In this paper, we are discussing the projects background, its network architecture, the implications on the transaction flow and what it means in terms of privacy and security. Finally, we evaluate its performance.

Contents

8.1	Abstract	205
8.2	Introduction	207
8.3	The Linux Foundation and Hyperledger Introduction . . .	207
8.3.1	The Linux Foundation	207
8.3.2	Hyperledger Introduction	209
8.3.3	Hyperledger Fabric Introduction	210
8.3.4	Hyperledger Fabric Model	212
8.4	Network Architecture and Application	213
8.4.1	Networking inside the Project	213
8.4.2	IT-Integration of HLF	216
8.4.3	Use cases and usage of Hyperledger Fabric	220
8.5	Transaction flow and its privacy/security implications . . .	222
8.5.1	Data Processing through Hyperledger	222
8.5.2	Privacy	224
8.5.3	Trust	226
8.6	Performance Evaluation	228
8.6.1	Performance Metrics	229
8.6.2	Performance Impacts	229
8.6.3	Performance Conclusion	233
8.7	Summary and Conclusion	233

8.2 Introduction

Since the beginning of the blockchain hype, a lot of projects have been started in this area. In their first generation, blockchains like Bitcoin mainly focused on value transfer. They allow us to make monetary transactions without the need for a trusted intermediary. The second blockchain generation introduced smart contracts that allow more complex operations than simple value transfers to be performed via the blockchain. Multiple parties can now participate in a contract without the need for a third party or trust in each other. Currently, the third generation is unfolding. This generation concerns itself with issues of scalability, governance, and interoperability between different blockchains. While this evolution was happening in the domain of public blockchains where everybody can participate, businesses were looking for ways to integrate the new technologies into their business cases. This usually means that the access to the blockchain needs to be restricted and authenticated. From that permissioned and private blockchains emerged and today three main enterprise-oriented distributed ledger technologies (DLT) are available. Corda from the R3 consortium, Quorum from J.P. Morgan, and Hyperledger Fabric from the Linux Foundation. They all try to tend to the blockchain needs of enterprises but differ in their architecture and functionalities, also because they have different industry backgrounds. They took the basics from the public blockchains, like smart contracts, and adapted or enhanced it according to the permissioned context.

This report takes a closer look at Hyperledger Fabric. Hyperledger Fabric, short Fabric, was created by merging work from the companies Digital Assets, Blockstream, and IBM [1]. Hyperledger itself is an umbrella project for open source blockchain technologies started by the Linux Foundation in December 2015, with Fabric being one of the most successful projects in it. Hyperledger is supported by many member and associated organizations from different industries with IBM being one of them. This and the fact that the project is fully open-sourced and maintained by the Linux Foundation promises a bright future and wide adoption in different industries. Stable versions of Fabric for production are available but further development is still in full swing. Since version 1.0 Fabric employs a novel technical architecture that differentiates it from most other blockchains and has many benefits. New enhancements in areas like privacy and performance are on their way into future releases and will make Fabric a capable business blockchain. Throughout this report, the key components of Fabric are discussed and important features pointed out and evaluated.

The rest of this report is structured as follows. Section 8.3 introduces the Linux Foundation and some of its projects. It then proceeds with an introduction to the Hyperledger project and Hyperledger Fabric in particular. Section 8.4 explains the parts and services that make up a Fabric network and how the network can be operated. It also shows three use cases in which Fabric was employed. Section 8.5 gives a more detailed insight into Fabric's transaction flow and what its privacy and security properties are. Section 8.6.3 takes an in-depth look at Fabric's performance. Finally, Section 8.7 summarizes and concludes the findings of this report.

8.3 The Linux Foundation and Hyperledger Introduction

8.3.1 The Linux Foundation

It all started with Linux and the open source revolution in the early 2000s. Later on, The Linux Foundation took the game to the next level so it would be fair to say that this organization is more than just Linux. Its story began in 2000 when it was established by

two consortia, the Open Source Development Labs and the Free Standards Group. Their initial idea was to standardize Linux and promote its adoption to the public at that time. Over the course of time, they have broadened their activities to many other open source projects, among which is the Hyperledger Fabric.

Nowadays, the Linux Foundation hosts hundreds of mission-critical, open-source projects across the globe. Big organizations have recognized the importance of collaboration so the outcome is that the Linux Foundation today has more than a thousand commercial companies as its members, all contributing to or supporting new projects, one way or another [2].

Since the number of projects under the Linux Foundation's umbrella has grown immensely, the management team increasingly felt the need to group and label them under several categories, so that each of the new projects can be properly classified, managed and accessed. A few of the most common ones will be briefly explained in the following sections.

8.3.1.1 Security

Security is paramount to the Internet. Consequently, that was prioritized by the Linux Foundation and they decided to offer open-source security socket layer (SSL) and transport layer security (TLS) certificates to site owners so as to prove their ownership and help them prevent malicious attacks. Their aim was to achieve 100% encryption on the Web. For that purpose, in 2014, several members created the certificate authority called Let's Encrypt [7]. The main goal of Let's Encrypt was to simplify the process of enabling HTTPS traffic on the websites, possibly in a fully automated manner without any human interaction. That was achieved by utilizing the ACME protocol [4]. Soon after its launch, the Let's Encrypt became the most used certificate authority across the globe, with over 50% of the global market share as of today [8].

8.3.1.2 Cloud

Cloud computing services have been adopted by almost every organization today. They see the benefits of the Infrastructure as a Service (IaaS) model. The global cloud computing market size has been expected to grow at a compound annual growth rate of 18% per year, from 2018 to 2023 [11]. The Linux Foundation member had noticed that cloud computing overtook the market so they had an idea to create an initiative with the aims to create a portability layer and to accelerate 'cloud-native' computing dev-ops, containers, microservices.

For that purpose, in 2015 the Linux Foundation together with Google created the Cloud Native Computing Foundation (CNCF) [3]. The project seeded with Kubernetes which was donated by Google. Kubernetes is the biggest project among many others hosted by CNCF. It is an orchestration tool for convenient deployment and scaling of containerized applications. With it, one can manage computing, networking, and storage infrastructure for a set of application containers depending on workloads [9]. This brings huge benefits to the users since now that are able to scale their environments based on needs, and therefore support the increasing amount of traffic.

Today, CNCF has more than 250 members and hosts 14 additional projects beyond Kubernetes.

8.3.1.3 Automotive

The goal of the Automotive Grade Linux (AGL) is to create a Linux OS stack that meets common requirements in the automotive industry. It encompasses a wide variety of

products, from the software that makes supply chain management more efficient, to the software that helps self-driven vehicles to operate.

In order to support the development of open source software for automotive applications, the world's biggest players in automotive industries have joined the project [10]

8.3.1.4 Blockchain

"The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value."[5] Blockchain technology created an underlying structure that provides a new way of how people make transactions. Everything started in 2009 when the Bitcoin white paper was published by Satoshi Nakamoto [12]. This paper claims that there is a new peer-to-peer electronic cash system which allows us to avoid going through financial institutions. This means that the people now can execute the transactions between each other without the middle man.

Since then, the blockchain technology had been increasing in popularity. People use various blockchain technologies/platforms more and more, but the use cases are usually limited to executing peer-to-peer transactions. Also, it was noticed that such systems have some issues, usually related to scaling, excessive energy usage, 51% attack and others. The Linux foundation noticed a huge potential besides all those issues and wanted to tap into its potential and decided to develop and launch an enterprise-grade blockchain system in 2016. The idea was to develop a system that tackles the aforementioned problems and brings the blockchain technology to the businesses. The outcome is the Hyperledger collaborative effort amongst 30 founding members who are big players in many industries [15].

8.3.2 Hyperledger Introduction

The Hyperledger is basically a blueprint meant to restructure the underlying data structure and thus bring the distributed ledger technology such as smart contracts to the business world.

Its objective is to improve cross-industry collaboration by utilizing blockchain technology beyond cryptocurrencies. Also, the goal is to improve the performance and reliability of those systems.

The project has attracted many companies to join and contribute, therefore the Hyperledger project is the fastest growing open-source project hosted by the Linux Foundation [6].

Today, the Hyperledger project is an umbrella project for many different projects/frameworks and tools as shown in figure 8.1. Short explanation about few of the most important Hyperledger frameworks is given in the following section.

8.3.2.1 Hyperleder Burrow

It is an Ethereum Virtual Machine (EVM) permissioned blockchain ledger. Its aim is to be fast, easy to deploy and make an EVM compliant blockchain technology accessible to the masses. Hyperledger Burrow nodes are capable of executing smart contracts written in Solidity programming language. Hyperledger Burrow consists of the following components:

- **Consensus Engine** - It uses the Tendermint protocol for the consensus protocol [16]. This allows high transaction throughput while preventing the forking of the network.

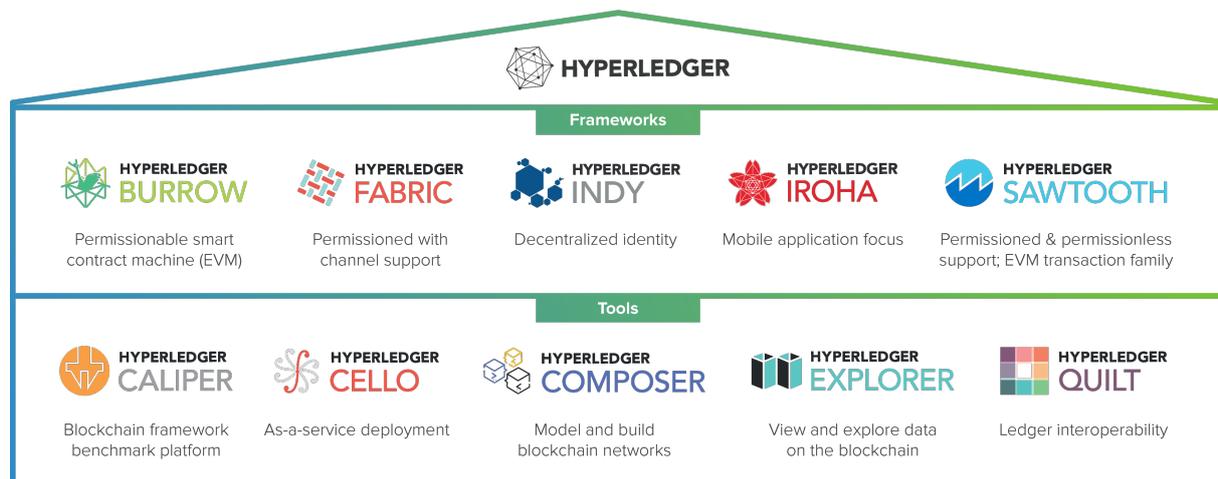


Figure 8.1: Hyperledger Framework Ecosystem [13]

- **Permissioned Ethereum Virtual Machine** - Since it allows the execution of the solidity smart contract, it implements the Etnereum Virtual Machine, but only in the permissioned mode, meaning that only certain nodes are able to call certain smart contracts. This is achieved by implementing secure native functions that underlay all smart contracts.
- **API Gateway** - In order to allow users to interact with the network, Burrow exposes REST and JSON-RPC APIs. It also implements WebSocket API which allows full-duplex communication over a single TCP connection.

8.3.2.2 Hyperledger Indy

Its main purpose is to build a decentralized identity by providing tools, libraries, and components build on top of the distributed ledgers. It creates the globally unique decentralized identities without any need for a centralized resolution authority. Hyperledger Fabric uses the pairwise identifiers that create one-to-one relationships between any two entities within the system. The identities are verified and exchanged within the system via a standardized pipeline designed by W3C¹. Other Hyperledger systems can use Indy as a component just for managing the identities of their users. This allows them to follow the microservice architecture.

8.3.2.3 Hyperledger Sawtooth

It is built from the ground up targeting the enterprise use, Hyperledger Sawtooth tries to make smart contract more secure and scalable. Its model uses a single node type which simplifies deployment. Participants can agree on configuration changes using transactions within the system, even the consensus algorithm can be changed in runtime. It also supports a wide variety of languages used to write smart contracts, some of the languages supported are: Solididy, Java, JS, Web Assembly. All those smart contracts written in different languages can work alongside each other.

8.3.3 Hyperledger Fabric Introduction

Thousands of companies are helping to build a Hyperledger Fabric able to support production business networks. The work started in 2015 with a simple framework that was meant to test the interaction between various applications and blockchain networks. It

¹<https://www.w3.org/>

allowed the testing of many different use-cases from many different industries, such as healthcare, capital markets, manufacturing, supply-chain, and others. While testing, it was inferred that every single node had simply too many responsibilities, such as:

- Executing every transaction
- Storing the whole ledger
- Running a consensus algorithm

This explains why traditional blockchain technologies have failed to provide systems that scale well and that can be implemented in many different industries, but they rather solve just very specific problems.

The aforementioned motivated the Hyperledger community to design a system that is scalable, secure and that can cater to many different use-cases from many different industries due to its modularity and configurability. This required the redesigning of many components from the traditional blockchain technologies. Some of them are:

- Modular and configurable architecture that enables the system to fit into different use cases across different industries.
- It is a permissioned system, meaning that the actors are known to each other beforehand and the permission rules can be applied with more granularity.
- It is the first blockchain system that allows us to run the smart contracts written in some of the general purpose programming languages. In 2015 only Go was supported, but, as of May 2019, two more (`Java` and `Node.js`) have been added and are currently supported. This eases the adoption and usage of the smart contracts since no additional effort in learning a new domain-specific language (such as Solidity for Ethereum) is required.
- It supports a pluggable consensus protocol which allows the system to be more efficient and cost-effective in particular use-cases [14].

The most notable change was that the nodes became specialized in different types of tasks[17]:

- **Client:** It acts on behalf of the end-user. It can invoice transactions, meaning they have to connect to other peers of its choice.
- **Peer:** Peers process transaction proposals from clients, receive finalized blocks from the ordering service and maintain the state and the ledger.
- **Ordering-service-node or orderer:** it schedules transactions on behalf of the clients.

Hyperledger Fabric is therefore a private and permissioned blockchain system. This offers a different way for the members to join and interact in the network. In most other blockchain networks, members have to input a certain amount of processing energy which is to be invested in validating the transactions (e.g. proof of work) that allows them to become members. On the other hand, a Hyperledger Fabric member can join the network through a trusted Membership Service Provider (MSP).

Since not all members should keep track and store every transaction, it is possible to create a separate ledger where only certain nodes can participate. This increases the security because only certain members have access to certain information, and also improves the scalability of the system since not all the data is replicated on each node, so the system is not overloaded.

8.3.4 Hyperledger Fabric Model

In the previous section, it was claimed that Hyperledger Fabric solves many problems of the traditional blockchain technologies. In this section, a few key design features of Hyperledger Fabric are explained, as well as the way those design features aim to solve those problems [18].

8.3.4.1 Channels

A channel represents a private isolated network of specific peers. Channels provide private and secure transactions only between the peers that are the members of that channel. Transactions are executed within a channel and they are not visible outside that channel. In order to become members of a specific channel, the peers have to be authenticated and authorized by the membership service provider (MSP). Upon successful authentication, the MSP assigns an identity to each member of the channel.

Even though a node can belong to multiple channels, there is no possibility to pass ledger data from one channel to another.

Channels allow the organizations to execute confidential transactions while on the same networks as e.g. their competitors.

Another benefit is that by using channels, authenticated members do not need to keep the whole ledger history, but just the one corresponding to the ledgers of the specific channels they belong to.

8.3.4.2 Smart Contracts

While ledgers keep a record of the history of the current state of the network and also keep track of all executed transactions, smart contracts define the executable logic that generates/triggers new transactions that are recorded on the ledger. As already mentioned, the Hyperledger Fabric is the first blockchain system that allows the execution of smart contracts that are written in some of the general purpose programming languages. Smart contracts are bundled into a so-called chaincode as shown in the figure 8.2 which gets deployed to a blockchain network. This means that one chaincode can encompass multiple smart contracts that are logically grouped together.

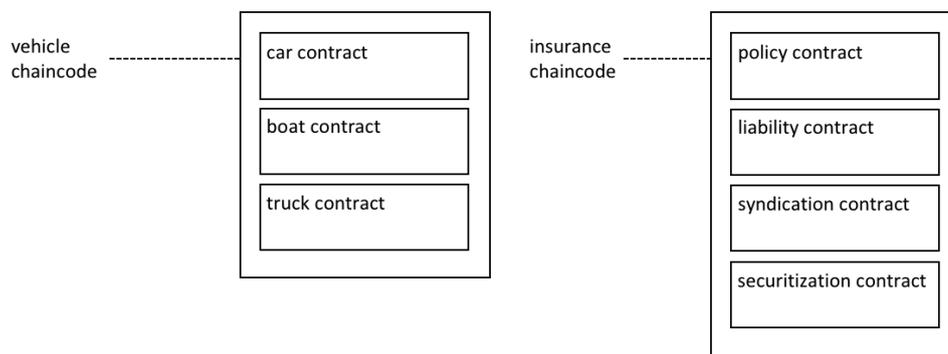


Figure 8.2: Relation between Smart Contracts and Chaincode [19]

8.3.4.3 Shared Ledger

The ledger is the sequenced, immutable record of all transactions in the system. The transactions can be triggered by chaincode invocation. Each transaction is stored on the ledger as a set of key-value pairs that can create, update or delete some asset in the system.

The first difference compared to traditional blockchain systems is that in Hyperledger Fabric there is one ledger per channel. Meaning only the members of the channel store and maintain that particular ledger.

The second difference is that Hyperledger Fabric introduces a new way of handling the ledger system. Each ledger is made of the two components:

- **World State:** It is referred to as a World State since it stores the key-value entries from the genesis block to the latest one, meaning it stores the whole history.
- **Transaction Log:** It serves as an update history of the world state. It records all transactions which result in the current value of the world state.

This means that the main ledger is made of the world state and transaction logs.

8.3.4.4 Consensus

Since the Hyperledger Fabric is a permissioned blockchain system, that implies that all its members are verified and they trust each other to a certain extent. This allows the usage of traditional Byzantine-fault tolerant algorithm as opposed to Proof-of-Work algorithms. Furthermore, Hyperledger Fabric supports modular consensus approach meaning that different consensus algorithms can be used based on the requirements.

8.4 Network Architecture and Application

8.4.1 Networking inside the Project

8.4.1.1 Blockchain Network

To understand how the networking within Hyperledger Fabric is working, the components of a blockchain network must be elucidated[20]. Not only on the technical side, but also the practical side, to understand how different users, e.g. different businesses, of Hyperledger Fabric form their decisions regarding their policies and implementation of such a network. A blockchain network is a framework that sets up the ledger and smart contracts (called Chaincode in this case). A smart contract is nothing more than a regular program, which can be written in most of the state of the art programming languages. The Chaincode reads and updates the data inside the ledger, in fact, only the Chaincode can execute such actions. This means that the Chaincode must be part of a given channel and it has to be installed in every peer to properly generate the wished transactions. As mentioned, in practice, there will be different users within a network, through policies the permissions will be distributed, this will only happen under an agreement of each user within the network.

8.4.1.2 Infrastructure of the Network

The three main components of the Hyperledger Fabric are the peer, certificate authorities (CA) and the ordering service. The concept of those three components forms the foundation, to understand how the network works and how transactions are executed. The ordering service can be seen as an administration point, which provides the order of operation and creates the block for the blockchain. Besides the ordering service, users must integrate CA into their network. Those certificates define the sets of users within the given blockchain network. By default, Hyperledger Fabric uses the X.509 standard for their CA and also provides a built-in CA, called Fabric-CA. Organizations though, have the possibility, to create their own CA which makes the system very flexible. Another

essential component is the MSP (Membership Service Provider). The MSP provides authority to different clients. Once they received such a credential, the clients can execute their transactions. All those components are integrated by a system administrator, who is part of an organization within the blockchain network. In practice, every organization will have a corresponding administrator. All the organizations that are working together inside the network are often called "consortium". With all those components, the foundation of the network is created. Next, let's have a look at how transactions are executed and how the network operates.

8.4.1.3 Operations and Configuration within the Network

Since new members may enter a consortium or new contracts are signed between certain parties, the network has to be flexible. Fabric delivers this through the flexibility of blockchain networks. Installing smart contracts into the nodes, creating new channels or defining consortia can all be executed rather quickly. This also means that one given system administrator could apply this to any infrastructure of any dimension. Not only does this concept work for consortia including big businesses or similar, but such a blockchain network could also be integrated for smaller consortia, which operate in any sector one could think of. Especially for developers who are engaging with Fabric, it may be of interest to have an overview of what is possible and how these operations can be integrated once the network is up and running. After the system administrators are named, they may require the following operations:

- Include a consortium
- Create or add a channel
- Adding peers
- Apply Chaincode

To execute any transaction, a network needs members. As mentioned, the term "consortium" is often used when talking about blockchain networks, which means that a group of businesses or organizations, have a shared objective. How to define one or adding new members to an existing one can be done very convenient. Once all the members are registered, the respective administrators of the different members can start defining different identities. IBM, for example, provides a console for its blockchain technologies[21]. With the help of CAs, every system administrator can then create identities and stores them in the wallet, which holds all of the information and is an essential part of a working network. This information includes an identification number, type and also optional attributes. After that, one could start to execute transactions within the network.

Inside a blockchain network, there might be still a complex arrangement on who works with whom. Channels solve that problem. If one administrator is part of a consortium, he can create channels between his organization and other parties. A channel holds certain configurations, which only the administrator of the parties who is connected through the channel can touch. This channel configuration holds its own set up and may be different to the overall network configuration. This means that for other members within the network, the channel and its configuration is invisible. After the members of a given channel have been declared, the administrators must link the channel to the Ordering Service and a peer, to execute transactions and also to be able to access the ledger.

As mentioned, the network also needs a peer. A peer contains a copy of the ledger and this is also the main function the peer has. But still, for a network administrator, it is important to integrate and connect the peer nodes properly into the network. A peer node is configured with an X.509 security key. If the peer is set up, it can be connected to a channel. Peer nodes, in general, are all the same, but they may have additional functions, which an admin can enable. Additionally to act as a committing peer, which every by default does, a peer can also be an endorsing peer. This is possible if Chaincode has been applied to a given peer. This means, that the peer then can execute a transaction, receive a response and forward it to a client application within the network. Also, a peer can be a leading peer if multiple peers are linked to a channel. A leading peer is then the anchor point for administering transactions. This means, that the Ordering Service first sends a block to the corresponding leader peer, which will then distribute them to the other

committing peers. Lastly, a peer can also be an anchor peer. An anchor peer is useful within a network, to communicate with peers that are part of another organization. After setting all those components up, the network administrators should start integrating smart contracts. As mentioned, the system administrator can define an endorsing peer, by installing Chaincode on it. The good thing is that the Chaincode just needs to be applied on one peer node within a channel. After the administrator-defined one peer as endorsing peer and the node is up and ready, the endorsing peer needs to be linked to a channel to operate. Once this is done, the members of a channel can make use of this endorsing peer; this is done via using client applications. The client application essentially sends an executed transaction to the peer, where the transaction servers as an input for the smart contract and then checks the validity of the proposed transaction. After that, a response will be returned to the client application. If the transaction is valid, it will be accepted and stored in the ledger, making it visible to every organization that is connected with the peer.

8.4.2 IT-Integration of HLF

8.4.2.1 Services and Functionalities

Since the Hyperledger Fabric goes belongs to the category of distributed ledger technology (DLT), it provides a lot of functionalities regarding scalability, performance and also security. This facilitates companies of different markets, for example, to find an individual solution to get the maximum out of this blockchain technology. Some of the major functionalities within HLF are the following;

- Identity Management
- Channel Management
- Possibilities with Chaincode

To get a specific network immaculate, proper identity management is required. Within a Hyperledger Fabric network, every user possesses a certificate with given attributes. The system administrator can configure this data. Those attributes can be defined as desired. One may be able to read a smart contract; the other may be able to set up a new Chaincode.

As mentioned, channels are also an essential part of a working and flexible blockchain network. Especially regarding privacy and communication, members of a consortium have the opportunity to make use of channels. Let's have a network including twenty organizations or businesses. In given cases, two participants of this consortium may want to share data between themselves and with no one else. Through setting up channels and integrating them, this is possible without any danger, but a more detailed report on channels will be shown in 8.5.

Since the Chaincode is nothing more than a written program, it is very flexible and may also be very individual for specific situations[22]. The developers working on the Smart Contract can set up rules and requirements which hold for everyone with the corresponding attributes. This allows a high grade of security. Additionally, the Chaincode applies to almost every dimension within the blockchain network. Configuration and requirements regarding the entire channel or transaction are all recumbent in the field of realization. The conclusion here is that using Hyperledger Fabric offers a lot of variety for developers. Scalability, efficiency and security questions can all be answered through the possibilities that HLF provides. From transactions between big companies to transactions between two small private parties, Hyperledger Fabric can offer and also cover all of those cases.

8.4.2.2 How to get started with HLF

Since this project is open-sourced, developers all over the world are continually contributing code snippets, tools and libraries to make the integration as trouble-free as possible. Before the developer can start writing applications within the Fabric, some prerequisites must be fulfilled. This means that the associates have to set up a development environment including a working blockchain network. Once the network is running properly, the developer has to understand how the Chaincode (smart contract) operates. Hyperledger administers a smart contract, called **FabCar** written in JavaScript. The developer though is open, to work with any smart contract which fits the best to the respective organization. Once these prerequisites are met, the developer can write the corresponding applications. This section will include very basic and abstracted guidance, on how the integration process may look like and how the respective operations are executed. For this, we can make some distinctions for the following steps.

- Initiating the blockchain network
- Appoint a system administrator
- Appoint users
- Working with the ledger
- Making use of FabCar
- Submit a transaction

Initiating the blockchain network:

Hyperledger Fabric provides all the required libraries and tools to get started with the network. The developer must get all the prerequisites to set up the blockchain network. All of the required data and a manual on how to specifically initiate the network can be found on the official Hyperledger website [23]. Once the network is running, the developer can start with the configuration of the network. If all of the requirements are fulfilled, the network may be launched and written applications can be installed.

Appoint a system administrator:

By creating and launching the network, the system automatically creates an admin user. This user can be accessed via "admin". The admin user is by default the registrar of the certificate authority. To make the user eligible as a system administrator he needs a private key, public key, and the X.509 certificate. This can be done via one command, which may look like this:

```
node enrollAdmin.js
```

We use the *enroll.js* function, which gets the user and allocates the proper credentials to the user. This is done by a Certificate Signing Request (CSR), where the CA takes the public key, ciphers it and is sent back to the system administrator which then has the attributes to work and operate as admin.

Appoint users:

If the system administrator is implemented, he can start registering the corresponding users. Those users are the ones, which will actively use the ledger. The different users may have different rights; therefore the proper credentials must be assigned to the respective users. The command to register a user looks like this:

```
node registerUser.js
```

Again, this program would use Certificate Signing Request, to allocate the credentials to the user, so that the network knows who is when allowed to do which operations.

Working with the ledger:

Now that all the necessary participants are registered within the network, it can execute operations on the ledger. Every peer in the network has a copy of the ledger and the Chaincode attached to it. If one user wants to for example query the ledger, he may use an application to do that. The following graphic 8.3 shows briefly how such a query would be executed.

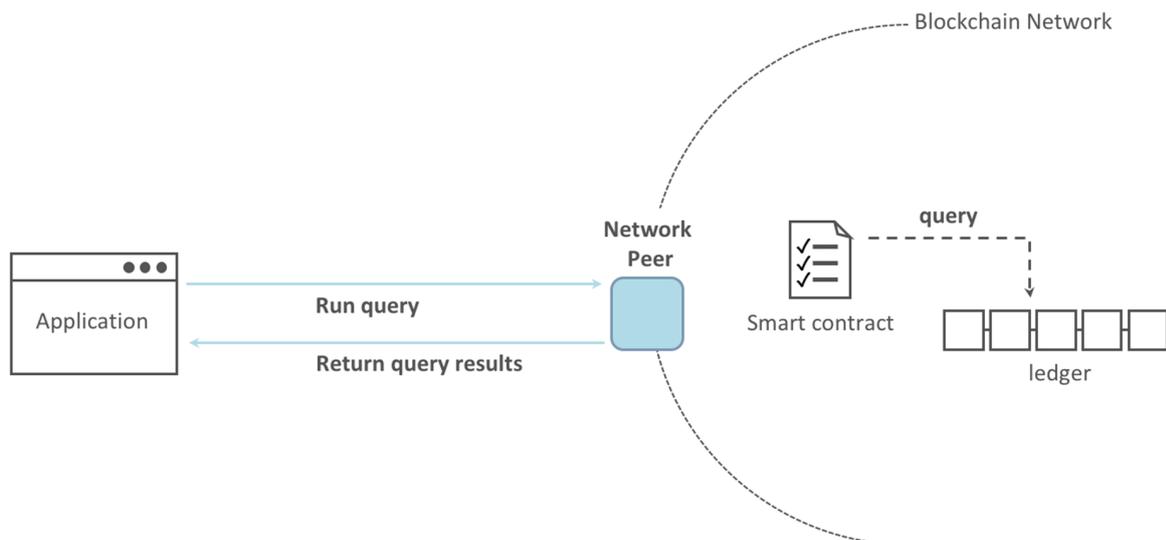


Figure 8.3: Run a query [23]

As the application will be run, it will send a query to the peer. The program gets the current state of the ledger and sends it back to the user. By default, the result will be returned as a dictionary, where the user also can get the current state of the ledger for some specific key. For example, a user could request all the transactions done by his department and work with that data. Chaincode is very dynamic, developers also can use one Chaincode to invoke another, even though it may be in a different channel, but in this case, Chaincode can only be used to read the ledger. This brings a lot of options and freedom to the developer team running the network and the business logic.

Working with Chaincode:

Chaincode is somewhat of the heart of such a blockchain network. It implements an interface so that people can start using and execute operations or transactions done to the ledger. To do that, also an application client is required which sends the commands by the users to the given Chaincode. It can be written by the developer himself in Go, Node.js or Javascript.

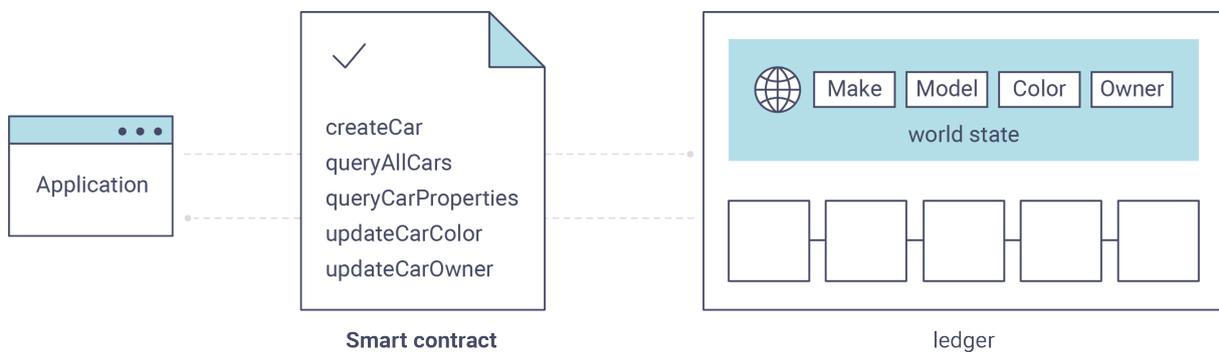


Figure 8.4: FabCar [23]

Submit a transaction:

Lastly, if a user wants to submit a transaction and update the ledger, the necessary applications are required. Again, the code is very simple and is similar to figure 8.3. The difference is that the user runs an application, which they will be sent to the ordering service. The ordering service will create a new block within the blockchain network.

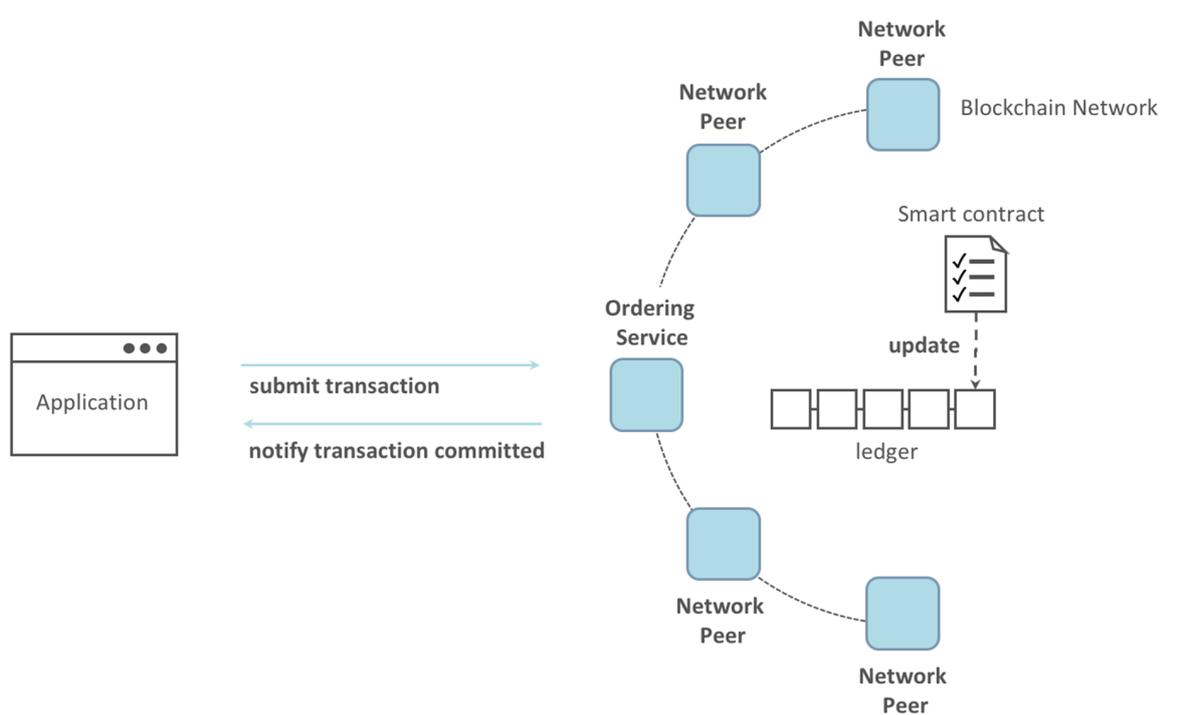


Figure 8.5: Submitting a transaction [23]

On the figure above are also all the parts of the network evident. The channel connects all the peers. All of the peers contain an Ordering Service, the ledger and the Chaincode. If the transaction is valid, the ledger will accordingly be updated. After that, all of the peers can see the transactions done within the network.

8.4.3 Use cases and usage of Hyperledger Fabric

8.4.3.1 Three different use cases

To highlight the possibility that Hyperledger Fabric holds, three different use cases which are derived from different areas. Important to mention is, that most of these use cases are in partnership with IBM itself and are rather the first step to get first impressions on how Fabric's performance is. Actual use cases, where Fabric serves as a tool used by big companies or organizations, excluding IBM, are yet to be introduced. Another interesting question is, on how IBM ultimately will use Fabric to generate monetary values. But since it is open-sourced, the most realistic approach seems to be that companies who will choose Fabric, will be provided with cloud services, subscriptions, technical support and consultancy by IBM.

8.4.3.2 Use case 1: Using Fabric to track pharmaceuticals

In April 2017, IBM reported that they partnered up with Chinese pharmaceuticals company Sichuan Hejia to solve problems regarding the distribution of the pharmaceuticals and also the securing of the payments. To do that they announced that they will use Fabric. This is a very good case, to show how well Fabric can be used to increase efficiency. Major problems for small and middle-sized companies are the underdeveloped credit system in China, credit evaluation and also a lack of risk control [25]. The modularity of Fabric offers the scalability to solve such issues, while still maintaining the necessary performance. Fabric will be used to increase the transparency for Sichuan Hejia so that the company always has an overview where their pharmaceuticals are using tracking. Additionally, they want to implement channels connected to the respective banks of their clients (mostly being hospitals [26]) to also have transparency when it comes to the payment aspect. IBM and Sichuan Hejia expect to reduce waiting times of 60-90 days to only 1-2 working days.

8.4.3.3 Use case 2: Using Fabric as a studying tool

A whole different take on Fabric was announced as well in April 2017. IBM and the University of Singapore partnered up, to provide deeper knowledge and more education within the topic of blockchain technology in general [27]. This makes a lot of sense. The demand for blockchain developers is very high and the blockchain market is a booming one; a decline is not in sight in the coming years[28]. Considering that, this is the right notion. The university and the students work together on blockchain technologies, to create viable e-banking software, but also on topics such as currencies and supply chain management[27]. With that, the students can have the first-hand experience on how blockchain technology can be used to make existing methods even more efficient. Another nice component of this course is, that the teaching staff decided to provide the course material via Fabric.

8.4.3.4 Use case 3: Fabric as energy manager

Another example of a potential use case was announced in May of 2017. Again, IBM reported that they would start working with Vandebrom, Sonnen, and TenneT [25] to im-

plement a smart energy distributing system in parts of the Netherlands and Germany. In comparison to the other stated use cases, this one is taking place on a different dimension, proving the scalability that Fabric offers. The three mentioned energy companies are leading businesses within Europe in the field of renewable energy and energy transmission[29]. With that and the fact with IBM's market position, this seems to be a promising project, which could potentially influence the future of how energy is being managed and distributed. As of now the energy grid in Germany, for example, is still a very static one [29]. This is a problematic factor, regarding the increasing demand of energy requirement for a rising number of different technologies. Smart homes are getting more attention, electric cars are about to be a new standard, with that only being a few examples. This means that energy management has to become smart and flexible. This is where IBM promises a solution with the possibilities that Fabric offers. The group is planning to update the energy grid with the inclusion of, e.g., batteries of electric cars[29] and other technologies to enhance the workload of the system. The flexibility of Fabric is then able to distribute the energy from the mentioned sources to other places where the energy might be due. This is not the only advantage that comes with blockchain technology, it would also be able, to deliver transparency on every energy transaction which was submitted, resulting in an overview and data set which provides additional information. Also, the integration of renewable energy sources to the grid would be a possibility [29].

8.4.3.5 Benefits of using HLF

One of the main benefits which are easy to be abstracted from the mentioned examples is the flexibility and the scalability the project offers. The stated use cases were all of a very different extent and, in theory, fabric proved to be one fitting solution in all of the cases. Another advantage that could be extracted from the examples was the fact that the ledger is immutable. Especially in the smart energy example, this factor is a key feature. Other features businesses might look for, before choosing a private blockchain technology might be components such as privacy, storage issues or lack of governance[30], just to mention some of the interests a business might have. Analyzing the use cases, Fabric seems to be a practical solution.

8.4.3.6 Use case evaluation

As seen in the three different use cases which were shown, Fabric offers a grand scope in theory. A lot of problems of different dimensions could be tackled and solved using Fabric. The fact, that the project is fully open-sourced may contribute to this fact. With such a big and versatile community springing from different markets, Fabric has the potential to become one of the leading technologies, regarding transaction management or providing data and information between organizations and companies. But this still has to be proven. Although the mentioned use cases seem to be very promising and almost appear revolutionary, a final statement is still far away regarding use cases in such scenarios. All of the mentioned use cases were conducted by IBM itself in early 2017. This also might have been just a marketing tool to get the attention of big companies. If and how Fabric will perform in everyday scenarios over a longer course of time is still an open question, which yet has to be answered. This is also because of the fact, that after further research the mentioned use cases cannot be easily tracked or examined; therefore, a final and factual review or conclusion cannot be done yet.

8.5 Transaction flow and its privacy/security implications

This section discusses the transaction flow through a Hyperledger Fabric network as described by the original paper [39].

8.5.1 Data Processing through Hyperledger

One of the core innovations the Hyperledger Fabric project introduced is the way it processes transactions. Its design is addressing the scalability problem which is one of the biggest hurdles to make blockchain technology enterprise ready.

All previous blockchain networks follow an order-execute paradigm where transactions are first ordered and then executed on all peers in the agreed upon sequence. Let's take a basic example of an Ether payment transactions on the Ethereum network. The network goes through the following steps to process transactions and establish consensus on the ordering and validity. (1) All validating/mining peers are collecting transactions they hear about, validate, order and assemble them into blocks. (2) Peers try to solve the proof of work (POW) crypto puzzle, meaning they try to find an input nonce which's digest will fall into a certain target space when concatenated and hashed with the previous block hash. (3) If a peer finds a solution before receiving any other solution from another peer first, it broadcasts the assembled block with the solution nonce. (4) Every peer receiving the block validates the solution nonce and all the transactions contained within the block. With this approach, each peer effectively repeats what the lucky, validating node already did. Namely, they walk the chain to check if transactions are valid which keeps getting more and more computationally expensive as the blockchain grows and they execute the transactions in sequential order. Obviously, this creates a bottleneck on throughput as it is limited by the execution latency of individual peers.

Hyperledger introduces a new execute-order-validate architecture which takes a more modular approach by splitting the transaction flow into three different steps: (1) Executing and checking the correctness of transactions (2) Establishing consensus on the ordering of transactions (3) Transaction validation. With this design, transactions are executed before the network settles on a final ordering. The inherent advantage in this is that only a subset of peers have to execute (endorse) transactions, allowing for parallel execution and thereby increasing scalability. Hyperledger Fabric provides flexible endorsement policies that define which peers and how many of them have to endorse certain transactions. To get this design to work, Hyperledger Fabric hosts the following modular components:

- **Ordering Service:** It broadcasts resulting state updates from transaction proposals to all peers and establishes consensus on the order of the transactions.
- **Membership Service Provider:** It is responsible for associating peers with cryptographic identities. It maintains the permissioned nature of Fabric and acts as its gatekeeper.
- **Peer-to-Peer Gossip Service(optional):** It disseminates the blocks between peers directly rather than through the ordering service.
- **Peers:** Each peer locally maintains the ledger state and can propose and validate transactions.

An actual application on Hyperledger Fabric consists of the following two components:

- **Smart contracts** also called Chaincodes which implement the application logic and run during the execution phase.

- Endorsement policies which are evaluated during the validation phase and specify how many and which peers on the network should vouch for certain transactions.

8.5.1.1 Phases to create a valid transaction

A Hyperledger Fabric transaction runs through three main phases to persist data on the ledger.

Execution Phase

Initially, a transaction is born as a proposal on the client side. This transaction proposal is then sent to endorsing peers as specified in the endorsement policy which execute the transaction. Executing a transaction and "endorsing it" means that an endorsing peer simulates it against its local state of the blockchain without synchronizing with other peers or recording the output. Instead, endorsers create a read/write-set representing the proposals input state and the proposals output state which is signed and sent back to the proposing client. As soon as the client collects all required endorsements specified by the policy and the read/write set align, the transaction is officially endorsed and can be sent off to the ordering service.

Ordering Phase

The client sends the transaction with the previously received endorsements and its payload (relevant parameters to invoke the according chaincode) to the ordering service. There, the transactions are organized in sequential order and packaged in blocks chained together with hash pointers. Block-creation is a technique to improve the performance of the broadcast protocol which distributes the established ordering among all network peers. Hashing blocks together in a chain further simplifies the verification of the integrity of state evolution over time. By completely isolating the ordering service from execution and validation, Hyperledger Fabric makes consensus as modular as possible allowing for different protocols to be implemented on the ordering service according to the different needs of the network. At this point of the transaction, the network created an endorsed transaction which conforms to the endorsement policy with expected results and it was arranged in a sequential ordering among other transactions. The main problem of double spending or more generally speaking, transaction duplication which blockchain is solving in a novel decentralized manner still remains at this stage.

Validation Phase

The validation phase consists of three subphases.

1. Endorsement Policy Evaluation

Peers receiving blocks from the ordering service will first check if the contained transactions fulfill the requirements of their endorsement policy. If the required endorsers signed and their simulation outputs match, the transaction is marked as valid. If that is not the case the transaction is marked invalid and its effects are disregarded. The validation process is handled by a static library called Validation System Chaincode (VSCC).

2. Read/Write conflict check

To ensure proper evolution of all involved states in transactions, the read-write set of all transactions are compared to the current state of the blockchain on each peer. If the read set is the same as on the current state, meaning there have not been any other intermediary changes to the states involved, the transaction is marked valid. But as mentioned earlier, broadcasted blocks from the ordering service may contain duplicate valid transactions trying to modify the same state. This problem is also known as the double spending problem in the context of a coin transfer transaction where the same coin is sent twice (or multiple times) to different receivers. All transaction proposals may be valid in the eyes of the system but only one of them is allowed to go through to keep the integrity of the blockchain intact. Hyperledger Fabric uses MVCC (Multiversion Concurrency Control) to avoid this issue. In simple terms, it means that the read-write

sets are versioned such that any state change triggers a version update on the read-write set in which it is involved. Any following transaction which would concern the same state that has already been updated would thus be rendered invalid because they don't refer to the same version anymore. So which proposal will finally go through? It's basically first come first serve. Whichever transaction was listed first by the ordering service gets to execute its payload and any following transactions on the same states would have to be re-proposed with a new updated read-write set.

3. Ledger update

Finally, all peers update their version of the blockchain with the results of the previous steps. Compared to other blockchain systems also invalid transaction proposals are stored which offers some interesting back-tracing capabilities for audits and further introduces certain incentives to behave well. Fear of punishment can be a strong enforcement mechanism to keep network participants in line. Failed attempts to cheat on other blockchain systems remain invisible and carry less risk.

8.5.2 Privacy

Recent news on data breaches highlights the importance of data privacy. Blockchain Systems are under the same scrutiny to protect user data and conform to regulations. Public blockchains such as Bitcoin [12] and Ethereum [31] provide user anonymity by hiding their identity behind pseudonyms or so-called addresses while having all transaction data public. For many uses cases, this does not conform to regulations that require involved participants to be known. Additionally, participants in a blockchain system may wish to keep some sensitive transaction information private. This section discusses different mechanisms offered by Hyperledger Fabric to enable privacy.

8.5.2.1 Private Transactions (Channels)

Similar to messaging applications, Hyperledger Fabric offers users the ability to create channels in which only selected users can communicate and transact with each other [32]. In WhatsApp, we would call it a group, where an administrator creates a virtual communication area with selected group members. Administrator rights come with certain predefined privileges such as the right to invite others and can be shared or changed. On Fabric, a channel represents a subnet of communication between member organizations and defines its separate blockchain to store shared information. Beyond a channels' users, it is defined by its deployed chaincodes (smart contracts), the responsible ordering service and its channel policy which offers flexibility on defining corresponding administration rights. Each transaction on a fabric network ultimately takes place on a channel. A channel may include all network participants to imitate a gossip protocol, just two or any number in between. Participants can also be part of many different channels which offers sheer endless combinations of channels in a network setup.

Creating a new channel requires a client request which calls the configuration system's chaincode with the specified setup properties. This in turn creates a genesis block (first block) for the channel ledger holding all configuration information about the channel policy and its members.

When a member is joining an existing channel it receives the channel's instantiating genesis block or an updated reconfiguration block.

So from a high-level privacy perspective, channels are useful when a number of organizations within a network have high transaction volumes between each other and don't require dependencies from other organization to process them. For example, bilateral agreements between a big dairy products supplier and a retailer can be kept private from

the rest of the supplier/retailer network to avoid giving away specific special deals and conditions.

Privacy in fabric networks can further be customized with private data (collections).

8.5.2.2 Private Data withing Channels (Collections)

Up until release 1.2 of Hyperledger Fabric, if a group of organizations within a channel wanted to keep some details private from the rest of the network they had to create a new channel to exchange that information. This implied additional administrative work and did not allow the whole network to see a transaction which was based on hidden private data. For example, on a banking network, participating banks are required to share information about their solvency with regulators. The exact numbers should not be published to all participants but the transaction that verifies that a certain bank complied with regulators should be visible. That is why private data within channels has been introduced [33].

Private data collections within channels feature two components:

- The actual private data: This data is stored on a private database on the peer. The ordering service is not involved and the data is only shared with authorized organizations.
- A hash of that data: The hash is stored on all peers of a channel and functions as a proof of state validity.

While private data transactions hide the actual involved data from unauthorized parties, they do not prevent other network participants from seeing when this private data is being modified. This is because the hashes of the private data, and therefore also the changes to them, are still publicly visible on the ledger.

Private transactions also expose all parties who are allowed to access the private data. This information is available and needed on the ledger to properly distribute private data to the authorized participants.

Transaction Flow with private Data

The transaction flow slightly changes when private data is referenced in transactions.

1. A proposal is created on the client side and sent to the endorsing peers listed as authorized organizations for the collection of private data involved.
2. Endorsers simulate the transaction against their current view of the ledger and distribute the private data to all authorized peers.
3. The endorsers send back their endorsements to the client. A response only contains public data meaning the hashes of the involved private key and value and all invoked dependencies. There is no private data send back again.
4. The fourth step remains the same as without private data. If the client received enough endorsements and from the right entities as specified by the endorsement policy, it sends it to the ordering service which packages the transaction including the hashes of the involved private data into a block. After block distribution all peers on a channel can verify transactions including private data based on their hashes.
5. At last, peers check the collection policy to see if they are entitled to have insight into the actual private data. If yes, they will check if they already received it from an endorsing peer, otherwise they will request it from another authorized peer. Once the data is available they will validate the data against the hashes and store the

transaction and the block. Once committed, the private data is persisted on the peers' private database.

8.5.3 Trust

Fabric's architecture is designed to feature different models of trust [35]. It recognizes that there is no one size fits all consensus protocol for blockchain networks. The crucial point of trust in every Hyperledger Fabric network is the ordering service which can implement different consensus protocols such as Paxos, Raft and more. Beyond trust in a consistent ordering service, each Fabric network defines its own trust assumptions on the application level with its network, channel, and endorsement policies. This section discusses the different ordering service setups and the available policies to tailor a Fabric network to different use cases.

8.5.3.1 The Ordering Service: Trust in a centralized service instead of math and rationality?

In public blockchains, every node can participate in finding consensus by collecting transactions and proposing blocks. Selection of blocks based on lottery-type consensus algorithms such as POW are probabilistic and introduce some level of uncertainty to those systems. When two nodes find a new block at the same time, the blockchain may diverge into two different directions (forks) and creates different views on the current state for different participants. In Hyperledger Fabric, block creation is done by the ordering service and is deterministic. Due to the provided finality, one could argue that a permissioned blockchain network on Hyperledger Fabric is more secure and should carry more trust. But what are the implications of different setups of the ordering service? Who is providing it? What are the incentives to run it correctly?

The ordering service may implement Solo, Raft or Kafka out of the box to establish consensus all of which have different pros and cons. No matter which protocol, consensus should satisfy the safety and liveness property:

Safety means that each node is guaranteed the same sequence of inputs and results in the same output on each node. Meaning nodes that receive an identical series of transactions should get the same state changes. The algorithm must ensure that it works identically to a single node system that executes each transaction atomically and in sequential order. Liveness means that each properly working node will receive every submitted transaction sooner or later. The system is required to make progress.

To guarantee those properties the service has to be fault tolerant to a certain level, meaning it must be able to continue working in the presence of errors or breakdowns. There are two main types of failures and thereof two types of fault tolerance classifications.

- **Crash Fault Tolerance (CFT):** A system is said to be fault tolerant if it can handle a certain number of crashes (meaning services/nodes simply shut off or are not reachable). In the context of the ordering service in Fabric, more than half of the nodes running it have to be available to establish consensus. In general, it implies that there have to be $2f + 1$ (f is the number of failures) nodes available to be crash fault tolerant.
- **Byzantine Fault Tolerance (BFT):** A system is said to be byzantine fault tolerant if it can handle arbitrary behavior of nodes. The nodes could lie or collude due to intentional attacks or simply due to software failures. Being byzantine fault-tolerant requires a minimum of $3f + 1$ (f number of failures) nodes to function properly.

Hyperledger Fabric currently only offers crash fault tolerant ordering service implementations but many BFT versions are being studied and developed.

Solo

The name says it all. The solo implementation only features a single node and is thus never fault tolerant. Solo is obviously not a valid option for production due to the centralization of power and the lack of CFT. However, it is the go-to implementation for testing and evaluating proof of concepts.

Raft

Raft [34] is a crash fault tolerant protocol that follows a leader-follower model and is the standard option offered out of the box by Hyperledger Fabric. A leader is elected among the ordering nodes in a random fashion which then replicates messages to the other nodes. Once elected, the leader node may also fail as long as a majority of ordering nodes remain available. In such a case, a new election term is started after a timer expires without getting any response from the current leader. It is obvious that this design is not able to handle arbitrary behavior of nodes. A malicious node could simply replicate wrong data once it is elected as the leader which will happen sooner or later due to the randomness of the election and the limited number of participating ordering nodes.

Kafka

Next to Raft, Fabric also offers Kafka [36] as a pluggable consensus protocol. It is similar to Raft and follows a leader-follower pattern as well which is limited to crash fault tolerance. If trust does not lie in the consensus protocol on the ordering service how is it established?

8.5.3.2 Relying on sensible endorsement policies and network configurations

When bootstrapping a network, an ordering service is started by an administrator of some organization. This initiating organization defines the network configurations which governs administrative capabilities for the ordering service and the network in general. The following components have to be defined and evolved over time in any Hyperledger Fabric network.

- Network configuration (NC) settings: The NC defines the administrative rights of participating organizations, its recognized certificate authority to identify those participants and configures the ordering service(s). It is important to note that the ordering service has insight into all transaction proposals (except private data collections) it receives.
- Channel configuration (CC) settings: The CC defines who the channel participants are and what administrative rights they have. The CC also sets the ordering service to be used.
- Chaincode endorsement policies: Endorsement policies define who and how transaction proposals should be validated. It makes sense to only require a subset of the specified endorsers' endorsements to let clients send the proposal to the ordering service. This allows for some crash fault tolerance and also avoids a single endorser to deny service. If all endorsements are needed for a proposal to be valid then a single endorser could simply block some proposals (maybe from a competitor in the network) from being processed further.

The utility and security of an individual network is heavily dependent on sensible configurations of those components. The flexibility offered by Fabric allows to tailor a network to many different use cases but requires deep knowledge about the privacy, security and performance implications of each setting.

8.5.3.3 Incentives in a permissioned blockchain?

Due to a missing native cryptocurrency and a mining scheme that rewards nodes for contributing to finding consensus there are no clear cut incentives to take up the role of the

orderer or endorser. Those are jobs that are absolutely vital for any Fabric network and have to be performed by someone. They require administrative overhead and computational work while not giving any additional benefits. Thus deciding on who should be an ordering and/or endorsing peer is difficult. Should it be the biggest organization? The organization with the "best reputation"? The organization with the biggest investment in the project? The network bootstrapping process requires answers to those questions which many projects simply cannot give from the get-go. Proper incentive mechanisms seem to be one of the biggest drawbacks for Hyperledger Fabric compared to public blockchains. Apart from missing rewards schemes in the network, Fabric has other interesting properties that incentivize good behavior. Since network participants are well known and invalid proposals are also persisted, fear of punishment becomes a governing factor. Misbehavior as a client, orderer or endorser can be detected and traced back to a known participant. It is then up to the network participants on how to punish violations. So the bottom line is that a Fabric network is governed by fear rather than rewards. It will be interesting to see if Fabric manages to introduce incentive mechanisms beyond the fear of misbehavior.

8.6 Performance Evaluation

Traditional database systems that are used for transactions and are run by a central, controlling authority achieve great performances. For example Visa's global payment system VisaNet can process up to 65000 transactions per second (tps) [37]. Blockchains, on the other hand, are struggling to attain such throughput. Ethereum, for example, has approximately 27 tps when we assume that 380 Ether transactions fit onto a block and the block time is 14 seconds. Blockchains, like Ethereum, which are based on prove-of-work are restricted in their speed because of their consensus mechanism. The required work for block creation needs to be hard and therefore time-consuming to make the blockchain secure. It prevents malicious network participants from changing the blockchain at their will as long as they do not reach more than 50% of the network's computing power [12]. But the consensus mechanism is not the only bottleneck. Even if Ethereum would change its consensus mechanism to prove-of-stake, all full nodes still need to execute and validate all transactions and smart contracts [38]. The execution furthermore happens in sequential order, because transactions could depend on each other. These things limit the network's scalability.

Fabric promises relieve from such scalability restrictions. First of all, it doesn't rely on proof-of-work consensus protocols to reach consensus but applies less resource-intensive crash-tolerant or BFT protocols. Another performance advantage is Fabric's execute-order-validate architecture. It allows multiple nodes to execute smart contracts in parallel because not all nodes have to execute all contracts [39]. Only the peers that are necessary for the endorsement of a specific transaction need to execute that transaction. After ordering and creating a block only the read and write sets have to be validated by each node (peer). They don't need to run the chain code again.

A few papers have been published that evaluate Fabric's performance. The ones that we deem the most useful and constructive are from [39], [41] and [43]. And so, they are the source of the performance evaluation below. In all three papers, the standard test setup consists of a few virtual machines in the same data center with 1 to 3 Gbps links between them. The VMs are equipped with up to 32 virtual CPUs and enough RAM for all the tasks (i.e. RAM was never a performance bottleneck). The number of peers is from 5 to 8. The specific setups for experiments that deviate from this standard are mentioned as they come up.

8.6.1 Performance Metrics

Before we can evaluate Fabric we need to define metrics by which the performance of blockchains can be measured and compared. Two intuitive and common metrics are the **throughput** and **latency**. Both can be defined on multiple levels, depending on the system component under examination (e.g. the endorsement phase). We define end-to-end throughput as the rate at which transactions are committed to the ledger². And end-to-end latency is defined as the time elapsed from the point a client sends a transaction request to the point the transaction is committed to the ledger. More specific definitions apply to the individual components of Fabric. If, for example, the ordering service is of interest, then the throughput is defined as the rate at which the orderers reach consensus on the transaction's ordering, i.e. produce new blocks. Or if we take a closer look at the endorsement phase, then the latency is the time needed for a client to collect all the required endorsements for his transaction. Other metrics built upon throughput and latency. E.g. **scalability** can be defined as the change in throughput and latency when increasing the number of network participants and the workloads. And fault tolerance is measured by how latency and throughput are affected by node failures [40].

8.6.2 Performance Impacts

The following Sections describe how different parameters and software parts of Fabric influence the performance.

8.6.2.1 Performance impact of the block size

As in other blockchains, the block size does influence Fabric's performance as well. As long as the transaction arrival rate does not exceed the maximal throughput of the system, meaning the system is not saturated, the block size determines the overall latency. So, given a manageable workload, a larger block sizes lead to higher latency. This is intuitive, because with a bigger block size it takes longer to fill the block than with a small block size at the same transaction rate. A transaction has to wait longer at the ordering service until it gets included into a block. As an example, the latency doubles when changing the block size from 50 to 100 [41]. But if the workload is higher and the saturation point of the network is reached, a larger block size leads to more throughput and less latency. The bottleneck in this scenario is not the time a transaction has to wait at the ordering service but the validation at the peers. The peers are faster in validating and committing a block of size n than m blocks of size n/m .

We can conclude that that the block size should be kept small as long as the network is not saturated. Or instead of setting a small block size, one could also set a short block timeout which also leads to quicker block creation in times of small workloads. In the opposite case of high workloads, some performance gains can be obtained by increasing the block size.

8.6.2.2 Performance impact of endorsement validation

In Fabric v1.0, a noticeable bottleneck can be found at the process in which peers validate the endorsements of transactions before storing them on the ledger. When increasing the transaction arrival rate Fabric's throughput at first reacts linearly until it reaches a saturation point, which is to be expected. At that point throughput flattens and latency increases. A closer look at what causes the latency reveals that the sequential endorsement validation is indeed the bottleneck. There is no increase in the latency in the endorsement

²defined as such in [45] and [41]

phase or in broadcasting the endorsed transaction to the ordering service. The serial manner in which the validation is implemented in Fabric v1.0 leads to the underutilization of the available CPUs. Only one CPU is used for the validation, while the peers have many more CPUs available.

By providing a parallel implementation of the endorsement validation the authors in [41] achieve an increase in throughput by a factor of six. This improvement was incorporated in Fabric v1.1.0³. Since that version, the overall throughput increases and the latency of endorsement validation on the peers decreases when more CPUs are added. The new bottleneck in the peers is the sequential checking of read and write sets and the ledger writes after the endorsement validation [39].

8.6.2.3 Performance impact of endorsement policies

Transaction endorsement policy syntax consists of `AND`, `OR` and `N-OutOf`. Fabric's performance depends on the value of these policies. This is apparent because if there are more endorsers required by a policy, then the peers have to check more endorsement signatures. Though, collecting the endorsements before is not affected by the policy since the endorsing peers are working in parallel. With an increasing number of endorsers, the byte size of a transaction increases too, because the transaction contains more endorser certificates and endorsement signatures. When validating a policy, the peer has to deserialize the certificates, validate the identities with a Membership service provider and verify the signatures on the transaction. For smaller transaction size (i.e. less network traffic) and less latency in the endorsement validation, it is desirable to have as few endorsers in a policy as possible. To further reduce the compute time for endorsement validation, the deserialized identities should be cached at the peers along with the corresponding information from the MSPs. The security risks of such caching are small if the certificate revocation lists from the MSPs are consulted.

Another negative performance impact of endorsement policies is the usage of sub-policies. An example for a policy with sub-policies is `OR [AND(a ,b), AND(a, c), AND(a, d), AND(b, c), AND(b, d), AND(c, d)]`. This defines that at least two endorsements of the organizations a, b, c, and d are required. In tests containing only transactions with such a policy, the overall throughput decreased by 7% compared to policies without sub-policies. For the policy `OR[AND(a ,b, c), AND(a, b, d), AND(b, c, d), AND(a, c, d)]` the it even decreased by 20%. Again the problem lies in the endorsement validation latency. Interesting is the observation that these two policies are more resource-intensive than the semantically equal policies `2-OutOf(a,b,c,d)` and `3-OutOf(a,b,c,d)`. For better performance it is desirable to have as few sub-policies as possible.

8.6.2.4 Performance impact of channels

The number of channels had a strong impact on the throughput and latency before the endorsement validation was parallelized. In experiments with Fabric v1.0 it was shown that with an increasing number of channels the CPU usage on the peers increased, thereby also increasing the throughput and decreasing the overall latency [41]. The increase in throughput when going from 1 to 16 channels was by a factor of six. This phenomenon is connected to the endorsement validation on the peers that is done sequentially in v1.0. In that case, more channels were actually a remedy for that problem because spreading validation over multiple channels automatically leads to parallelization and thereby better utilization of the computing resources.

With the parallelization of the endorsement validation in Fabric v1.1.0, this dependence of the performance on the channel number was removed. We know of no newer experiments

³see <https://jira.hyperledger.org/browse/FAB-5932>

that show a dependence of the performance on channel number. But speculatively one could imagine that the validation of the read and write sets and the committing to the ledger perform better with more channels because those operations are still implemented sequentially.

8.6.2.5 Performance impact of resource allocation

Fabric's performance greatly improves when adding more CPU power to the peers. This again is true since parallel endorsement validation has been implemented in version 1.1.0. When increasing the number of CPUs from 2 to 32 the latency in endorsement validation reduces in each step and the throughput increases linearly. From 4 to 16 CPUs the throughput doubles from about 1500 to 3000 tps [39]. This shows the direct gain of parallelizing the endorsement validation. In the specific experiments done in [39], the increase in throughput flattens after 16 CPUs because the endorsement validation is not the bottleneck anymore. The latency of the endorsement validation falls beneath the ones of the read/write set validation and committing to the ledger. Those two operations are less computational heavy than the endorsement validation but are still done sequentially. Their latency is the same independent of the number of CPUs. To get higher throughput in the future these two operations would also have to be optimized.

An interesting point of resource allocation is that even if most peers have a high CPU allocation the performance can still suffer if one peer has few resources. For this, imagine a peer *A* with 2 CPUs and an incoming workload that saturates its resources because of the work it does in the validation and committing phase. Transactions that depend on the endorsement of peer *A* will experience higher latencies in the endorsement phase because of the resource exhaustion on *A*. Endorsement requests might be delayed or they might even time out. Even if the other endorsing peers have not reached their saturation point yet, the latency is decided by *A*. Additionally because of the lower rate at which the ledger of a weak peer is updated, compared to fast peers, more conflicts in the read/write set checks can appear. So if an endorsement request is sent to peer *A* and to a fast peer, it is possible that the execution outcomes of the two peers are different because their current ledger state is different. The client is unable to obtain a valid set of endorsements and will have to retry.

8.6.2.6 Performance impact of the state database

It seems obvious that the performance of the database system used to store the ledger state to some extent dictates the performance of a peer in the network. Fabric offers the choice between CouchDB and LevelDB as the ledger's state database on the peers, while LevelDB is the default [42]. LevelDB does not function in a client-server fashion but is instead running in the same process with the peer application. CouchDB, on the other hand, runs in its own process and is accessed by the peer via HTTP requests. This difference also leads to a difference in performance. In [41] the maximum throughput that could be achieved on a single channel was 3 times higher with LevelDB than with CouchDB. When using CouchDB the latencies for endorsement validation, validation of read/write sets and updating the ledger were higher because HTTP requests to the state database have to be made for every transaction which incurs a bigger overhead than with LevelDB.

Another problem with CouchDB was found when increasing the number of writes in the transactions, i.e. changes to key-value pairs on the ledger. Increasing the number of writes increased the latencies in the endorsement phase and in committing to the ledger state database. The explanation for this is as follows. For updating one key-value pair in the state database, a peer has to first issue a GET request, for fetching the previous version,

and then issue a PUT request, for actually updating the key-value pair. So, obviously, the latency increases when more writes are needed in comparison to reads. But probably more important is that the writes to the state database require an exclusive write lock on the whole database. And so, the more write operations are performed when committing to the ledger, the more race conditions will appear when the peer needs to acquire a shared lock for reading from the ledger in the endorsement phase. This increases latency.

We can conclude that using LevelDB is the more performant option as the ledger state database. But for some applications CouchDB might be more desirable because of a richer query API. To improve the performance with CouchDB its bulk read and write functionality should be utilized to reduce the number of HTTP requests to the database and thereby reducing the duration of locks.

In [43] the improvements for the data storage go even further. First, the disk-based database (LevelDB or CouchDB) is exchanged with an in-memory hash table for storing the key-value pairs of the ledger state. This assumes that a peer can keep all key-value pairs from the ledger state in memory. It also accepts that the state is lost when the peer crashes, although it can be rebuilt by going through the whole history of the ledger on startup. Second, peers are split into two components of which one only does the endorsement and the other the validation and committing to the ledger. The peer that commits to the ledger also sends updates to the endorsement peer. The endorsement peer can thereby retain an up-to-date ledger state without having to deal with transaction validation and committing changes. Third, the actual storage of the blockchain (not the ledger state) is separated from the peers and put into an own distributed storage cluster. All of these changes improve Fabric's performance but are not part of a Fabric release at the time of writing.

8.6.2.7 Performance Impact of the ordering service

The orderers in Fabric 1.0 receive endorsed transactions from clients and forward the whole transaction to the consensus mechanism, e.g. to the Kafka cluster, in case Kafka is used. This leads to significant communication overhead for transactions of large size (several kilobytes). Instead, the orderer that receives the transaction from a client could only forward the transaction ID to the consensus mechanism since the ID is enough for generating an order between the transactions [43]. The orderer that received the transaction stores the transaction's body and reassembles it with the ID after ordering. This is intuitive when only one orderer serves all peers, but in the case of multiple orderers, it is not clear how the other orderers will get to know the transaction body. A peer can request new blocks from any of the orderers, therefore every orderer needs the full transaction information. Nevertheless, by only sending the transaction ID, the throughput can be increased. For example, for transactions with 4096 bytes in size, it can almost be tripled [43].

The above result was obtained on a Kafka cluster which only provides crash tolerance. There are no experiments with a Byzantine fault tolerant protocol, although there exists an integration of BFT-SMaRt [47] for Fabric. Additionally, the above experiments were conducted with only one ordering node that used a small Kafka cluster of three nodes. It is questionable if the throughput of the consensus phase can be maintained when adding more orderers.

8.6.3 Performance Conclusion

The performance impacts discussed in Section 8.6.2.1 to 8.6.2.7 are being addressed in Hyperledger Fabric Jira issues⁴ of which some have already been implemented and others are still open and planned for Fabric version 2 releases. With the optimizations from [43] which were built upon Fabric v1.2, the authors achieved a maximum throughput of about 20000 transactions per second. This result was obtained on a LAN with a single orderer using a Kafka cluster, one committing peer connected to the orderer and 5 endorsing peers connected to the committing peer. The separation of peer responsibilities is one of the optimizations presented in Section 8.6.2.6. The endorsing peers synchronize their ledger state database with the committing peer and the committing peer writes the ledger updates to external storage. This performance is approaching Visa's VisaNet stress test performance of 65000 tps and could probably already handle normal loads on VisaNet [37]. Without the optimizations of [43], Fabric's throughput can be expected to be around 3500 tps in a LAN setup and 2500 tps in a WAN setup [39]. It is clear that Fabric outperforms the permissionless and currently proof-of-work-based blockchain Ethereum [46] but more interesting is a comparison with the competing business blockchains Corda and Quorum. In experiments from 2018, Quorum achieved up to 2100 tps [48] using a crash tolerant consensus mechanism. For Corda, we did not find any scientific paper on performance. In a presentation slide set from 2018, found on the R3 website, it said that Corda's node implementation could reach a throughput of up to 1800 tps [49]. No statement was made for a whole Corda network. In summary, performance-wise Fabric is well positioned within the field of business-focused blockchains.

If the optimizations proposed in [43] will make it into a future Fabric release, another performance evaluation on a WAN and using a Byzantine-fault tolerant (BFT) consensus protocol must be performed to confirm Fabric's leadership position. So far, the test setup in [43] used only a crash-tolerant consensus protocol and a quite small consortium size. It is questionable if the throughput of 20000 tps can be maintained when extending the Fabric network to more peers and orderers. A higher consortium size has been shown to have negative effects on the performance of Fabric v1.0 [45]. Running peers on AWS EC2 instances with 8 vCPUs and 16 GB memory, a consortium with 16 peers can be up to 3 times slower than with 4 peers for high transaction arrival rates (1400 tps). One can expect that exchanging the crash-tolerant consensus protocol with a BFT protocol will not hurt performance. As of version 1.3 Fabric provides a BFT implementation based on BFT-SMaRt [47]. It has been shown that a throughput of 20000 is also achievable with Fabric's BFT-SMaRt implementation [50].

8.7 Summary and Conclusion

Hyperledger Fabric has gained a strong foothold in the domain of enterprise blockchains. Since it has been incorporated into the Linux Foundation in 2016 its developer community has grown to nearly 200 developers from 35 different organizations [51]. It is backed by many member organizations and associate organizations from different industries. Among these are big names like IBM, Intel, and Cisco, which have an interest in being part of a successful blockchain platform but also in providing convenient blockchain services to their customers. So regarding the community and ecosystem, Fabric seems to have a good stance, which is an important part of a successful platform. Even the R3 consortium, that develops their own blockchain solution Corda, is part of the project. And even though many corporate organizations have an interest in the project, it is in the hands of the Linux

⁴Jira Epics <https://jira.hyperledger.org/browse/FAB-6421> and <https://jira.hyperledger.org/browse/FAB-14513>

Foundation which embeds the project into an active open-source community. From the three competitors Quorum, Corda and Fabric, Fabric is also the one that has received the most academic attention. Multiple papers are available evaluating Fabric and proposing improvements in the areas of performance, privacy, and consensus.

Fabric differentiates itself from other blockchains most prominently with its execute-order-validate architecture. Most other blockchains are based on the order-execute architecture, in which transactions are first ordered, put into a block, disseminated through the network and then executed and validated on all the participating nodes. In Fabric's architecture, a transaction submitted by a client is first executed on the endorsing peers and only then is it ordered by the ordering service and disseminated to all nodes for validation. This has several advantages. For example, performance is improved because not every node in the network has to execute every transaction, only the required endorsing peers do. More interestingly, it allows for non-deterministic smart contract code. Because a client has to first obtain transaction endorsements with equal results before sending a transaction to the ordering service, possibly non-deterministic code is not a thread to the ledger consistency. Only the client's request is hindered in getting executed on the blockchain. Most other blockchains only allow deterministic smart contract code because of the order-execute architecture. This gives rise to the need for blockchain-specific programming languages or restricted subsets of common high-level programming languages. With Fabric the developer is free to use languages like Java, Python or Go without restrictions. But of course, now the developer is responsible for the determinism of his code. If he is not careful, his smart contract might lead to different results on different endorsers and never gets committed.

Another advantage of Fabric's execute-order-validate architecture is that it protects against denial of service attacks with endless loops or computational heavy operations in smart contracts. Endorsing peers can set up policies that define at which point a smart contract is terminated if it consumes too many resources. This removes the need for a currency-based execution cost for blockchain operations, as it is seen for example with GAS in Ethereum.

Performance-wise, Hyperledger Fabric is ahead of its competitors in the enterprise blockchain domain. Since the first releases, performance has improved a lot and a few papers have been released, evaluating throughput and latency. New optimizations are still being developed and current academic work promises that Fabric can scale up to 20000 transactions per second.

In the area of privacy, Fabric currently offers the concept of channels and private data. While channels simply instantiate a separate blockchain between a subset of the network's peers, private data is more complex. With private data, you can specify in detail which parties can see what data of a certain transaction. But that does not yet provide anonymity for the transacting parties. Anyone in the same channel can tell who is exchanging data. For this purpose, Idemix was introduced, which allows clients to use zero-knowledge proofs to hide their real identities when making transactions. In the future, even private transfers of assets will be enabled, which will hide the transacting parties and the transacted asset amount's, while still enabling auditability.

In summary, the future of Fabric looks bright. It is already being used in real-world use cases, and big companies like IBM or Oracle offer Fabric-based blockchain services for business customers. It has a good chance to actually survive the blockchain hype and establish itself as another useful software tool in the IT landscape.

Bibliography

- [1] Hyperledger Wikipedia article, <https://en.wikipedia.org/wiki/Hyperledger>, Last visited May 2019.
- [2] Linux Foundation Website, Membership Page, <https://www.linuxfoundation.org/membership/>, Last visited April 2019.
- [3] Linux Foundation Website, Cloud Native Computing Foundation Receives 9 Million Cloud Credit Grant from Google Cloud to Fund Kubernetes Development, Empower Community, <https://www.linuxfoundation.org/press-release/2018/08/cncf-receives-9-million-from-google-to-fund-kubernetes/>, Last visited May 2019.
- [4] ACME Wikipedia article, https://en.wikipedia.org/wiki/Automated_Certificate_Management_Environment, Last visited May 2019.
- [5] D. Tapscott and A. Tapscott, "Blockchain Revolution", Portfolio Penguin, 2016.
- [6] Hyperledger Introduction (slide 2), www.hyperledger.org/wp-content/uploads/2018/02/Hyperledger-Overview_February-2018-2.pdf, Last visited April 2019.
- [7] Let's Encrypt website, <https://letsencrypt.org/>, Last visited April 2019.
- [8] Censys Website, HTTPS Certificate Market Share Report, https://censys.io/certificates/report?q=tags\%3Atrusted&field=parsed.issuer.organization.raw&max_buckets=50, Last visited May 2019.
- [9] Kubernetes Website, What is Kubernetes, <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>, Last visited May 2019.
- [10] Automotive Linux Website, Members Page, <https://www.automotivelinux.org/about/members>, Last visited May 2019.
- [11] Prnewswire Website, Cloud Computing Market Report/Predictions, <https://www.prnewswire.com/news-releases/the-global-cloud-computing-market-size-is-expected-to-grow-from-usd-272-0-billion-in-2018-to-usd-623-3-billion-by-2023--at-a-compound-annual-growth-rate-cagr-of-18-0-300806908.html>, Last visited May 2019.
- [12] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2018, <https://bitcoin.org/bitcoin.pdf>
- [13] Hyperledger Website, <https://www.hyperledger.org/>, Last visited April 2019.
- [14] Hyperledger Docs Website, Introduction <https://hyperledger-fabric.readthedocs.io/en/release-1.4/whatis.html>, Last visited April 2019.

- [15] The Linux Foundation Website, Linux Foundations Founding Members, <https://www.linuxfoundation.org/press-release/2016/02/linux-foundations-hyperledger-project-announces-30-founding-members-and-code-proposals-to-advance-blockchain-technology/>, Last visited May 2019.
- [16] Tendermint Website, <https://tendermint.com/docs/introduction/introduction.html#what-is-tendermint>, Last visited May 2019.
- [17] Hyperledger Wiki Website - Nodes, <https://hyperledger-fabric.readthedocs.io/en/release-1.4/arch-deep-dive.html#nodes>, Last visited April 2019.
- [18] Hyperledger Wiki Website - Model, https://hyperledger-fabric.readthedocs.io/en/release-1.4/fabric_model.html#hyperledger-fabric-model, Last visited April 2019.
- [19] Hyperledger Wiki Website - Smart Contracts, https://hyperledger-fabric.readthedocs.io/en/release-1.4/_images/smartcontract.diagram.02.png, Last visited April 2019.
- [20] Introduction to the topic, <https://hyperledger-fabric.readthedocs.io/en/release-1.4/whatis.html>, Last visited May 2019
- [21] Manuals on blockchain networks, <https://cloud.ibm.com/docs/services/blockchain/reference?topic=blockchain-hyperledger-fabric>, Last visited May 2019
- [22] Information on Chaincode, <https://hyperledger-fabric.readthedocs.io/en/release-1.4/whatis.html#smart-contracts>, Last visited May 2019
- [23] Manuals on Hyperledger, <https://hyperledger-fabric.readthedocs.io/en/>, Last visited April 2019
- [24] IBM Blockchain Platform, <https://cloud.ibm.com/docs/services/blockchain/howto?topic=blockchain-ibp-console-organizations>, Last visited April 2019.
- [25] Different Real Use Cases, <https://openledger.info/insights/hyperledger-enterprise-solutions-top-5-real-use-cases/>, Last visited May 2019.
- [26] Hichuan Use case, <https://www.ibtimes.co.uk/ibm-sichuan-hejia-building-large-scale-chinese-pharmaceuticals-blockchain-1616418>, Last visited May 2019.
- [27] Educational Use case, <https://www-03.ibm.com/press/us/en/pressrelease/52160.wss>, Last visited May 2019.
- [28] Report one the market situation, <https://www.finanzen.ch/nachrichten/aktien/global-blockchain-market-in-retail-sector-2019-2023-booming-e-commerce-industry-need-for-it-scalability-to-adopt-blockchain-technology-competitive-landscape-1028164079>, Last visited May 2019.
- [29] Smart Energy Management Use Case, <https://www-03.ibm.com/press/us/en/pressrelease/52243.wss>, Last visited May 2019.
- [30] Pros and Cons regarding Fabric, <https://www.devteam.space/blog/pros-and-cons-of-hyperledger-fabric-for-blockchain-networks/>, Last visited May 2019.

- [31] Ethereum White Paper, <https://github.com/ethereum/wiki/wiki/White-Paper>, Last visited in May 2019.
- [32] Hyperledger Fabric documentation on private data, <https://hyperledger-fabric.readthedocs.io/en/release-1.4/private-data/private-data.html>, Last visited in May 2019.
- [33] Hyperledger Fabric documentation on channels, <https://hyperledger-fabric.readthedocs.io/en/release-1.4/channels.html>, Last visited in May 2019.
- [34] D. Ongaro and J. Ousterhout, "Raft: In Search of an Understandable Consensus Algorithm", 2014, <https://web.stanford.edu/~ouster/cgi-bin/papers/raft-atc14>.
- [35] Hyperledger Architecture Volume 1, https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf, Last visited in May 2019.
- [36] Kafka Documentation, <https://kafka.apache.org/documentation/>, Last visited in May 2019.
- [37] Visa Fact Sheet, <https://usa.visa.com/dam/VCOM/download/corporate/media/visanet-technology/aboutvisafactsheet.pdf>, Last visited April 2019.
- [38] Blockchain and Mining in Ethereum's White Paper, <https://github.com/ethereum/wiki/wiki/White-Paper#blockchain-and-mining>, Last visited in April 2019.
- [39] E. Androulaki, et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains", Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 15 pages, 2018, <http://doi.acm.org/10.1145/3190508.3190538>.
- [40] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH: A Framework for Analyzing Private Blockchains", Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD 17, 2017.
- [41] Thakkar Parth, Nathan Senthil and Vishwanathan Balaji, "Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform", arXiv e-prints, arXiv:1805.11390, May 2018, <https://ui.adsabs.harvard.edu/abs/2018arXiv180511390T>.
- [42] Hyperledger Fabric documentation on the ledger state database, https://hyperledger-fabric.readthedocs.io/en/release-1.4/couchdb_as_state_database.html, Last visited in April 2019.
- [43] C. Gorenflo, S. Lee, L. Golab, S. Keshav, "FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second", arXiv e-prints, January, 2019.
- [44] Qassim Nasir, Ilham A. Qasse, Manar Abu Talib, and Ali Bou Nassif, "Performance Analysis of Hyperledger Fabric Platforms", Security and Communication Networks, vol. 2018, Article ID 3976093, 14 pages, 2018, <https://doi.org/10.1155/2018/3976093>.
- [45] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat and S. Chatterjee, "Performance Characterization of Hyperledger Fabric," 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, 2018, pp. 65-74, <https://doi.org/10.1109/CVCBT.2018.00013>.

- [46] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads", in 26th International Conference on Computer Communications and Networks, 2017.
- [47] A. Bessani, J. Sousa and E. E. P. Alchieri, "State Machine Replication for the Masses with BFT-SMART," 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, GA, 2014, pp. 355-362, <https://ieeexplore.ieee.org/document/6903593>.
- [48] A. Baliga, I. Subhod, P. Kamat and S. Chatterjee, "Performance Evaluation of the Quorum Blockchain Platform", 2018, <https://arxiv.org/abs/1809.03421>.
- [49] J. Carlyle, "Corda Performance To infinity... and beyond!", 2018, <https://www.r3.com/wp-content/uploads/2018/04/Corda-Performance-ENG.pdf>, Last visited in May 2019.
- [50] J. Sousa, A. Bessani and M. Vukolic, "A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform," 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg City, 2018, pp. 51-58, <https://ieeexplore.ieee.org/document/8416470>.
- [51] Hyperledger Wiki Website - Introduction, <https://hyperledger-fabric.readthedocs.io/en/release-1.4/whatis.html>, Last visited in May 2019.

Chapter 9

An Overview of Information Visualization for Data Exploring in Blockchain Universe

Basil Fuchs, Jeremy Kubrak, Tim Grimm, Severin Wullschleger

This paper describes the current status of blockchain visualization. Basic concepts of blockchain technology and visualization techniques are explained in order to foster a deeper understanding of the discussed topic. In order to display data saved in the blockchain, data has to be extracted from it. The difficulties in terms of usability and the status of the chain being in-sync are key challenges in blockchain data extraction. Nevertheless the data extraction may offer additional opportunities such as data integrity for big data projects, real-time analysis of available data (e.g network traffic analysis, transaction fee evolution).

The set of available blockchain visualizations is mostly based on cryptocurrencies. Therefore, several examples for bitcoin and ethereum have been selected to show how blockchain data can be visualized. Furthermore, benefits of visualizing data as well as challenges and future implications are considered and presented.

Contents

9.1	Introduction	241
9.2	Background	241
9.2.1	Blockchain	241
9.2.2	Information Visualization	243
9.2.3	Information Visualization on communication systems	247
9.3	Blockchain Data Extraction	257
9.3.1	Challenges	257
9.3.2	Opportunities	258
9.3.3	Techniques & Examples	258
9.4	Blockchain Visualizations	263
9.4.1	Example 1: Bitcoin transaction flow visualization	263
9.4.2	Example 2: Ethereum transaction activity	267
9.5	Discussion	270
9.5.1	Benefits	270
9.5.2	Challenges and Limitations	272
9.6	Summary and Conclusions	273

9.1 Introduction

This work presents an overview of information visualization for data exploring in blockchain universe including real life applications for cryptocurrencies. Bitcoin and Ethereum dominate the cryptocurrency market and presents researchers with a rich source of real-time transactional data. Some attempts has been made to visualize this Blockchain transaction flow, which will be discussed in this research. Therefore the report gives an insight in different information visualization techniques with extracted blockchain data.

9.2 Background

In this section, we will introduce the main concepts of Blockchain and Information Visualization. Also, we give examples of successful applications of visualizations in the communication systems area as well as examples for different blockchain data extraction applications.

9.2.1 Blockchain

Describing blockchain in one sentence is difficult. There are many who tried to do that with differing results. In the following listing there are two blockchain definitions. In the paragraphs below the more important different aspects of blockchains and its definition are discussed more in detail.

- Condos et al. (2016) define a blockchain as a digital ledger for data sets, events or transactions which are maintained by a distributed network.
- Walport (2015) defines a blockchain as a kind of database where the records are grouped together in blocks. The blocks are chronologically connected with each other by using cryptographic signatures.

In the following subsections different key aspects of blockchain are described. By understanding them, one can also understand why information extraction from blockchains is not a trivial task.

9.2.1.1 Structure: A chain of blocks

The structure of the blockchain is an important aspect. The name itself states it very well: A blockchain is a chain of blocks. The blocks are cryptographically connected with each other. Every block contains the cryptographic hash of its previous block in its block header. This structure is one of the reasons why blockchain technology is so powerful. One little change in a single block changes all hashes of the following blocks and can be immediately recognized. This makes distributed ledgers tamper-proof.

One block contains a so called *Merkle Tree* of the valid transactions: Two transactions are concatenated and hashed until only one hash is left; the merkle root. This makes the validation of a block very fast and efficient.

9.2.1.2 Decentralization

Another key aspect of blockchain and its definition is *Decentralization*. The blockchain itself is saved decentralized which means there is no central entity and therefore no central point of failure. Distributed shared ledger, which is a synonym for blockchain, states that it is *distributed* over the network; over a peer-to-peer network. Every peer that is part of network, also called node, stores the whole blockchain or a big part of it. Nodes which

are only validating the integrity of the newly mined blocks are, in some networks, able to store the block headers only.

9.2.1.3 Types of blockchains and their openness

There are three types of blockchains which differ in their level of openness. The following list provides a short overview:

- **Public blockchains** are, as the name states, open to the public. Permission-less ledger is a synonym for public blockchain. These blockchains have no owner and anyone can participate by maintaining their own copy of the ledger. A consensus mechanism is used to decide who will write a block to the ledger. [18]
- **Private blockchains** are shared only among a defined group of entities and only this group can write on it. It is also called consortium blockchain [18]. There is a widely distributed opinion that a private blockchain contradicts the general purposes and definition of blockchain.
- **Hybrid / semi-private blockchains** As the name describes it probably well enough, a semi-private blockchain has parts which are private and parts open to the public [18]. A common use case is that only defined entities can write to the blockchain but the validation and the reading is open.

9.2.1.4 Anonymity / Heuristics

Anonymity is one of the most interesting key words when it comes to discussions about blockchain. Is it anonymous and transparent? Is transparency and anonymity both to a large extent even possible? Every public blockchain is globally readable but the users are still able to be anonymous? These are all questions which leave room for discussions.

Bitcoin protocol makes effort to stay anonymous e.g. every unspent transaction input goes to a newly generated address. Heuristics (described in the listing in section 9.3.3.3), however, can help to at least partially de-anonymize transactions respectively their sender and receiver addresses.

9.2.2 Information Visualization

Visualization Foundations

How are real world data mapped to digital representation. Introduction to loss of relation between data in the visualization process as well as what aspects are there of a visual representation of data.

Visualization Process

There are three main tasks when transforming raw data into a visualization[25]. First, the data has to be preprocessed and transformed. Then the data has to be mapped for visualization. Lastly we have to render the mapped geometrical data to the 2D image.

Preprocessing and transforming is performed by making sure that the data values are mapped to fundamental data types that can be used by computers. Then we have to deal with specific application data issues, such as too large data, missing values and input errors.

Mapping for visualization consists of mapping the data values to visual cues as well as the euclidean space. This can be done in numerous ways, which will be discussed in detail in the section information visualisation techniques. The eight visual variables are presented in the next item.

Rendering stands for transforming the mapped data to the 2D or 3D space. Viewing parameters are set here and the final visualisation is calculated.

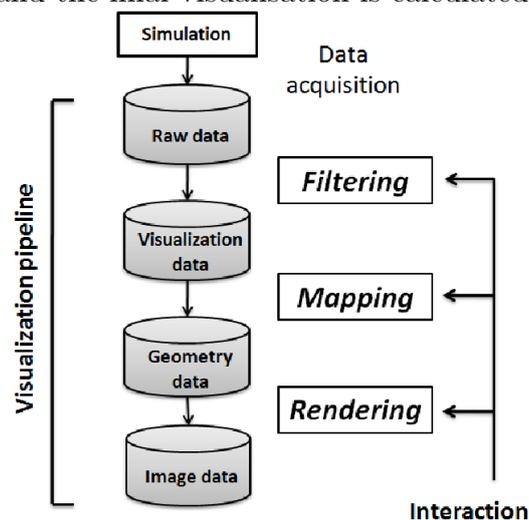


Figure 9.1: Visualization pipeline[9].

Eight visual variables

The eight visual variables refer to the mapping of attributes to a data value[25]. These eight variables are the main tool to maximize the effectiveness of the visualization. When mapped incorrectly the visualization may convey wrong correlations and introduce false conclusions. This is strongly connected to the human perception. The way humans build an understanding of what they see has to be the basis for choosing the mapping of the data.

1. Position

Position is a highly required variable. The spatial placement of the data values is the first step in perceiving data. If position were not used, the marks would overlap and the other variables could not be fully perceived. Optimal visualisations map each data value to a unique position where it is possible to distinguish the marks from each others and they do not overlap.

2. Shape

A number of graphical objects present themselves to be used to distinguish different marks from each other or group them together. Most often a point is used to

represent a data value. When using multiple unlike shapes, they should be easily distinguishable. When too many different shapes are used it can also lead to the marks not being easily indistinguishable. Every visualization will at the least include one shape.

3. **Size**

This refers to how much space the mark occupies in the visualization. As shape can be gradually changed (as opposed to shape) it often is used to map size to interval and continuous data variables. While size can be applied to categorize items, small increments are hard to be detected and two categories might be perceived as one. The choice of shape has an impact on the effectiveness of size. Size is not applicable as intuitively to all shapes.

4. **Brightness**

Brightness is, similar to size, gradually applicable, which makes it fit for being used on interval and continuous data variables. There is a limited range where we can perceive the difference in brightness.

5. **Color**

The color variable is divided into hue and saturation. Hue is defined as the dominant wavelength on the visual spectrum, whereas saturation stands for the level hue relative to gray. Using color, the different data variables have to be mapped to specific color. A common approach to map color is to manually map the colors to individual data, when there are not many colors used.

6. **Orientation**

The sixth variable, orientation, a preattentive visual cue. This means that we perceive the orientation of a mark before we cognitively notice it. It describes the rotation of a mark. Therefore orientation cannot be perceived on all shapes. Circles for example are perceived the same at any rotation. Therefore in information visualization the preferred shapes for conveying orientation have a natural single axis.

7. **Texture**

This visual variable can be considered a combination between shape, orientation, color and possibly others. It is most of the time associated with surface or region.

8. **Motion**

The most common uses of motion lie in the frequency of change and direction. Humans perceive similarities as well as outliers easily through motion. Motion does not only include position but a change of brightness can also be viewed as motion.

	<i>Points</i>	<i>Lines</i>	<i>Areas</i>	<i>Best to show</i>
<i>Shape</i>		<i>possible, but too weird to show</i>	<i>cartogram</i>	<i>qualitative differences</i>
<i>Size</i>			<i>cartogram</i>	<i>quantitative differences</i>
<i>Color Hue</i>				<i>qualitative differences</i>
<i>Color Value</i>				<i>quantitative differences</i>
<i>Color Intensity</i>				<i>qualitative differences</i>
<i>Texture</i>				<i>qualitative & quantitative differences</i>

Figure 9.2: First seven visual variables by bertin. In the last column it is noted where the best use of this variable lies in.

Effects of visual variables

As stated before the effectiveness of the visual variables is dependent on mapping them accordingly. A possible categorization of the effects is shown below and examples of fitting visual variables are given.

Selective visual variables: This can be used to divide data variables, nominal values, into groups.

Ordinal visual variables: Simplifies ordering the values (ordinal or qualitative data).

- Color
- Size
- Shape
- Brightness
- Texture
- Orientation
- Texture
- Size
- Brightness

Associative visual variables: Here all factors have the same visibility. Mostly used for nominal values.

- Texture
- Color
- Direction
- Shape

Separating visual variables: These make the elements visible, whereas the others are not visible.

- Texture
- Color
- Orientation
- Shape

Proportional visual variables: Variables in this group obtain a direct association of the relative size. Used most with ordinal or quantitative data.

- Size
- Orientation
- Brightness

Human Perception

How does the human process visual data. How can we use that as an advantage.

What is perception

Perception describes the process of recognizing, organizing and interpreting sensory information[25]. It deals with the signals from our sensory receptors such as the photoreceptors on our retina or the nerves in our fingers. For mapping data values to a visualization it is crucial to incorporate the way that people think. This helps conveying information. Studies show that the human visual system performs automatic computations and interpretations that are based on experience and assumptions. These have to be incorporated in creating data visualizations.

Preattentive Processing

We process sensory input before it reaches our brain. Tasks that can be performed in less than 200 to 250 milliseconds are called preattentive. By choosing the visual variables accordingly we can use this to our advantage in mapping data values to marks and conveying information. Preattentive processing can help in target detection, boundary detection, region tracking and counting & estimation. The visual features that can help in preattentive tasks are length, width, size, curvature, intersection, closure, hue, intensity, flicker and direction. In brightness, color, texture and shape are perceptual attributes, where differences are preattentively noticed.

Feature Hierarchy

While using preattentive cues can lead to an easier understanding and correct re-mapping of the data values, visual interference can occur that could hide or mask information. Feature hierarchy shows how certain tasks one type of visual feature is favoured over another by the visual system. When detecting boundaries, color is favoured over shape. Other such interferences are luminance-on-hue, hue-on-texture as well as hue-on-form. Here it is important to map the most main visual features should be mapped to important data attributes.

9.2.3 Information Visualization on communication systems

9.2.3.1 Visualization Techniques

When mapping data without an explicit spatial attribute, there are four main technique groups with which these multivariate data can be mapped[25]:

1. Point-based Techniques

These include the techniques, that represent the data values as single-point marks. The most well known are scatterplots (Figure9.3) or scatterplot matrices. Here a point is mapped to the euclidean space along the axes, depicting one attribute. The matrix variant means that the data attributes are plotted against each other in a matrix of scatterplots (Figure9.4), here the data attributes are plotted on the y-axis of the matrix for one row and on the x-axis for one column so all possible scatterplot variations are visible. Other examples include the rugplot (Figure9.5), where similar to the scatterplot matrix a number of scatterplots are connected along the fringes. The principal component analysis, where the two most salient data attributes are plotted on one axis and the other attributes are plotted on other visual variables. RadViz (Figure9.6) is a technique where the different attributes are divided similar to stokes on a wheel inside the circular plot. The intersection of the stoke with the circle is set as an anchor point. The higher the value of a data point on one attribute, the closer it is set to the respective anchor point. Here we have to keep in mind that we get a different result when applying a different order on the anchor points.

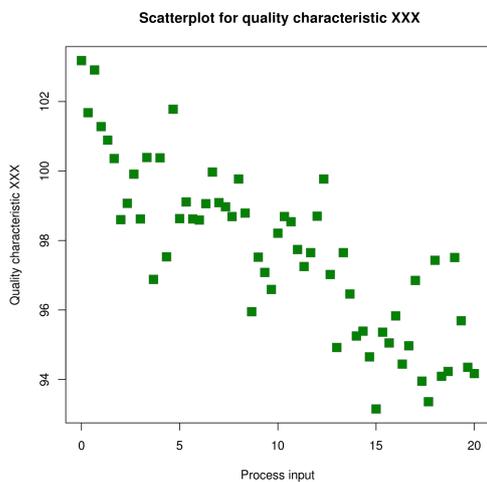


Figure 9.3: Scatterplot example.

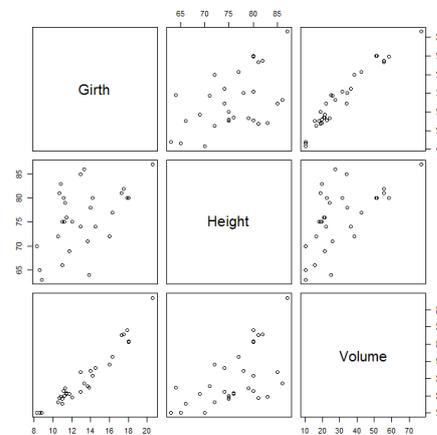


Figure 9.4: A scatterplot matrix for visualizing the height, girth and volume of trees.

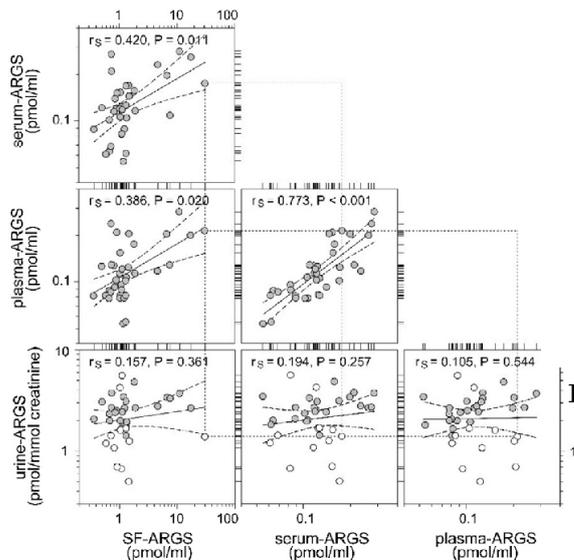


Figure 9.5: Rug plot example.

2. Line-based Techniques

These include approaches like the line graph (Figure9.7), where a line connects the different data values over one attribute, most commonly time. Another line-based technique is parallel coordinates (Figure9.8), where all the data attributes have one axis line along the x-axis and the y-axis depicts the value at this attribute, the values of one data value are connected by a line. The radial version of the parallel coordinates also exists in different variants (Figure9.9).

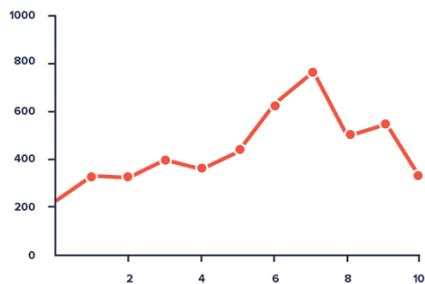


Figure 9.7: Line-Graph example.

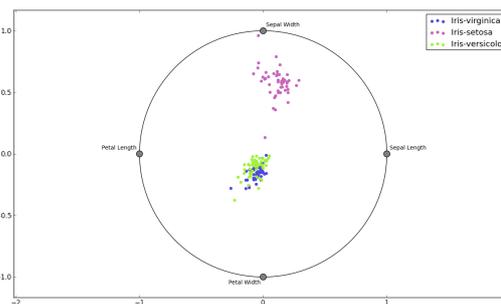


Figure 9.6: RadViz example of the Iris dataset.

Parallel coordinate plot, Fisher's Iris data

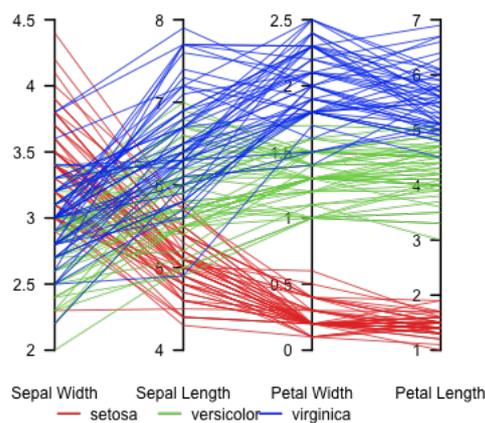


Figure 9.8: Parallel graph representation of the Iris dataset.

Gymnast Scoring Radar Chart

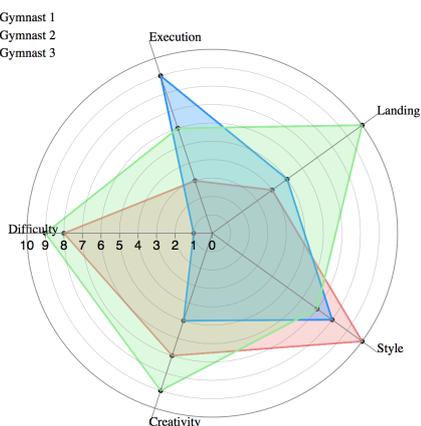


Figure 9.9: Radar Chart depicting gymnasts performance.

3. Region-based Techniques

Region-based visualizations use elements with a 2D area. Here the size and shape allow the mapping of multiple data values. It is often used to aggregate and summarize data. The most common example is the bar chart (Figure9.10), where the height of a bar can be easily perceived by a person. The histogram is an extension of the bar chart, where the y-axis depicts time. There are also 3D variants of the bar chart to compare multiple data side by side. In Heatmaps (Figure9.11) a matrix is drawn and the fields are colored according to their value, most often from red to green. The radial axis bar chart (Figure9.12) plots, similar to the point based RadViz, bar charts onto a circular plot, this can be extended to a Spiral, where the spiral axis depicts time like in a histogram, usually one loop is mapped to a uniform measure (day, month, year, etc.).

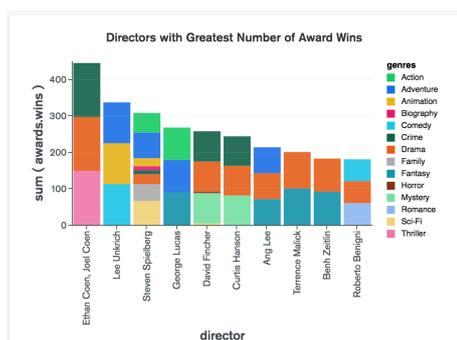


Figure 9.10: Example of a stacked bar chart.

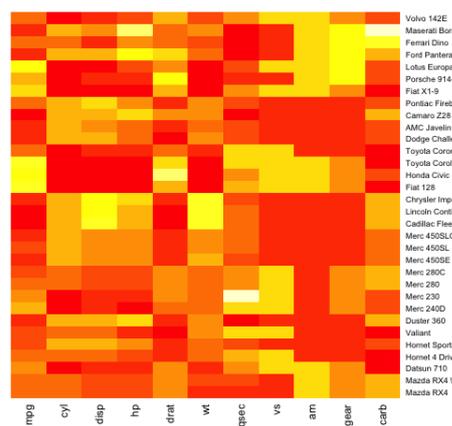


Figure 9.11: Heatmap mapping Car with its attributes.

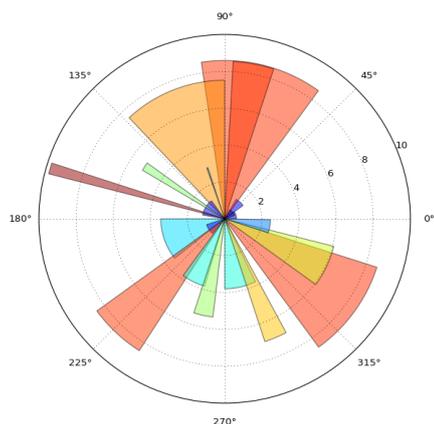


Figure 9.12: Radar Chart depicting gymnasts performance.

4. Combinations of Techniques

When combining techniques we inherit many of the advantages but also of the disadvantages from the used techniques. Here common techniques are glyphs like the flower used in the communication garden system described in the next section or Chernoffs Faces (Figure 9.13). Various attributes can be mapped to a glyph such as a face or arrows. Here the users often have an inherent association with the glyph that has to be included in the thought process during mapping. Dense pixel displays (Figure 9.14) are a way of visualizing large quantities of data. Here each pixel is mapped to a data value by color and filled polygons represent a data attribute. The hardest part about dense pixel displays is the mapping of position, as there is no inherent structure to the presenting subimages resulting from one data attribute. The layout of these can be done in various ways, especially when there is no inherent order on the values themselves.

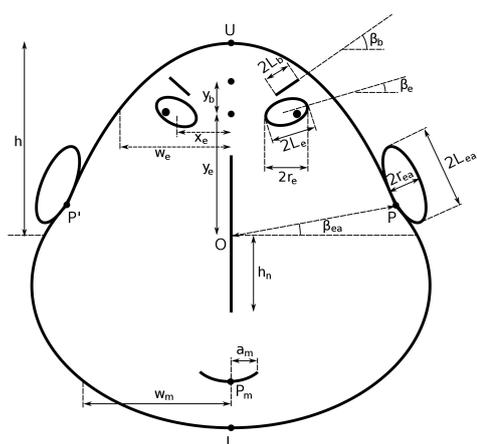


Figure 9.13: The chernoff faces are one of many variants of the glyph technique.

Dense Pixel Displays

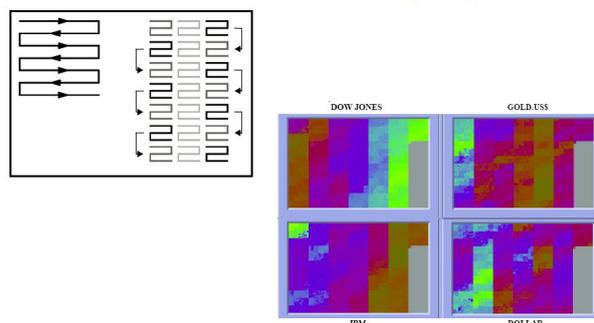


Figure 9.14: Dense pixel display.

9.2.3.2 Application 1: Communication Garden System

The communication garden system[2] is the prototype of a scientific attempt to visualize computer-mediated communication. Here archives of such communication are represented

using a flower to visualize the discussion content and participants' behavior. The goals of this research was to improve CMC archives, by making the data comprehensive and giving an overview, to find the right thread easily. This would improve the problem of knowledge loss in corporations and help newly employed understand the community of these threads and gather the knowledge efficient.

The research proposes a three-layer system architecture (Figure9.15) that is built on information representation, information categorization and information visualization.

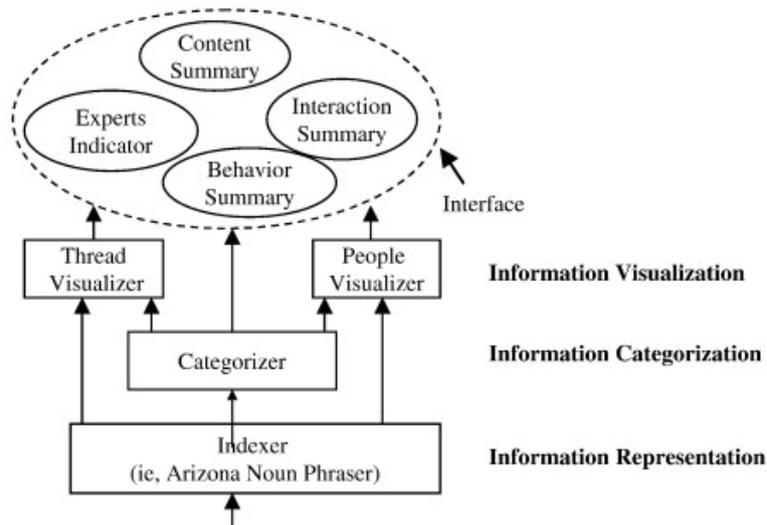


Figure 9.15: The three-layer system architecture as shown in the research paper[2].

- **Indexer**

The indexer uses the noun phrase technique to capture the linguistic representation of document content. It extracts key noun phrases from the threads.

- **Categorizer**

Taking the input of the indexer, the categorizer automatically categorizes the content of a thread and identifies subtopics. The technique of self organizing maps, a common technique for mapping data values to the euclidean space using an artificial neural network, is applied here to map the large quantities of information with keeping the relations as much as possible. The result is a spatial representation of the indexer's output.

- **Thread Visualizer**

The paper employs an unique mark as their data points. The threads are represented as a flower, where the petals of the flower represents the number of messages in this thread, the number of leaves are representative of the number of people that participate in this discussion and the height of the flower correlates to the length of the threads' existence. Additionally the flowers are mapped to the location in the interface according to their starting time. This is called the Thread Visualizer (Figure9.16).

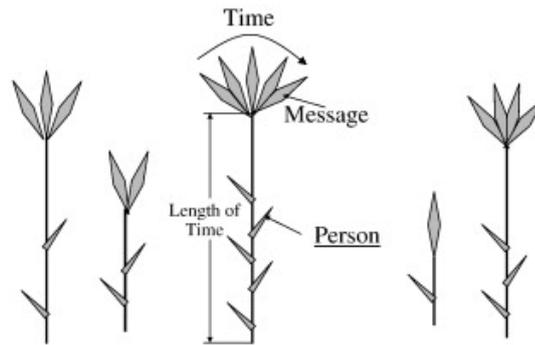


Figure 9.16: Example of a thread visualizer[2].

- **People Visualizer**

Similar to the thread visualizer, the people visualizer (Figure 9.17) uses the metaphor of a flower to display the users activity. The leaves stand for the number of threads in which the user has posted, the petals equals the number of messages, the height indicates the time that a user has been active in the community or topic area.

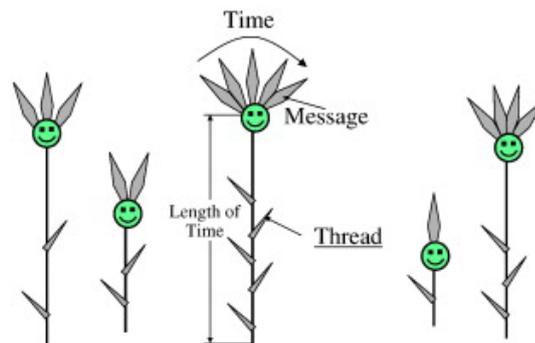
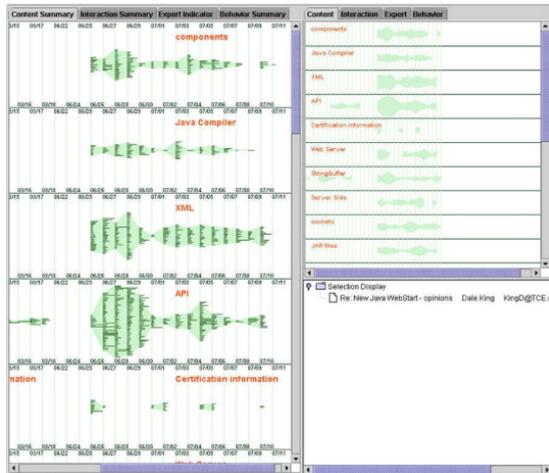


Figure 9.17: Example of a people visualizer[2].

Interfaces

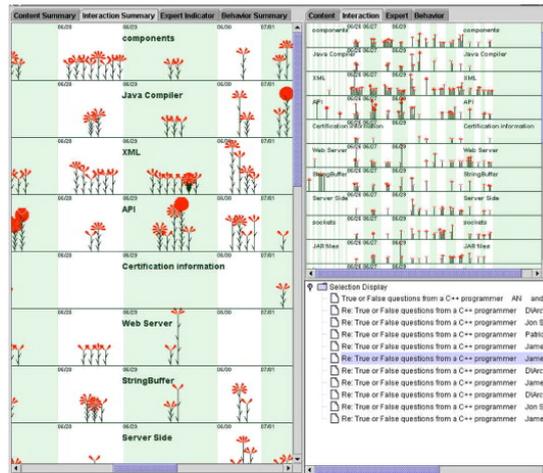
Upon this use case the researchers built a prototype interface that consists of a visualization of the content summary (Figure 9.18), the interaction summary (Figure 9.19), expert indicator (Figure 9.20) and behavior summary (Figure 9.21). Each of these visualizations is representative to one aspect of a CMC process. The content summary describes the change over time of each subtopic, the interaction summary describes how active a discussion is within a subtopic. The expert indicator uses the people visualizer to locate the most active users and the behavior summary describes the users behavior during the CMC process.



Description of the Display panel:

- The x-axis represents time.
- Categories generated by the SOM are laid out vertically.
- Each green line represents one message.
- The vertical thickness of each subtopic indicates its activity on a particular day.
- The length in the x-dimension of each subtopic = time duration of that subtopic.

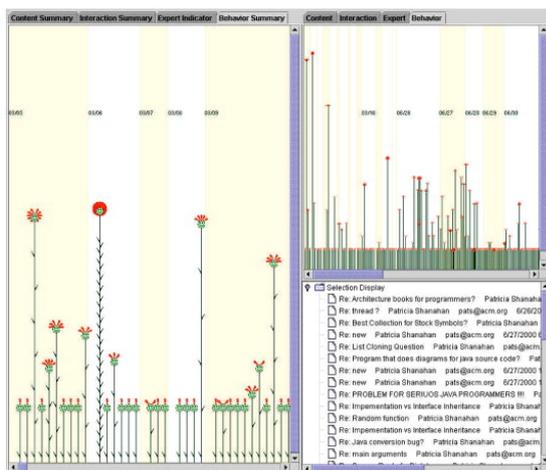
Figure 9.18: Content summary view of the prototype[2].



Description of the Display panel:

- The panel is divided into sub-gardens based on the SOM output. Each sub-garden is a subtopic.
- Each flower is one thread.
 - number of petals = number of messages posted for this thread
 - number of leaves = number of participants in this thread
 - height of flowers = the time duration of this thread

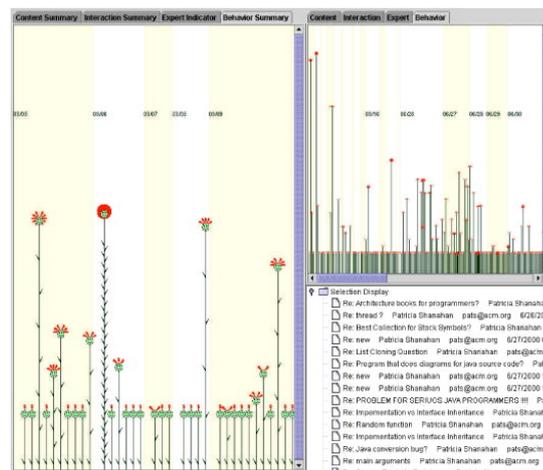
Figure 9.19: Interaction summary view of the prototype[2].



Description of the Display panel:

- The interface is divided into sub-gardens based on the SOM output. Each sub-garden is a subtopic.
- Each flower is one Person.
- number of petals = number of messages posted by this person for this subtopic
- number of leaves = number of threads this person has participated in the subtopic
- height of flowers = how long this person stayed in this subtopic

Figure 9.20: Expert indicator view of the prototype[2].



Description of the Display panel:

- The entire community is one garden.
- Each flower is one Person.
 - number of petals = number of messages posted by this person in this community
 - number of leaves = number thread this person participated in this community
 - height of flowers = how long this person stayed in this community

Figure 9.21: Behavior summary view of the prototype[2].

In each of these interfaces the left-hand panel is the display panel, the upper-right panel represents an overview, both of these panels consist of tabs that represent the four views. The last panel is for the messages that can be displayed, for the current view of interest. The paper concludes that the combination of existing information analysis techniques can improve the understanding of discussion content and users behavior. It shows its limitations that lie in design, i.e. the smiley faces that might suggest emotion, limitations of the computer screen, not all threads/subtopics can be viewed at once, and last that this prototype does not include the social network aspect of CMC archives.

9.2.3.3 Application 2: Interactive Visualizations for Planning and Strategic Business Decisions in NFV-Enabled Networks

The goal of this work[16] is to assist network operators in improving their capabilities of NFV planning tasks. Their database consists of static and dynamic information from NFV environment. This will help in understanding the business strategy and tenants' demands. The used visualization techniques are Hierarchical Edge Bundling and Sankey Diagram.

Hierarchical Edge Bundling is a technique where all the edges are symmetrically divided on a circle plot. All the relations are represented by a line between the two edges. Lines that are going in the same direction are bundled for less noise. Sankey Diagrams depict a flow of some kind. Most common usage shows the division of some data into smaller subparts in a hierarchical manner. The hierarchy though is not mandatory.

Architecture

The papers' prototype builds on the existing VISION architecture (Figure9.22). There are three main steps in creating the visualization. First step is for the data collector to read the relevant information to the database. This data is gathered from the Management and Orchestration framework, VNF monitors and descriptor files. The next step consists of the data processor to prepare the information for the Information Manager. Lastly, according to the operators input, the visualization manager selects the templates and creates the visualizations.

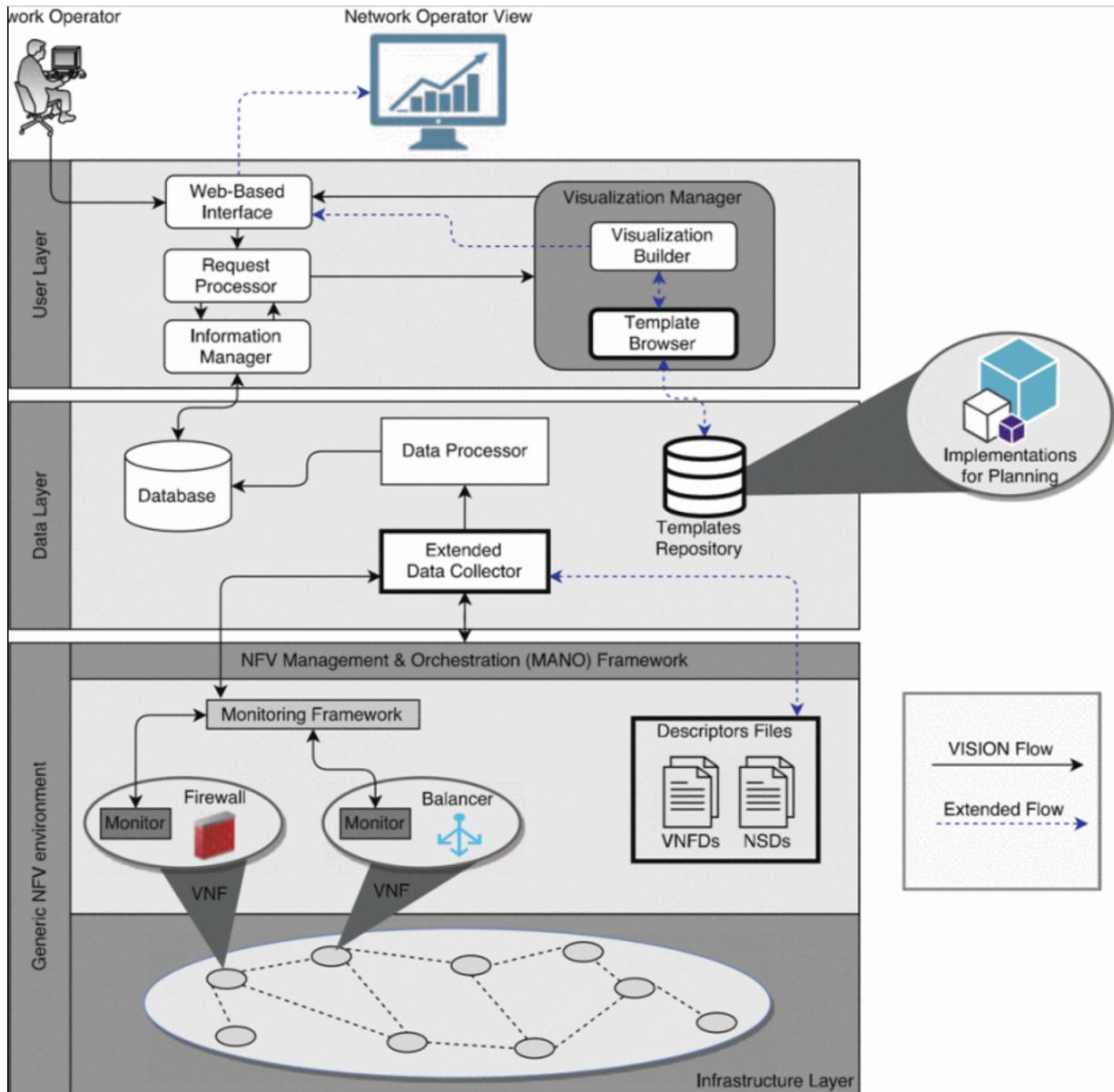


Figure 9.22: The extended vision architecture[16].

Visualization of Relationships and Business Demands

The technique used to visualize relationships and business demands is an alteration of the Hierarchical Edge Bundling, ClusterViz (Figure9.23). Here additional to the edges in the central circle plot, that are of categorical type. The data points have other numerical attributes mapped to circular bar charts around the central plot. The papers' adaption represents the business actors, blue depicting tenants, their relationships are mapped by blue lines in the centre. The first ring outwards represents the subscriber count of the business actor in a blue bar chart. The outermost circle depicts the packets per second associated with the node. These mappings can be adjusted to any of the values in the database by the data collector.

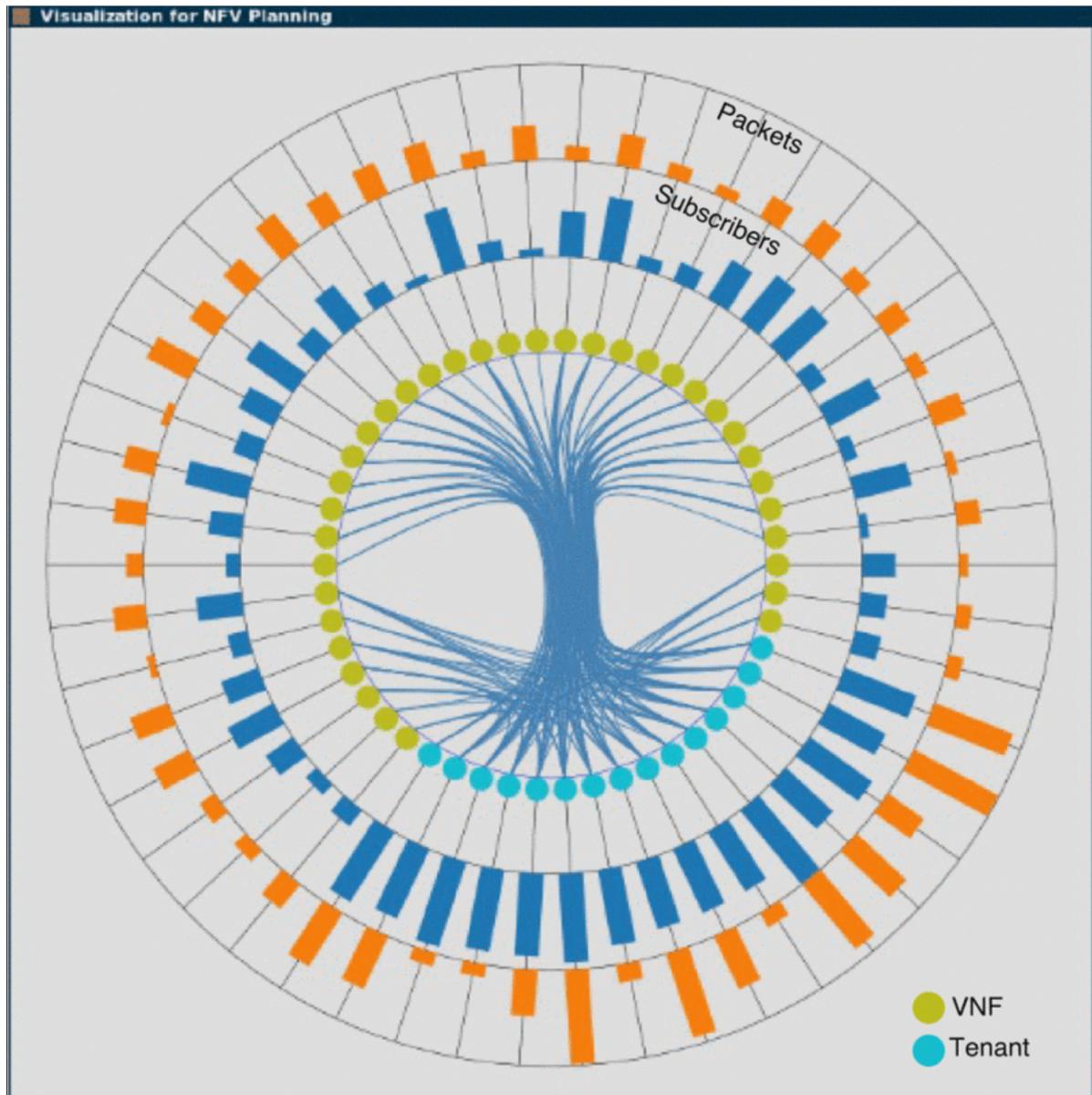


Figure 9.23: ClusterViz generated from the prototype[16].

Visualization of Allocated Resources and Revenue Generation

The Sankey diagram (Figure 9.24) allows the business operator in gaining insight about the profit potential of individual services, supporting decision-making for the business strategy. It does so by visualizing the amount of resources allocated to run VNFs of a specific group as well as the total income obtained by all groups. The diagram supports the interpretation of how many resources are spent to generate the business revenue. The flow goes from left to right and represents the resources available through the rectangles. The height of the diagram is mapped to 100%. The operator can select resources to be included in the diagram as well as the group of VNFs

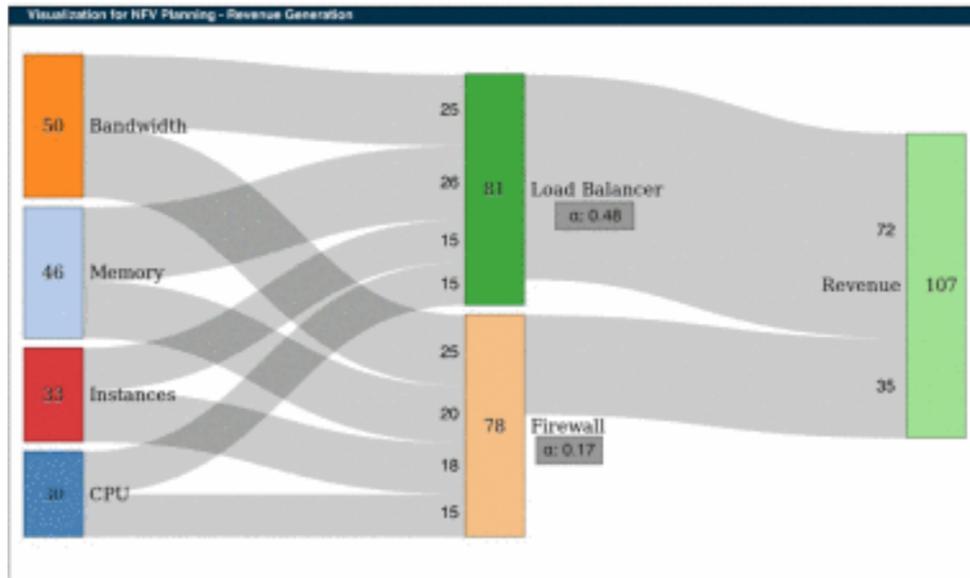


Figure 9.24: The example Sankey diagram[16].

The paper concludes that the visualizations ”provide insight for a better planning of infrastructure investment, resource allocation policies, and services pricing”. This can make infrastructure investment and resource allocation more effective and improve service pricing. It gives a better oversight to the operators by giving more insight.

9.3 Blockchain Data Extraction

In this section, a short overview on challenges and opportunities of blockchain data extraction is provided. The second part presents examples and techniques how some of these challenges are tackled and some of the opportunities are seized.

9.3.1 Challenges

One of the biggest challenges for blockchain access and data extraction is the usability problem. Developers as well as users are experiencing this problem because compared to a normal database or also a distributed database it is not easy to extract information from a blockchain [14].

Another challenge is the anonymity. For some people it is an advantage but exactly because of these people, a lot of researchers try to reduce the anonymity of blockchain respectively to de-anonymize the data [21]. But de-anonymizing the blockchain data is a big (computational) effort. Most blockchains, such as Bitcoin and Ethereum anonymize the information of the sender of a transaction to protect the privacy of the user [19]. The user identity is hid by providing an address only.

Moreover, the amount of data leads to an additional challenge: Every time data needs to be extracted from the blockchain, basically the whole blockchain needs to be synced respectively all the data downloaded to the local machine. Syncing from scratch needs a lot of machine performance and traffic and can take days if the internet or machine is not well enough. Not only the data size is a challenge, but also the type of data which is saved to the blockchain: Often only meta data is saved, combined with a hash-like object which represents the data itself which is, in these cases, too big to be saved on the blockchain. Since a hash input cannot be reproduced from the hash output, no user other than the sender knows what the data represents. Therefore, the only thing where interesting can be fetched from is the non-hashed meta data.

Another approach that is used when trying to save too big data to the blockchain, is splitting the data for multiple transactions. Lets say, the user wants to save an image on the blockchain not only for proving that he saved the image to a specific point in time but also for storing the image publicly available for everyone. Then it is not possible to save the hash only. The user needs to split the bytes data of the image in multiple chunks and send these chunks in separate transactions to the blockchain. Indeed, the use case is not the most common one, but it exists. The extraction of data from these multiple transactions includes figuring out that the transactions are belonging together and rebuilding the whole input data (e.g. the image). This is another challenge faced by blockchain data extraction.

9.3.2 Opportunities

There are many opportunities for blockchain data extraction. The following list briefly provides a few of them:

- de-anonymizing blockchain (e.g. the Bitcoin blockchain) can improve security by identifying criminals, malicious addresses and makes for example the anti money laundering laws enforceable [19].
- by extracting and displaying the data on the blockchain the blockchain advantage *transparency* can be made accessible for normal users without blockchain knowledge.
- When combining blockchain with big data projects, the data integrity of the big data pool can be assured. For this to happen the blockchain data needs to be extractable in an efficient and automated way.
- data integrity for big data projects -> blockchain ascertains the origin of data through its linked chains
- real-time blockchain data extraction would allow the user to instantaneously process data extracted data and calculate additional information.
- simple blockchain data extraction services, so called blockchain explorers (see section 9.3.3.1), allow real-time analysis such as network traffic analysis, transaction fee involvement, mining difficulty history etc.
- data extraction allows to but splitted data sets in multiple transaction back together. It therefore makes the storage of big, not hashed data useful by providing a method to read the stored data.

9.3.3 Techniques & Examples

9.3.3.1 Blockchain explorers & parser

There are several systems around which all provide a similar service: So called *blockchain explorers* provide simple overviews and insights into almost real-time blockchain interactions and activities. Pratama et al. [14] defines blockchain explorers as an online service which provides blockchain information to its users. This data is related either to an account, a single transaction or a block of the blockchain.

One of the most known blockchain explorers is Etherscan.io, which provides minimalistic functions and information about data on the Ethereum blockchain. Developers can also use the REST API of Etherscan to pull data from Etherscan to their own applications. Another well-known explorer is Etherchain.org. It provides almost identical information to the user than Etherscan.

Both websites provide some simple data visualizations too. This includes market capitalization charts, block difficulty growth charts, node tracker maps or visualizations about miners and their size.

The chart in figure 9.25 shows the top 25 miners for the Ethereum blockchain. For more information about visualizations, see the next sections.

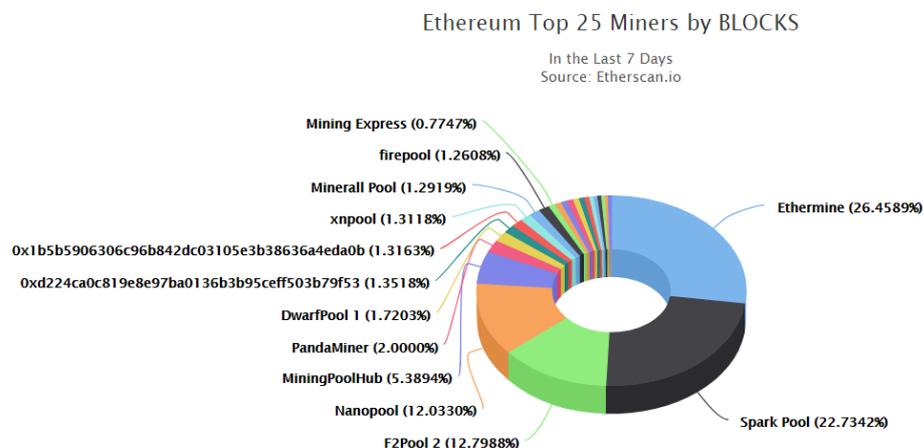


Figure 9.25: Ethereum’s top 25 miners. A visualization from the Etherscan.io blockchain explorer

In the previous paragraph, Ethereum explorers were listed, but of course also blockchain explorer for worlds first crypto currency do exist: There are a lot of Bitcoin explorers such as blockchain.info, which is also exploring Bitcoin Cash and Ethereum, BTC.com or Blockcypher.com (which provides more than one explorer too).

Functionality of blockchain explorers

Almost every blockchain client such as Geth or Parity implement JSON RPC endpoints. These endpoints can be called to get data from the blockchain. An interested user can either to that with Parity or Geth itself or the user, also users unfamiliar with command line or code, can visit Etherscan which does the JSON RPC call for you.

Etherscan also includes a virtual machine with which it is possible to track internal transaction from Ethereum blockchain [26]. Often this is combined with a cache: The explorers cache the more important data to a database that the call to the blockchain does not have to happen every time. This improves performance and usability.

Znort987’s blockparser

Znort987 is a blockchain decoder respectively parser which ”chews ’linearly’ through the block chain and calls ’user-defined’ callbacks when it hits on certain ’events’ in the chain”, as it is defined on its Github page [17]. This tool is not the most efficient tool, as the owner states, but it makes its job and parses the whole blockchain and the user can specify which *events* should be triggered. The parsed data is saved to storage and can be used later for all kind of analysis. This parsers is used by some of the following examples.

Some applications do not keep the parsed data in the local storage. There exist several services which try to provide the parsed data as a database which can be queried. For examples `blockchainsql.io` [6] enables the user to query the Bitcoin blockchain with SQL statements.

9.3.3.2 Query Layer support

When it comes to querying the blockchain data every blockchain explorer, as mentioned in the previous section, comes to its limitations. But why is this the case? One could argue that since the blockchain need to be fully downloaded to interact with it, for an interested

party it should be possible to access the locally stored blockchain and pull information from it. This is possible indeed but accessing and query information is not the same thing. What makes the whole process of querying blockchain data difficult is the underlying storage architecture of most blockchains. The main blockchain usage is not querying the data it is writing to the blockchain. For example the Ethereum blockchain is based on *LevelDB*. LevelDB is high performing in writing data, especially in writing large amount of data. Some additional advantages are that it is very space-efficient and that no additional database needs to be run separately. But for analytical task with more complex queries, LevelDB is not very well suited [26].

Li et al. [26] provide a query layer for analytics on a blockchain system. The proposed system is developed for Ethereum but it is claimed that the techniques can be applied to other blockchains too. It can be used with a local API or also by a RESTful service and range queries and top-k queries are supported.

Figure 9.26 shows the architecture overview of EtherQL. It contains basically four important layers which will be described in the following paragraphs. The *Sync Manager* is listening on blockchain events and syncs the new data to the cache. From the cache three different handlers (block, transaction and account handler) read the data and process it in different ways. The handlers are writing the data into the *CRUD repository* in the *persistence framework* which support querying with SQL statements. The API and the RESTful interface access the data from the CRUD repository and therefore build the interface for the developers [26].

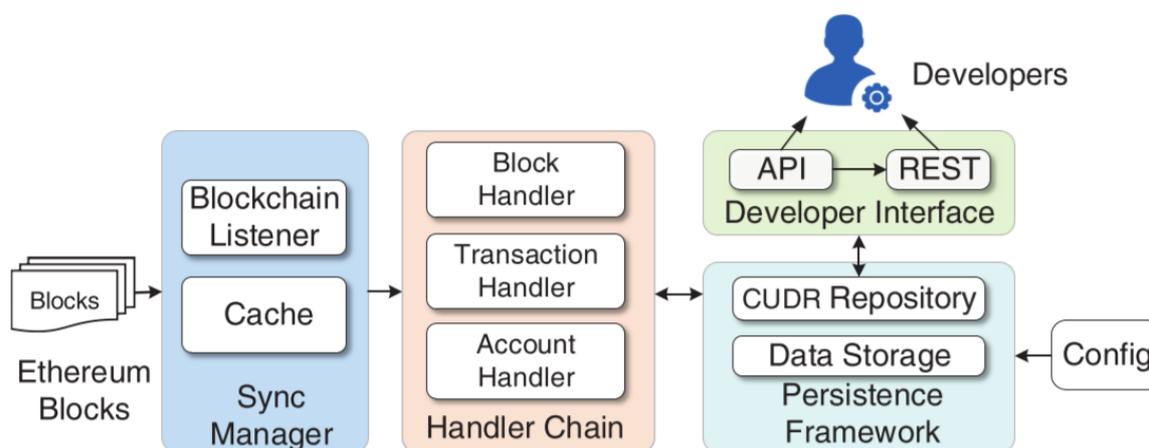


Figure 9.26: The architecture overview of EtherQL: A query layer for blockchain systems [26]

Reverting states and undo changes is for a SQL database a more complex task. In every blockchain, in some more often than in others, unintentional consensus forks can happen and need to be handled. Li et al. developed the Sync Manager which can identify forks beforehand and can reduce the chance of forks [26].

The handler chain layer can be seen as an interface between Ethereum blockchain and the data storage of the persistence framework. In the reference architecture in 9.26 the blockchain events are processed by three different handlers: the block, transaction and the account handler. Each handler reads different data from the blockchain state and stores it to the CRUD repository. The good thing is that this handler chain can be extended with other subhandlers focusing on other data [26].

The persistence framework layer is used to persist the raw blockchain data and to allow SQL queries on the stored data set. So creating, updating and deleting functionalities are used for persisting the data and read functionality to allow SQL queries (CRUD). For

this purposes MongoDB [22] is used; a NoSQL database which is easily configurable for scalability purposes [26].

The developer interface provides the ability for developers to query the CRUD repository. Developers can use the four listed query modules:

- queries supported by Ethereum (the ones which are also available in an Ethereum client)
- extended Ethereum queries
- range queries (e.g. listing all uncle blocks for a specified time window)
- top-k queries (e.g. the five richest Ethereum accounts)

The introduced system was tested according performance against native Ethereum clients such as *go-ethereum*. Range and top-k queries cannot be compared since these kind of queries are not even possible with *go-ethereum*. Figure 9.27 shows the performance comparison for three queries. Namely the queries *get a block by block number*, *get a transaction by transaction hash* and *get the balance of an account by its address* [26]. One can see that query language support system not only extend the querying possibilities but also shows better performance for queries which are also possible from an Ethereum client such as *go-ethereum*.

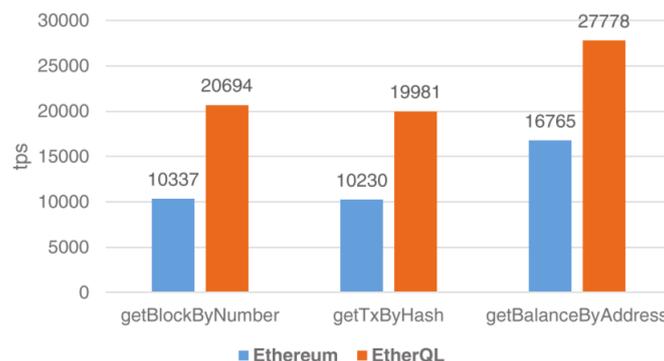


Figure 9.27: The comparison between Ethereum client (*go-ethereum*) and EtherQL performance in three different blockchain queries [26]

9.3.3.3 Blockchain de-anonymizers

As already mentioned in section 9.2.1.4, the anonymity of blockchains is often part of interesting discussions. Also because a lot of researchers are working on that topic.

There are several projects which try to de-anonymize the blockchain data, especially on the Bitcoin blockchain. It is possible to define so called *heuristics* to, at least partially, de-anonymize blockchain users.

These heuristics are prerequisites for a lot of de-anonymizing tasks and two of them are therefore described in the following list. These are heuristics based on the Bitcoin blockchain.

1. **Multi-input transactions:** When a user has two wallets with available Bitcoins and he/she wants to perform a payment where the amount is higher than both of the available Bitcoin wallets. He/she will take both of the wallets as an input for the transaction. He/she usually won't split the transaction because of losing money due to the transaction fee. Therefore, we can assume that two addresses belong to the same entity when they are input for the same single transaction [21].

2. **Shadow/change address "guessing"**: The second heuristic is based on the opposite than the first heuristic. While the first heuristic is triggered when one wallet has not enough money to perform a payment, the second is triggered when the wallet has too much Bitcoin for a transaction: Since the Bitcoin protocol is forcing every transaction to spend the whole input (transaction input is always transaction output), every time the input is bigger than the output, the protocol generates a new address where the change/leftovers is transferred to. This is done automatically to improve anonymity. Now the question is, how the heuristic can figure out which of the addresses is the sender's address and which one is the receiver's. For this, there is deterministic approach: It is safe to assume that the address which never appeared on the blockchain before, is the so called *shadow address* [21].

Spagnuolo et al. [21] developed a framework called *BitIodine*. BitIodine is a collection of modules which is able to automatically parse the blockchain. It clusters addresses, classifies addresses and users, builds graphs and allows the user to export and visualize information from the Bitcoin network [21].

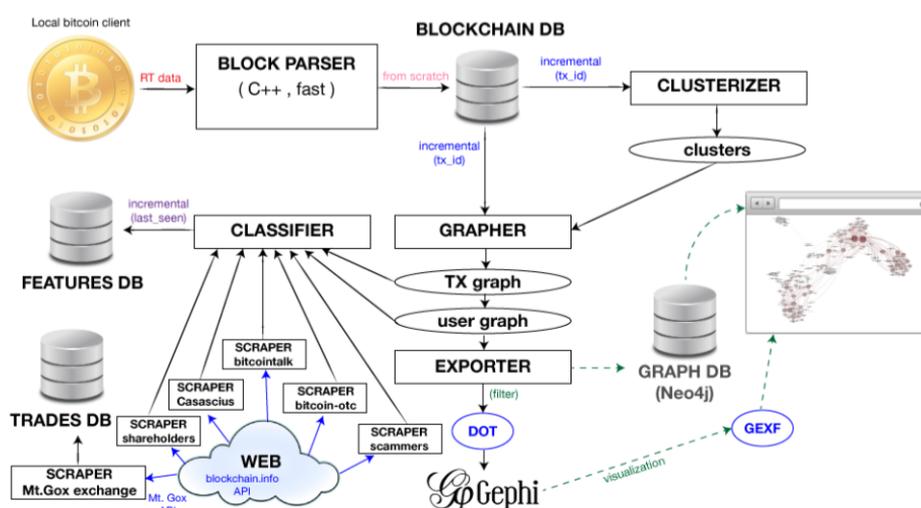


Figure 9.28: The architecture of the tool *BitIodine* [21]

BitIodine starts with parsing the blocks from the folder of the *bitcoind* client and stores the parsed data to the so called *Blockchain DB*. This relational blockchain DB uses a customized schema to be as performant as possible.

The so called *Clusterizer* accesses the data on the database and tries to find groups of addresses belonging to the same user by clustering the data. It uses the two heuristics described above [21]. During the same time the Clusterizer is working, Scrapers are crawling the web for Bitcoin addresses already known belonging to some real users. It also uses blockchain explorers (see section 9.3.3.1) to gather additional information. The scrapers which are used are easily adaptable.

The *Grapher* generates a transaction and a user graph by reading the Blockchain DB and the cluster output. The *Classifier* then reads these graphs and classifies both single addresses and recognized users. Boolean flags such as *one-time address*, *disposable*, *old*, *new*, *empty*, *scammer*, *miner*, *shareholder*, *FBI*, *Silk Road*, *killer* and *malware*. The user can specify if the classification should happen on the whole blockchain or on a predefined set of addresses. The classifications are saved to the *Features DB*.

Spagnuolo et al. [21] evaluated the software by investigating real-world cases: They did for example a so called *ransomware investigation* on *CryptoLocker*. CryptoLocker is a ransomware that encrypts the personal files of a victim and the criminal ask for a ransom-payment in usually Bitcoin to decrypt the data again. They used BitIodine to

gather the clusters of CryptoLocker addresses and to find payments from victims. Figuring out addresses which belong to CryptoLocker is an easier task: They googled parts of the text of the malware and read Reddits threats, both resulted in several addresses belonging to CryptoLocker. It was confirmed by BitIodine's classifier that the addresses belong to plenty of clusters totaling in 2118 addresses. They could also identify several ransom payments (771) which total an amount of 1226 BTC (which equals an approximately value of USD 6.5 Mio on April 17th 2019).

A lot of other interesting projects according de-anonymization exist. For example Zhu et al. [19] realize a system to analyze the Bitcoin network from two different aspects and combine these aspects. They analyze the blockchain data itself to resolve Bitcoin addresses as well as the Bitcoin P2P protocol to evaluate IP addresses. Their goal is to map Bitcoin with IP addresses.

9.4 Blockchain Visualizations

9.4.1 Example 1: Bitcoin transaction flow visualization

Visualization techniques can be used to display Bitcoin transactions all over the world. Nodes of different size determine the value of a transaction, whereas edges may offer information about the origin of the transaction and the recipient. A broad set of illustrations already exist. Examples of them are presented below.

9.4.1.1 Bit Bonkers

Bitbonkers[5] is a live streaming of bitcoin transactions from all around the world. If you go to the website first of all a cube is placed on a plate. The cube stands for the latest block from the blockchain and the size is determined by how many kilobytes it is. On average every 10 minutes a cube is mined. And if somewhere on this planet a Bitcoin transaction is made it is shown as colored ball dropping on the plate. The different colors of the balls representing the value of the Bitcoin transactions. The six different colors of the transaction balls stand for:

- Purple balls stands for transactions with a value greater than 1000 BTC.
- Blue balls for transactions between 100 BTC and 1000 BTC.
- Yellow balls for transactions between 50 BTC and 1000 BTC.
- Orange balls for transactions between 10 BTC and 50 BTC.
- Green balls for transactions between 1 BTC and 10 BTC.
- Red balls stand for all transaction with a value less than 1 BTC.

Also you can select a transaction ball to get the exact value of the transaction in bitcoin. It will be displayed on the right side of the page. Additional to the value of the selected ball you can see there the value of the biggest transaction. Above the total value of all Bitcoin transactions since you started the BitBonkers visualization and there on top the value of the current dropped ball. Press space and all balls on the plate will be removed and the BitBonkers visualization starts again.

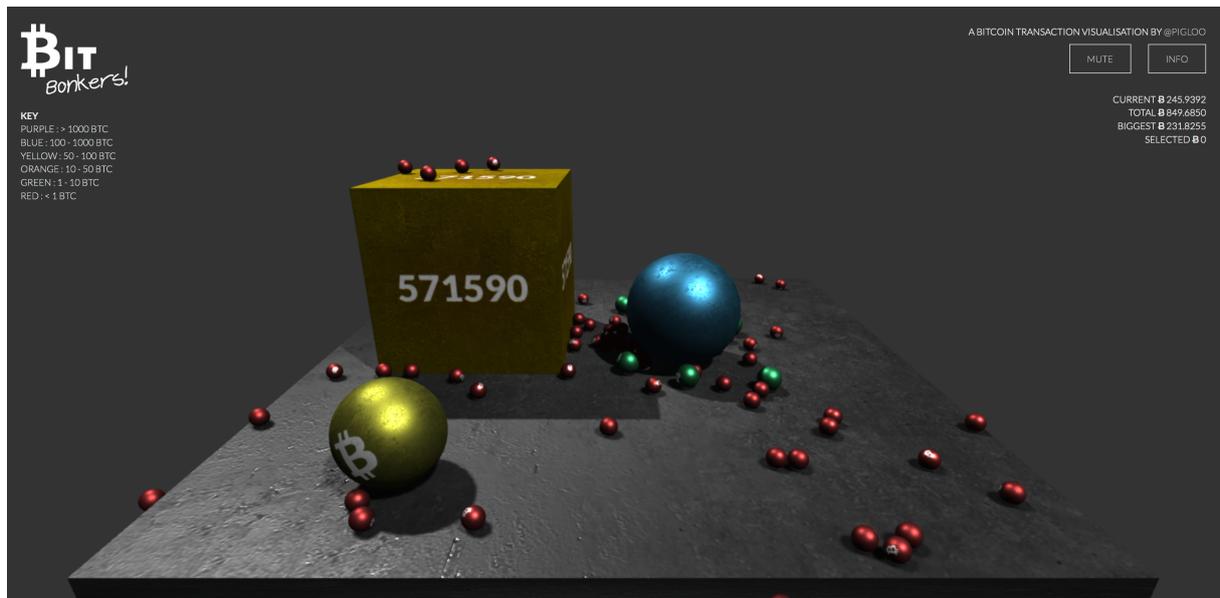


Figure 9.29: Balls representing transactions in Bit Bonkers

9.4.1.2 Bitnodes

Bitnodes[4] is developed to gather informations on Bitcoin nodes in order to estimate the size of the Bitcoin network. This will be done by finding all the reachable nodes in the network.

The Bitnodes website is clear structured and easy to navigate. It includes a live map of the world showing the concentration on Bitcoin nodes across found in different countries around the world. Also you can get more informations by visiting the leaderboard table, there you get which country has the most nodes in volume and percentage terms.

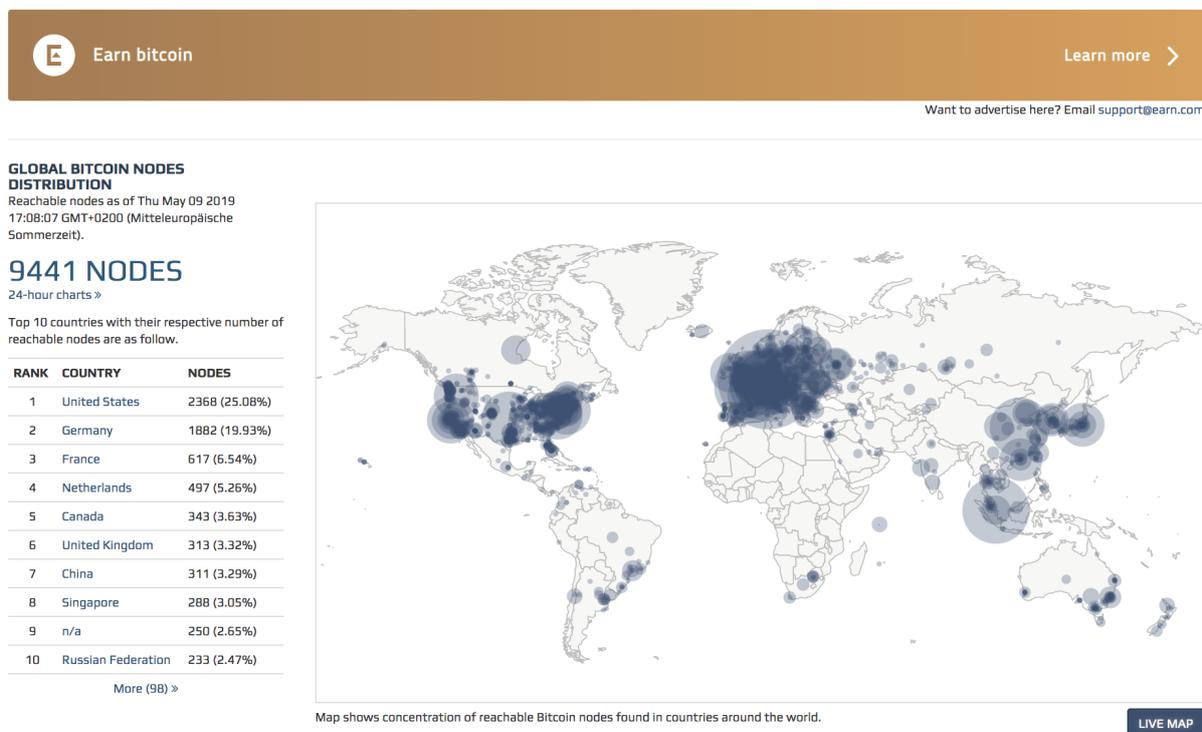


Figure 9.30: Main page of the website including the leaderboard table and snapshot of the map with button to get to the live map

By clicking on the live map button 9.30 it is possible to access it and displaying all reachable nodes on the map. Additionally informations will also be visible there like total number of reachable nodes and top user agents 9.31.

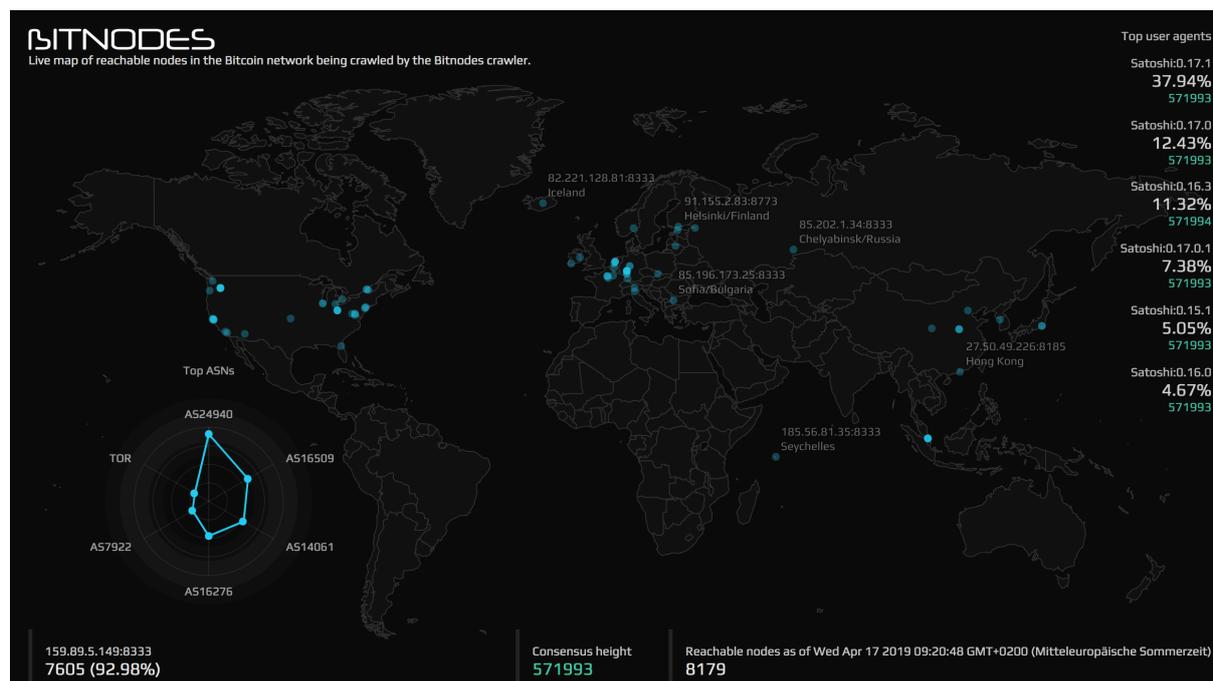


Figure 9.31: Live Map of reachable nodes in the Bitcoin network

There are other visualizations that are developed by Bitnodes and accessible through the main page:

- 24-hour Charts - shows how many nodes were online in the last 24 hours, details about the used bitcoin versions, transaction informations etc.
- Network Map - visualize all reachable nodes in the Bitcoin network

9.4.1.3 Blockchain 3D Explorer

Blockchain 3D Explorer[7] visualizes Bitcoin transactions in 3D and Virtual Reality(VR). It is an open-source and multi-platform application. There are different visualizations developed by Blockchain 3D Explorer with different functionalities:

- Flow Visualization Real-time 3D rendering of Blockchains to see how networks are organized.
- Virtual Reality Experience Blockchain networks in Virtual Reality.
- Blockchain Analysis 9.33 Analyze specific addresses and transactions and trace the flow of Bitcoins and tokens around networks.

Every of these three different visualizations types has the same basic 3D construction, which is shown in the image below 9.32.

The red cubes stand for Bitcoin addresses and the blue sphere is a bitcoin transaction. The arrow linking to them show the flow of value for a single transaction. The next image shows the Blockchain Analysis functionality including the network flow with details of the nodes: Amounts, IDs, dates and so on can be displayed by clicking on the cubes or spheres. As future work, visualizations for Ethereum and other cryptocurrencies are planned.

- Building - each building shows either an input(north side of the street) or an output(south side of the street), the size of the building depends on the value of the transaction
- Window - only displays on buildings when input and output are at least 1 BTC
- Roof - the colors symbolize if the input/output is spent(red) or unspent(green)
- Tree - symbolize that no fee is paid for the transaction

In the image below 9.34 you can see an example of a transaction visualized in Bitcoin City. It is difficult to estimate different transaction details by inspecting only the city and predict its value. So the user has the option to see the exact details of the transaction by clicking on a city and chose the option full view.

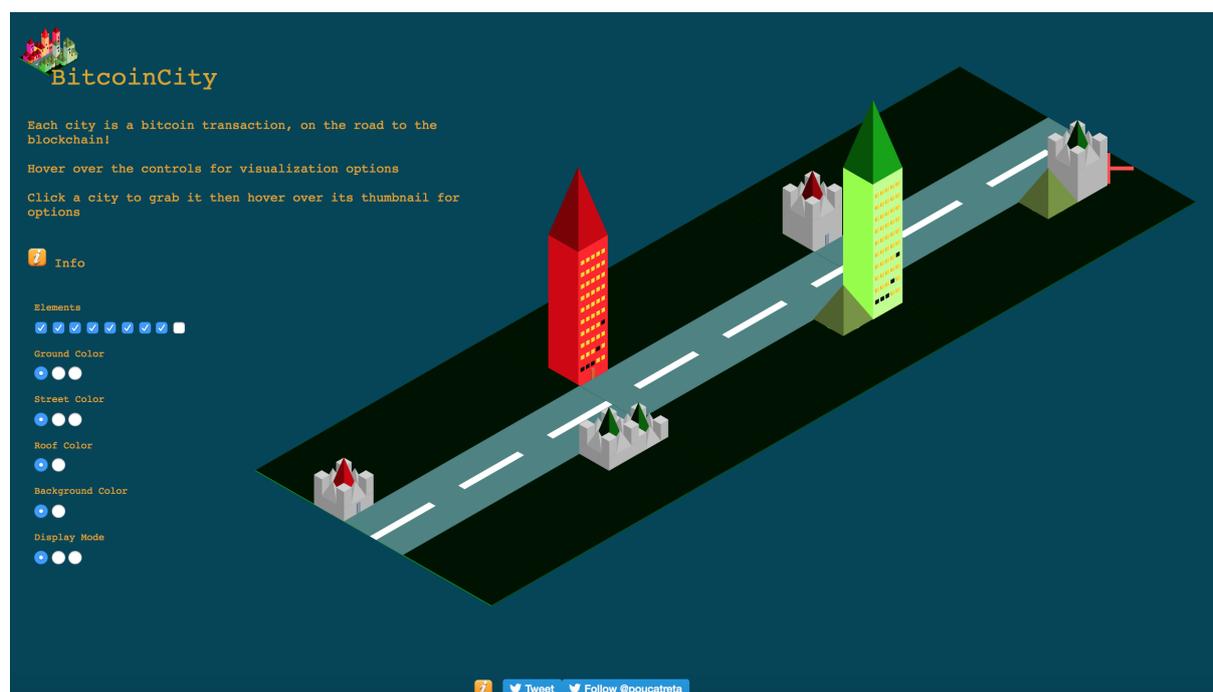


Figure 9.34: A Bitcoin transaction visualized in Bitcoin City

As user you have the possibility as well to chose the design of the city out of a selection of visualization options. You can switch trough different graphic settings like color of the street, background color or display mode.

9.4.1.5 Fiatleak

Fiatleak[15] is another application to visualize the trade with Bitcoins in realtime including geographical aspects based on the data from seven trade markets. With the integrated map it is possible to see from where money floats into Bitcoin and references to one of the world currencies which are mapped on the bottom of the map. Information about average Bitcoin price, transaction volume and network power consumption can be displayed 9.35.

9.4.2 Example 2: Ethereum transaction activity

Almost as well-known as Bitcoin is the cryptocurrency Ethereum. Although both can be categorized as cryptocurrencies, Ethereum extends its usage by enabling smart contracts to be written in the chain. Therefore, additional data is available that can be displayed. The mere amount of data available in the network is enourmous. A great overview of

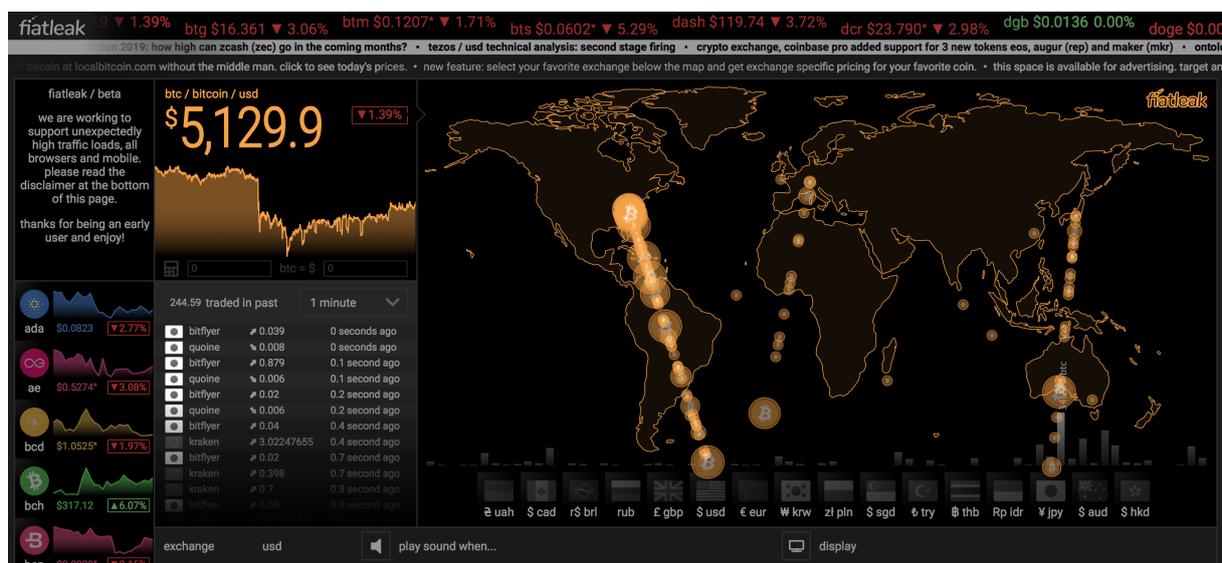


Figure 9.35: Watch the world currencies flow into BTC in realtime.

data is given by the website¹.

There are few working examples on the internet that try to visualize the data of Ethereum transactions and their content. Working examples will be discussed in the following sub-sections.

Each of these visualization tools offers the possibility to explore data within the ethereum network. Every tool has chosen a different depiction of the data and will therefore be explained in detail.

9.4.2.1 Ethviewer.live

Ethviewer.live [13] graphically displays the recent history of the publicly accessible Ethereum blockchain. More precisely, it shows the 24 most recent blocks of the blockchain as boxes. The boxes contain the current transactions as circles that sum up to a transaction pool for the according block. These transactions inside the block are waiting to be included in the block. The color of the transactions (circles) do also offer additional information, as a grey circle stands for simple transactions, blue circles represent transactions in which contracts are being created yellow circles stand for transactions in which contracts are invoked. Clicking on a single transaction opens a link to <https://etherscan.io> in which the detailed information about the transaction is displayed.

Additional information about the blocks is given in the header section of box. The color indicates whether a block is part of the main chain (green) or if the block has become a so-called uncle block (red). The fuel gauge in the header represents the amount of gas used. If the pointer of the gauge points at E (empty), it means that all the gas has been used. Contrary, the F (full) indicates that no gas has been used by now. This is the case when the block is empty.

9.4.2.2 Ethervis

Ethervis [11] allows users to paste a user-defined transaction address in order to further investigate to which other addresses it is linked. When hovering over a transaction node, the amount of ethers sent or received is displayed on the edges that are connecting the transaction addresses. Also, the edge adjusts its width depending on the amount transferred from one node to another.

¹<https://ethstats.net>



Figure 9.36: Blocks and transactions in Ethviewer



Figure 9.37: Transaction nodes in Etherscan

The transaction node entered at the beginning is colored in orange. Connected transactions are colored in blue, if there are all transactions determining a transaction address are loaded, the node is displayed in green. The tool has a great explorative touch by letting users click on any node which leads to all transactions connected to the clicked address. Etherscan is not only limited to personal addresses, but also lets users search for contract addresses. The usage of Etherscan is intuitive and lets users explore the connection between participants in the blockchain as well as connections to smart contracts.

9.4.2.3 Ethviewer.now.sh

Ethviewer [12] is another graphical explorer for the Ethereum blockchain. The information that can be extracted from the graphics are held rather simple, neglecting information about uncles and internal contract message calls. Nevertheless the visualization offers a great overview on how the blockchain is built.

The single transactions are shown as small colored squares. Green squares define contract executions, whereas the red squares stand for contract creations. Another interesting detail integrated in the visualization are the simple transactions in different greyscales. The darker the square of a simple transaction is, the higher is the actual value transferred in this transaction. Unfortunately there is no legend that indicates the exact values of

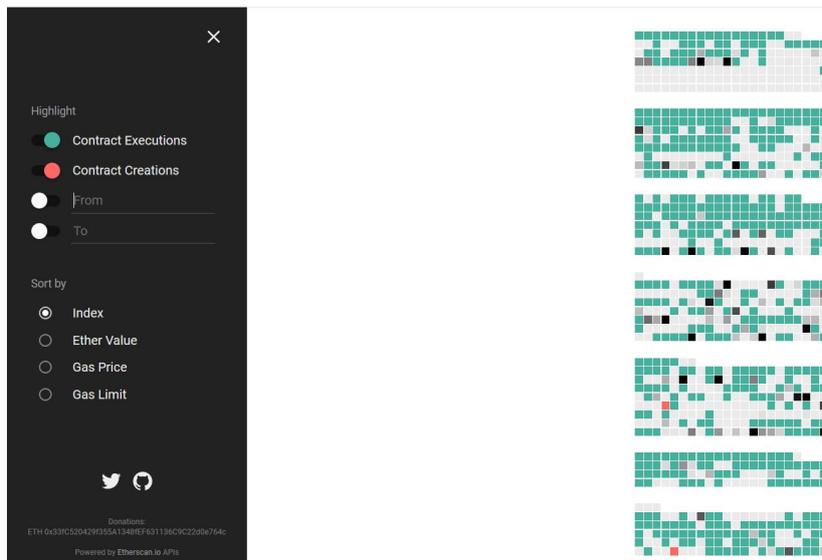


Figure 9.38: Transaction nodes in Etherscan

the greyscales, but by observation a black square indicates a transaction of roughly 15 or more ethers.

The squares are grouped in rectangles which represent a single block of the chain. Each consecutive block is appended at the top of the page.

9.4.2.4 Blockchain

Although Blockchain[8] is not as visually appealing as the other examples, the possibilities of filtering network data is very nuanced. There is a wide variety of filters that can be applied to search for transactions, blocks, addresses and calls. Sadly, the data is not processed into any visual outputs. Solely some attributes of the data are displayed over time and displayed as a bar chart or a line chart. For further visualizations of interest Blockchain could be very useful for future visualizations since the filtered results can be exported to a csv file.

9.4.2.5 Other visualizations

Even after intensive research for different applications of blockchain data visualization, the results imply that current visualizations focus on the transactional details. Although the mentioned visualizations all take a different approach to visualize transactions, we were hoping to find visualizations that visualize other data than transactional details.

Google integrated Ethereum in their BigQuery[1] service, allowing smart contract function analysis. The function analysis allows to compare smart contracts in terms of how they're enforced and might also uncover if they're being used for similar purposes. With the BigQuery tool it is also possible to find the most popular smart contracts. Nevertheless visualizations of smart contracts content are almost non-existent although the data is available.

9.5 Discussion

9.5.1 Benefits

First of all an important benefit is the availability of real-time data which assure that all the visualizations are up to date. Furthermore the visualizations can be used to explain and give the general public a first impression of the use and its operations of the Bitcoin

system. So visualizations make it easier to understand the linking between transactions by exploring a visualization than just looking at the raw data. For example the Blockchain 3D Explorer shows how we combine the transaction with a address graphs into one visualization to see from who to who Bitcoin floats and which address has a lot of transactions. Also an application like Fiatleak gives an insight of how much money is invested in Bitcoin divided by different countries. With BitBonkers you become an impressive overview of the frequency of Bitcoin transactions that are made every minute. Blockchain visualization also gives opportunities for the discovery of human and algorithmic behavioral patterns which complies with the interests of financial regulators, protocol designers and security analysts [10].

The list below shows an overview of the main benefits summarized:

- to provide a first introduction on the operation of cryptocurrency systems to the general public
- to get an overview of the frequency of use and the amount of money that floats into different cryptocurrencies
- up to date attribute of the visualizations
- to find and understand transaction patterns
- to evaluate and explore these patterns

All in all blockchain visualization is more or less a new research field and provides a lot of potential for future work what may also can be viewed as a benefit. Future work may include business models linked to blockchain-technology as for example trading goods or visualizing smart contract details in a comprehensible way. With new business ideas emerging, the possibility to visualize data inside the blockchain could potentially be used to create a wide variety of visualizations.

9.5.2 Challenges and Limitations

Although information visualization of blockchain data offers great benefits in exploring transactional patterns and anomalies within a network, blockchain visualizations has several limitations that may be hard to overcome in the future. Within the context of cryptocurrencies, the presented visualizations should be handled with caution, as unconfirmed transactions are often included in the visualization. The fact that neither of the presented visualizations did differentiate unconfirmed from confirmed transactions is a limitation to those visualizations and might display misleading information.

Moreover, most visualizations are displaying information in real-time. Even though this fosters the explorative aspect of the visualization, it happened quite often that the visualization looked too crowded to effectively search for patterns. Whenever movement was involved in the visualization of real-time data, the visualization frequently acted fidgety would not allow a proper overview.

Another challenge for blockchain visualization is that, although the concept of the blockchain is already a decade old, research and enterprises are still uncovering the potential of the blockchain technology for future products and applications. The possibilities of visualizing blockchain data is therefore limited to the current usages of the technology. In detail, a lot of visualizations for cryptocurrencies exist, whereas for other products based on blockchain no visualizations exist.

9.6 Summary and Conclusions

Simple blockchain data extraction tools can be divided in different categories. Very well-known tools such as Etherscan.io or Blockchain.info represent the first group: the so called blockchain explorer or blockchain parser. The second group are query support layer systems which push forward the satisfaction of the demand to query blockchain data the same way as we do databases. One goal is to have a system which includes almost real-time data and can be queried with SQL statements. The third category are tools which try to de-anonymize the blockchain. They usually include a blockchain parser and work with a clustering system based on heuristics in a second step.

Visualization proves itself as a effective tool for conveying information. To not lose the relations or create a wrong understanding it has to be carefully evaluated, what technique should be used to coherently map the data to convey as much information as precise as possible. Understanding Human Perception is a key factor in doing so. There exist a vast amount of sources on creating good visualizations. The main use of visualization in the Bitcoin world is to display transactions based on the provided real-time data. For these use a lot of different applications exist and help us to find and understand transaction patterns in the Bitcoin network. But Bitcoin is not the only cryptocurrency with existing visualizations for ethereum a small variety of visualizations exist. Unfortunately their focus is rather on transactional details than the content of smart contracts. Although a lot of effort has been put into those representations, a visualization which offers new perspectives about smart contracts could be interesting to explore.

To conclude, available graphical representations of blockchain data mainly focused on cryptocurrencies.

Bibliography

- [1] Google Cloud Analytics Products: "BigQuery Cloud Data Warehouse, 2011, [On-line] <https://cloud.google.com/bigquery/>, last visit: May 1,2019.
- [2] Bin Zhu, and Hsinchun Chen: "Communication-Garden System: Visualizing a computer-mediated communication process", *Decision Support Systems*, Volume 45, Issue 4, 2018, pp. 778-794.
- [3] Bitcoin City, [On-line] <http://bitcoincity.info/>, last visit: May 2019.
- [4] Bitnodes, [On-line] <https://bitnodes.earn.com>, last visit: May 2019.
- [5] Bit Bonkers, [On-line] <https://bitbonkers.com>, last visit: May 2019.
- [6] BlockchainSQL.io, [On-line] <http://blockchainsql.io/>, last visit: May 2019.
- [7] Blockchain 3D Explorer, [On-line] <https://blockchain3d.info/>, last visit: May 2019.
- [8] Blockchair, [On-line] <https://blockchair.com/>, last visit: May 2019.
- [9] Daisuke Matsuoka, Fumiaki Araki: "Survey on Scientific Data Visualization for Large-scale Simulations", 2011, JAMSTEC Report of Research and Development.
- [10] Dan McGinn, David Birch, David Akroyd, Miguel Molina-Solana, Yike Guo, and William J. Knottenbelt: *Visualizing Dynamic Bitcoin Transaction Patterns*, 2016.
- [11] Etherscan, [On-line] [www.Etherscan.com](http://www.etherscan.com), last visit: May 2019.
- [12] Etherscan, [On-line] [www.Etherscan.com](http://www.etherscan.com), last visit: May 2019.
- [13] Etherscan, [On-line] [www.Etherscan.com](http://www.etherscan.com), last visit: May 2019.
- [14] F. A. Pratama and K. Mutijarsa: *Query Support for Data Processing and Analysis on Ethereum Blockchain*, International Symposium on Electronics and Smart Devices (ISESD), Bandung, October 2018, pp. 1-5.
- [15] Fiatleak, [On-line] <https://fiatleak.com/>, last visit: May 2019.
- [16] Franco Muriel et al.: "Interactive Visualizations for Planning and Strategic Business Decisions in NFV-Enabled Networks", 2017.
- [17] Github: Znort987's blockparser, [On-line] <https://github.com/znort987/blockparser>, last visit: May 2019.
- [18] Imran Bashir: *Mastering Blockchain - Distributed Ledgers, decentralization and smart contracts explained*, 2017.
- [19] Jiawei Zhu, Peipeng Liu, Longtao He: *Mining Information on Bitcoin Network Data*, 2017.

- [20] Livio Pompianu: *Analysing blockchains and smartcontracts: tools and techniques*, 2018.
- [21] Michele Spagnuolo, Federico Maggi, and Stefano Zanero: *BitIodine: Extracting Intelligence from the Bitcoin Network*, 2014.
- [22] MongoDB: NoSQL database, [On-line] <https://www.mongodb.com/>, last visit: May 2019.
- [23] Tao-Hung Chang, and Davor Svetinovic: *Data Analysis of Digital Currency Networks: Namecoin Case Study*, 2016.
- [24] Walport, M.: *Distributed Ledger Technology: beyond block chain*, 2015.
- [25] Ward Matthew, Grinstein Georges, and Daniel Keim: *Interactive Data Visualization: Foundations, Techniques and Applications*, 2015, ISBN 9781482257373.
- [26] Yang Li, Kai Zheng, Ying Yan, Qi Liu, and Xiaofang Zhou: *EtherQL: A Query Layer for Blockchain System*, Soochow University, China, 2017.

Chapter 10

Evaluation and Comparison of Blockchain Consensus Algorithms

Joel Barmettler, Özgür Acar Güler, Marc Laville, Spasen Trendafilov

Abstract - While the public caught the interest in blockchains through cryptocurrencies like Bitcoin, the technology behind these currencies emerged to a general field of research around secure, decentralized and trustless computing system. Fundamentally, these systems are formed by nodes that all have a redundant copy of a shared data structure, in most cases an immutable blockchain. In order to update this data structure among all nodes consistently, the nodes run a consensus protocol that ensures safe and consistent modification of the distributed ledger among all nodes participating in the network. Over the years, a number of different consensus algorithms formed. This work discusses a few of the most influential, controversial or innovative consensus algorithms and established a comparison based on a selection of key features.

Contents

10.1 Introduction	279
10.2 Consensus	280
10.2.1 Different Consensus protocols	281
10.3 Byzantine Agreement	282
10.3.1 Practical Byzantine Fault Tolerance	283
10.3.2 Federated Byzantine Agreement	285
10.4 Proof of Work	287
10.5 Proof of X	289
10.5.1 Proof of Stake	289
10.5.2 Proof of Elapsed Time	294
10.6 Ouroboros	294
10.7 Open Representative Voting	297
10.8 Conclusion	300

10.1 Introduction

With the success of Bitcoin over the past years, the Blockchain technology has rapidly gained popularity among computer scientists as well as in the financial sector. Although the hype surrounding the cryptocurrencies has slowed down rapidly as the currencies lost in value, the technology behind them remains a fast growing scientific field.

In [61], a concept similar to blockchain was first described. The goal of [61] was to come up with a way to digitally time-stamp a document in a way that can not be manipulated and that does not require a service to keep track of the records [61].

In 2008, someone known under the pseudonym of "Satoshi Nakamoto" came up with the first concept of a blockchain as it is known today. In 2009, the concept was implemented and builds the core technology behind the cryptocurrency Bitcoin to this day.

A blockchain is a data structure that stores facts like transactions in an time-ordered manner without a central party. The fundamental concept of a blockchain and most Distributed Ledgers, a broader term for decentralized data structures, is creating a secure environment in which participants don't have to trust each other. While in classical computing systems, data is stored in large databases that are in complete control of whoever owns the dedicated server infrastructure, therefore relying on trust that nobody alters the data without the users knowledge, distributed ledgers eliminate all central authorities by decentralizing both the data and the data-manipulation algorithm. Data in a distributed ledger is stored redundantly throughout computers across the world, ensuring that no single person can alter the content of the ledger, making it a secure medium to permanently store data.

By using a decentralized system in which all participants store their own version of the blockchain, a blockchain is invulnerable to some risks a centralized system is vulnerable to. It is for example harder to delete or modify data if every node in the system stores a copy of the data, because to do so, every node would have to be attacked. In comparison with a centralized system, only a single point of failure (SPOF) would have to be successfully attacked. The same principle of risk reduction can be applied for technical failures such as hardware failures: if a single node fails, the data is still secured on all other nodes who carry a copy of the data.

Most ledgers are publicly visible and anyone can find and locate any facts stored in the data blocks. To add new data to the ledger, the computers that participate in the network have to validate whether the request is legit or not. In the case of Bitcoin, these computers, also called network-nodes, check for each new transaction request whether the person trying to spend money also has sufficient funds. If the nodes come to a consensus that the transaction is indeed valid, they all alter their local ledger state.

A distributed ledger can therefore be characterized by what data structure the ledger has and what consensus protocol the nodes are running. While the data structure of most distributed ledgers is the famous blockchain, the consensus algorithms are the reason why blockchains are challenging to engineer. Not only does the network need to find consensus among nodes that are distributed around the whole world, it also needs to deal with latency, corrupted data, network attacks and malicious nodes.

There are three different fields types of blockchains [18]: Permissionless blockchains, permissioned blockchains and private blockchains. While permissionless blockchains are open for anybody to invoke transactions on the blockchain, run a validating node or participate in the consensus finding mechanism, permissioned blockchains only allow limited and validated entities to run the network: Nodes are identified and controlled, transactions request often limited to known parties. Private blockchains are a subsection of permissioned blockchains with only one organization operating all parts of the blockchain, creating a single trust domain. Permissionless blockchains like Bitcoin, Ethereum or Litecoin represent the understanding of a traditional blockchain, while permissioned blockchains

like Ripple or private blockchains like Hyperledger have their own legitimate applications. The type of a blockchain defines which consensus algorithms may be suitable. Often, in private blockchains, there is a central list of a small number of nodes participating in the consensus protocol. The consensus algorithms can therefore assume that no malicious nodes can be spawned in high numbers, that there is low latency in the network, and that only a relatively small number of mostly trusted nodes are part of the consensus finding protocol. This looks different for permissionless blockchains: Everybody can participate in the network, implying that malicious nodes can be spawned in a large number, overflowing the network with carefully chosen information to hinder nodes from reaching consensus.

According to the Scalability-Trilemma [69], a general trade-off between decentralization, scalability and security exists. A well-scalable protocol can handle thousands of transaction requests per second. If it can resist against all known attacks as long as a fixed majority of nodes in the network are honest, it is seen as secure. Furthermore, if the network is backed by hundreds of nodes, distributed all over the world and potentially hosted by unknown parties, it can be called decentralized.

Each consensus algorithm known today seems to perfectly solve two of them while making a trade-off on the third. Thus, no consensus algorithm is superior: The trade-off leaves room for many different algorithm implementations, addressing different needs. While private blockchains are often highly centralized, they scale well and have a high security standard. Other consensus algorithms like Bitcoin prioritize decentralization and security, while failing on scalability.

10.2 Consensus

The rate of involved trust varies between blockchain types, but they are all distributed computing systems, sharing the need for a robust and stable ecosystem. In order to work properly, all nodes of the distributed system have to eventually reach consensus, even in the presence of failed or malicious nodes. There are two main tasks involved in finding consensus among distributed nodes that communicate peer-to-peer [28]. First, each node has to run a (deterministic) state machine (DSM) implementing the service the blockchain provides. Second, the nodes share their state using a consensus protocol such that each node performs the same actions on its state machine - they have to find consensus about what operation to apply on the current state, such that the overall network remains stable. The first step, the DSM, is an implementation of the famous Automata as explained in [55]: Each node has a state, and new transactions can change their state. In a functioning blockchain, all synchronized nodes are in the same state. Since the State Machine is deterministic, the same transaction will result in the same, new state for all nodes. Therefore, if all nodes were synchronized in the beginning, and with applying the same transaction to all nodes, the blockchain shall never be in an inconsistent state. While it is easy to have a common state among all nodes, the hard step is having a protocol that ensures that all nodes will always change their state in the same manner. This leads to the second step, the consensus protocol. It needs to follow a few properties: All nodes, or state replicas, need to agree on the same value in a non-trivial case, meaning they do not just always agree on a trivial state change, but rather on an output at least one of the nodes vote for.

Finding consensus in a distributed system with completely reliable actors is a trivial task. However, in real world scenarios there are plenty of possible faults, from losing connection to the other nodes over process crashes to sending duplicate messages. A reliable consensus protocol shall therefore overcome such faults, as long as the number of malicious nodes is not overwhelming.

Speaking of a distributed computer system in general, a fault-tolerant system is a system in which the overall reliability exceeds the reliability of the individual parts [33]. A traditional fault in a distributed computing environment is a computer, or node, sending wrong or missing data, which can be overcome by having a certain amount of data redundancy, allowing the system to detect errors and continue working in the absence of certain data. A fault in the system may or may not cause a failure of the system, depending on whether the fault can be properly handled. The rate to which a system is tolerant to faults without causing failure is called fault-tolerance.

Distributed systems are always dependable. The formal definition of dependability is given in [40]: "Dependability is quality of the delivered service such that reliance can justifiably be placed in this service". To increase the dependability of a system, one either avoids fault on a node level by increasing the node quality, or increase the fault-tolerance of the overall system. For permissionless blockchain networks, absolute fault avoidance is impossible: Even if all participating nodes are in perfect condition, the network still has to deal with malicious nodes and overall churn. Blockchains therefore focus on having highly robust consensus mechanisms that ensure that even with a relatively high amount of fault in the system, the network remains up and is running correctly.

For traditional distributed computing systems, such as cloud computing systems in which many computers work together to solve one common task, consensus algorithms like Paxos and Viewstamped Replication are used that allow a certain number of nodes to fail while still reaching consensus [33]. While Paxos is well suited for controlled environments, permissionless public blockchains can not function on such consensus protocols since nodes may intentionally become malicious and work against the common goal of reaching agreement. The traditional distributed-computing algorithms need to be expanded to become fault-tolerant against any arbitrary or even malicious behaviour.

10.2.1 Different Consensus protocols

Consensus algorithms in blockchain ecosystems can have different natures. Two extremes are characterized on a spectrum in terms of the basic principle which the protocols are based on. On one end of the spectrum there are protocols which are based only on computation to prove their validity. These protocols use a computational task to determine which node to entrust with the choice of the next operation on the blockchain. A node can qualify by solving a computational task faster than its rivaling nodes. Once a solution is found, the node can propose its solution to the network, where it will be validated. Bitcoins Proof of Work (PoW) consensus algorithm is a perfect example for a purely computation-based consensus algorithm. On the other extreme of the spectrum are consensus algorithms which are based only on communication. In communication-based algorithms each node gets to vote to decide which node will be allowed to perform an operation on the blockchain. Multiple rounds of voting may take place to reach consensus. For this system to work, the nodes have to assume that the other nodes who participate are allowed to vote.

There are many different consensus algorithms which combine or modify the concept of a purely competitive based with the concept of a purely communication-based protocol to create hybrid protocols with the goal to improve the performance and eliminate their respective drawbacks.

Over the next few chapters, a closer look will be taken at different consensus algorithms by analyzing how they work. The goal is to be able to compare the found characteristics of the different algorithms and put them in relation to one another. Among the wide variety of algorithms, a selection has been made to feature the most prominently used ones. In addition to that, a set of algorithms which take a different approach will be featured as well.

10.3 Byzantine Agreement

The blockchain problem is "to allow an arbitrary large network of processors to agree on the blockchain state under the assumption that the computation power of malicious processors is bounded" [46]. Assume that the network consists of a total of n processors, or nodes, out of which f are malicious and do not behave as expected. A blockchain only works if there is a protocol that ensures all honest nodes agree on the correct state of the blockchain when a new block of transactions is created as long as f is smaller than n by a known degree.

The literature describes a problem of reliable computer systems using a metaphor with Byzantine generals [44]. A group of Byzantine generals, each controlling a part of the Byzantine army, camp around an enemy city. The forthcoming battle may only be successful if all loyal generals attack at the same time, *e.g.* they have to find consensus about the attack time while only communicating via oral messages. However, some of the generals may be traitors, intentionally misinforming other generals about their attack intention. The loyal generals need to have an algorithm in place that ensures that all loyal generals agree on a common and correct plan of action, while dealing with traitors that do anything they want to hinder the loyal ones from reaching consensus.

Each general observes the enemy and communicates his attack/retreat recommendation to all other generals. Each general therefore has a set of opportunity, provided by all other generals, and bases the decision whether to attack/retreat on a previously defined protocol, like deciding based on the majority. While this works under ideal conditions (no traitors are in place) the algorithm fails as soon as traitors send different pieces of information to different generals: Some loyal generals may receive only "retreat" messages from the traitors, while the others receive "attack", preventing the loyal generals from finding consensus. A loyal general can not rely on the information given by a general directly, since it may be malicious. However, in order to find consensus, all loyal generals must have a set of information that will lead to a globally uniform attack/retreat decision. The problem of how generals communicate their message can be restricted such that everybody has the same set of information. Since in a computer environment, it can be assumed that every message is delivered correctly, it can be identified who sent it and absence can be detected. For simplicity reasons, whenever a message is absent, a default value, a *retreat*, is used. Split the generals into commanders and lieutenants, while under each commander, there are $n - 1$ lieutenants, where n is the number of generals in total. All loyal generals must obey the same order, and a loyal general must send the message in a way that ensures that all loyal generals trust his order. These two conditions are called the "interactive consistency conditions" [44].

Indeed, there is an algorithm that solves the Byzantine Agreements problem with tolerating at most f traitors with $n = 3f + 1$ generals in total [44]. The commander tells its $n - 1$ lieutenants about his plan, attack or retreat. For each lieutenant i , let v_i be the value general i received from the commander. Now, lieutenant i sends the value v_i to the $n - 2$ other lieutenants. This way, the message propagates redundant to all lieutenants, with each lieutenant not only receiving the commanders message directly but also as forwarded information from any other lieutenant. For each i and each $j \neq i$, v_i is the value lieutenant i received from lieutenant j . This message delivery network creates a state of complete information in which every node knows what node sent which message to whom. With the information about what message node i sent to all the other nodes, each node can decide whether node i is potentially malicious, *e.g.* sent different messages to different users.

With using the majority value for all received v_i, \dots, v_{n-1} , it can be guaranteed that each general has sufficient information, and therefore reach consensus among all loyal generals. To this point, it is still unclear how many nodes may be malicious while consensus can

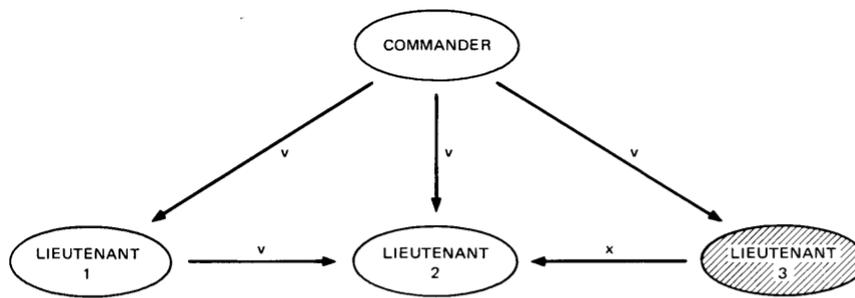


Fig. 3. Algorithm OM(1); Lieutenant 3 a traitor.

Fig. 4. Algorithm OM(1); the commander a traitor.

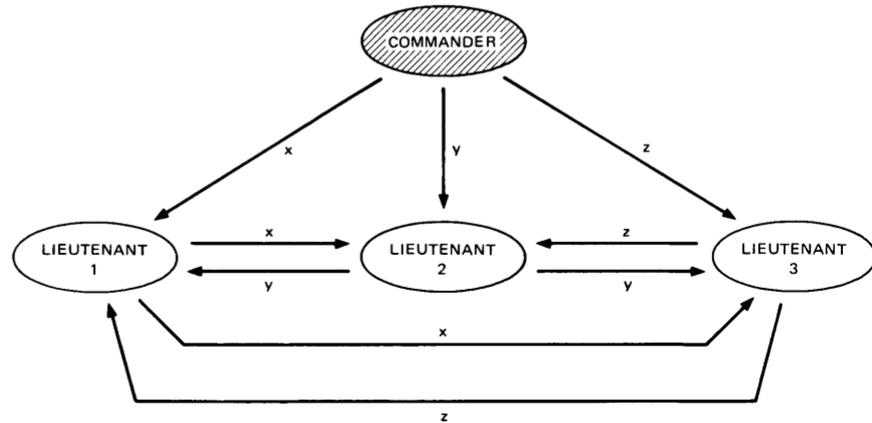


Figure 10.1: Byzantine generals problem with three loyal and one malicious party. If a lieutenant is a traitor (first image), the other lieutenants easily recognize him since he provides values different from the commander (x instead of v). If the commander is a traitor (second image), he is identified since the lieutenants get different messages from the other lieutenants about what the commander said (x , y or z). [44]

still be reached. Clearly, it has to be less than 50%, since otherwise the malicious nodes could reach consensus on their own. Moreover, it can only guarantee that if less than $1/3$ of the nodes are malicious. If one third were malicious, and there was a 50/50 disagreement between the honest parties, or if by any reason the honest parties are split and unable to talk to each other, the $1/3$ malicious nodes could reach $2/3$ of the population votes with one half of the honest nodes, and $2/3$ with the other half, destroying consensus. This implies that less than $1/3$ of the population may ever be dishonest.

However, [44] also states in the conclusion that the proposed algorithm is expensive in both time and number of messages sent, making it an inappropriate choice for decentralized systems that invoke many nodes that participate in the consensus finding process. Traditional Byzantine agreement is therefore heavily bandwidth- and latency-limited, with having exponential communication complexity with respect to the number of participating nodes [46]. Byzantine agreement further suffers from the problem of Sybil attacks [41], where nodes act as multiple instances and therefore increase their voting weight in the system. Such frauds are typically overcome by having a central list of participating nodes with a central authority that decide whether a new node can register itself for participation in the consensus finding mechanism or not.

10.3.1 Practical Byzantine Fault Tolerance

In the year 1999, a new and more practical algorithm based on Byzantine Agreement was developed: Practical Byzantine Fault Tolerance (pBFT) [49]. The fault-tolerance remains at $m = 3f + 1$, where f are malicious nodes, with guaranteeing liveness and safety in the network. While safety ensures that all operations are either processed or reverted by all nodes together, liveness guarantees that the network won't stop processing

new transactions even when the elected leader is Byzantine, e.g. behaving arbitrary or malicious.

The pBFT protocol is used by several modern cryptocurrencies, including NEO, an Asian competitor to Ethereum. While NEO made some modifications to the algorithm, they follow pBFT to a large extent but with using different naming conventions and additional features like delegation. For simplicity reasons, the classical pBFT naming conventions are used.

pBFT distinguishes two types of nodes: primary nodes and backup nodes. A single primary node acts as the leader, while all other backup nodes can switch out the current primary if it seems Byzantine through a view change mechanism. In each view, only one node is the primary, and all nodes seek for consensus for one single view. If consensus is found, the view ends and a new primary is chosen.

pBFT ensures integrity and authenticity of any message sent in the network via applying a cryptographic hash to each message sent, including the message and the sender nodes signature. Furthermore, all new requests are numbered sequentially using a timestamp such that each request can only be executed once and in chronological order.

Every node that participates in the consensus protocol has a record table including the current consensus status. Each view has its own record table, while a view is one process of finding consensus. If nodes reach consensus right away, only one view per set of transactions is needed, but multiple views may be needed if consensus finding turns out to be difficult. Views are labeled sequentially, starting from 0, and bring a form of synchronization into the network. Nodes always search consensus in the same view, and if consensus can not be reached for too long, the view is discarded and a new one is created.

Besides the views, the nodes are numbered as well, starting from 0 to $n - 1$, with n being the total number of nodes that are included in the consensus mechanism. In each view, one node is chosen to be the primary node. At the end of the view, the view number is increased and a new view is generated if no consensus could be found, or a new block is generated if consensus was found, e.g. $n - f$ signatures of backup nodes were reached. If a new block is created, the view number drops to 0 again and a fresh view is started.

When a new transaction is being started, the primary node broadcasts the transaction to the entire network, including the sequence number, the message as well as digest [35], [25]. This phase is called the pre-prepare phase. The backup nodes receive and store these transaction data in their memory if the signature is valid and the message has not yet been seen. Before sending a response, backup nodes validate the transactions and abandon the consensus protocol if invalidity is found, aborting the current view and creating a new one. The view also changes if no consensus was found for a too long period of time. The backup nodes respond to a valid transaction with a broadcasted prepare statement, showing that they will be prepared as soon as they received at least $2f$ other nodes respond to the same transaction with a prepare message as well. When $2f + 1$ non-Byzantine nodes are prepared, the network has a committed state. All committed nodes then broadcast the "commit" message, and as soon as node i has received again $2f + 1$ such commit messages, node i is committed-local. A network that contains committed-local nodes will always become committed as well, ensuring that the transaction will be accepted. The view is finished, consensus is reached.

pBFT is one of the first usable Byzantine Agreement algorithm, ensuring consensus despite Byzantine failure of nodes. It reaches consensus fast and energy efficient, while having decoupled trust from resource ownership. However, all nodes involved must agree on a list of known participants in the network that they include in information distribution. Furthermore, if anybody could join this list, attackers could join multiple times in so-called Sybil attacks, representing many nodes at once and increasing their power inside the network. Furthermore, even though pBFT reduces the bandwidth complexity from

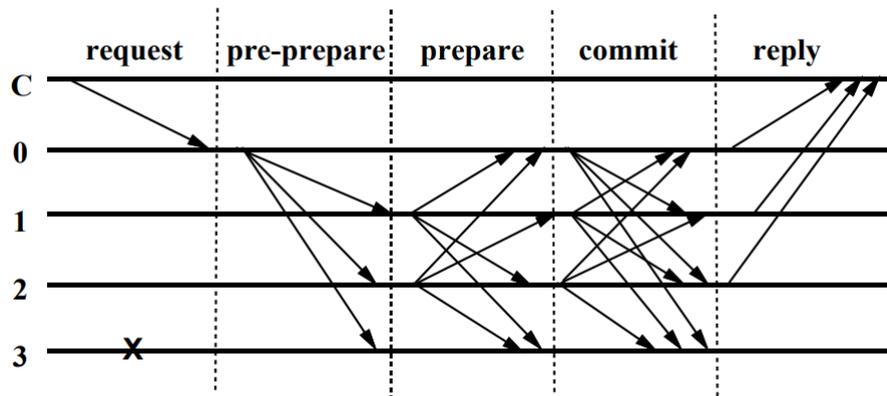


Figure 10.2: Message broadcasting between Primary (C) and backup (0 - 3) nodes in pBFT, with node 3 losing connection in the process. [49]

exponential to quadratic, a distributed network with a large number of nodes running on pBFT would still quickly run into scalability issues.

Today, blockchains like NEO that are running on practical Byzantine Fault Tolerance Consensus algorithms are highly centralized, with NEO only having seven consensus nodes in total and five of them being run by the NEO foundation itself [52]. Even though NEO offers plans to include new consensus nodes run by third parties in the future, the limits of pBFT will hinder any blockchain based on this consensus algorithm to reach a true decentralized state.

10.3.2 Federated Byzantine Agreement

The Stellar network uses its own adaptation of Byzantine Agreement called the Federated Byzantine Agreement (FBA). While the consensus mechanism is similar to pBFT, Stellar adds quorum slices to the protocol [59]. A quorum is a set of nodes that is large enough to reach agreement in the network. Quorum slices are a subset of a quorum that can convince a node to agree to a certain opinion. Quorum slices are typically significantly smaller than a quorum. If all nodes belong to the same quorum slice, FBA would be the same as non-federated Byzantine Agreement. While normal Byzantine Agreement requires all nodes to agree on the state of the network where each node must be known and verified, in FBA nodes choose their own quorum slices which are the nodes they trust. Companies, Banks or individuals running nodes on the Stellar network can manually choose, based on personal interest or personal feelings of trust, whose institutions they want to trust and therefore add to their quorum.

In a good network state, quorum slices overlap and form quorum intersections. These overlaps are necessary to ensure that no two disjoint quorums can both reach consensus within each other, but a different consensus throughout the quorums: Quorum *A* could agree on state *X*, while Quorum *B* agrees on state *Y*. Disjoint quorums can therefore undermine consensus. Nodes must choose what quorum slice they belong to with ensuring to not violate quorum intersection. Nodes must further ensure that slices remain large enough and that the nodes it already contains have something to lose when lying in the consensus finding process.

In FBA, nodes in a quorum slice wait for the vast majority of other nodes in the same slice to agree on a state before considering it settled. Furthermore, these nodes also rely on other nodes which they consider important that may lie in other quorum slices, allowing the network to only reach consensus if the majority of the most trusted nodes agree on the next state. Network-wide quorums arise from individual decisions made by

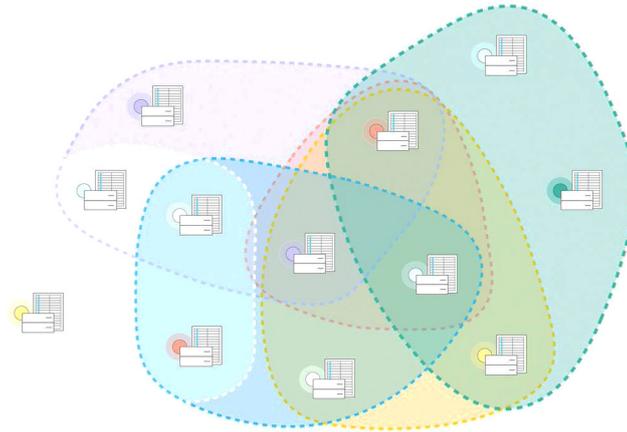


Figure 10.3: *Visualization of Stellar quorum slices among trusted nodes, run by different corporations. [59]*

the nodes, while no node has complete knowledge about the whole network, while still reaching consensus network-wide.

Each node has a set of trusted parties. As soon as these parties agree on a new state of the blockchain, the node accepts the state himself as well, reaching consensus quickly in a quorum slice. Each node that depends on other nodes can also be the dependant for other nodes as well. Through this concept, peer pressure arises: A normal node does not fully commit to decision but rather states its opinion. When enough of the other nodes in the same quorum slice formulate a different opinion, the peer pressure forces the node to accept this opinion as well if all quorum slices of the node have an other opinion. An opinion can never be changed on self behalf but only on peer pressure.

In federated BFT [22], the voting process for a new decision consists of 4 phases: Initial Voting, acceptance, ratification and confirmation. In the initial voting phase, each node specifies its personal opinion about what is the correct vote. A set opinion can not be changed on self-behalf, but a node can change its opinion if enough of the trusted nodes in its quorum slice have another valid opinion. In the acceptance state, each node make a final decision based on other nodes opinions. A node only accepts a decision if it is not contradicting with its own opinion, or if a vast majority of trusted nodes voted for the other opinion. In the following ratification phase, all nodes from a quorum slice accept one statement. Finally, the confirmation phase is a network-wide agreement on the same decision. It is reached if a sufficient amount of accepting statements is received. The confirmation phase broadcasts the confirmation decision over the network, eventually leading still unsure nodes to a decision.

Consensus within a quorum slice is found via a normal Byzantine Agreement such as pBFT, with using PoW to prevent Sybil attacks and securely splitting the quorum slices into roughly equally powerful portions. A committee combines the quorum slices votes into a new block in the blockchain.

Since quorum slices come to agreement on their own, the protocol has to overcome the risk of liveness: nodes or slices blocking the whole quorum from finding agreement. The protocol neutralizes blocked statements when they risk blocking the consensus finding process using ballots, referendums to the values being voted on. Each node can vote on either committing or aborting any ballot. If a quorum slices is stuck, the nodes can vote on "commit value X" or "abort value Y". Committing a value will automatically result in consensus, while aborting a value leaves the room open for a new ballot but with less opinions.

According to Stellar's own whitepaper, FBA is the only provably safe consensus algorithm that provides Decentralized Control where everybody can participate in the network without a central authority, low latency that finds consensus within all nodes in at most a few seconds, flexible trust where nodes can decide on their own which parties they trust, and finally asymptotic security that allows for tweaking the protocol to protect against third parties with large computer power.

The Stellar Consensus Protocol (SCP) addresses the largest downsides of normal pBFT. Since anybody can decide which parties to trust, SCP not just ensures flexible trust but also a much lower latency and asymptotic security since quorum slices are a magnitude smaller than the set of all nodes. Anybody can join the network and choose its own trusted nodes, making SCP a permissionless blockchain.

10.4 Proof of Work

As mentioned in section 10.2.1, PoW is a purely computation-based consensus algorithm. This means that a node has to prove that it "worked". The work done in this case is computational work for which resources have to be invested. This work is called mining in a PoW algorithm. A way to ensure that a node has done work is to pose a problem which is proven to not have any shortcuts to it. This way it is possible to estimate an average amount of tries to solve the computational problem of a given difficulty. With this, one can assume on average how much effort in form of computational power has to be invested so solve the problem. Methods for encryption are perfectly suited to be used for such a problem, since their whole purpose is to encrypt something in a manner which requires an extraordinary amount of computational power to solve while not offering known possibilities to shorten this process. Another key feature of those problems, aside of being hard to solve, is that they have to be easy to validate. As a freshly found solution to the problem is sent to the other nodes, they have to be able to validate the found solution easily. Since the hash inversion is not a decision problem, one can not talk about np-completeness, but the basic principle of "hard to solve, easy to verify" is similar. More specifically the miners need find a solution to a cryptographic hash function of the following form:

$$H(b.n) < d \quad (10.1)$$

where n is the solution nonce which has to be found. Transactions are represented by b which get concatenated to the nonce. H is the hash function of the respective blockchain. In the case of Bitcoin, SHA-256 is used [43]. There is a threshold d which depends on the difficulty of the problem. In a PoW blockchain, all miners try to solve the hash problem given the current latest block, starting with the genesis block. Once a miner found a solution, a block is added to the blockchain and the solution is broadcasted to the network. When the other miners receive a proposed solution which contains more accumulated computational work done than their current blockchain, they don't search for solutions to their problem anymore but try to validate the received solution. If they agree with the solution, they immediately start mining the next block on top of the updated chain.

Forks describe the situation when a blockchain is split into two leading chains. The blockchain now exists as two different versions in the network. The first part up to the fork is identical among the network. After the fork, there are two versions with valid solutions to the last common block present in the network. Such a fork can occur if two miners find a valid block at approximately the same time and broadcast it to the network. Because of the delayed information propagation in a blockchain network, some nodes will validate chain A and some nodes will validate chain B . A forked chain is a situation that wants to get resolved back into one single accepted chain. In a PoW algorithm, the

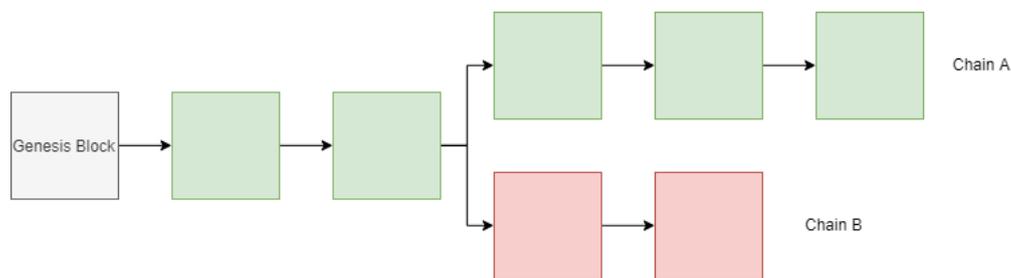


Figure 10.4: *The main chain is represented by the green blocks. The red blocks represent a two block fork which eventually got replaced by chain A. [64]*

longest chain always wins and this can be used to resolve the forks. When a miner starts mining for a new block on top of chain A it puts its computational power behind this chain. Eventually, one of the miners will find a new block and send its version to the network. If this happens to a miner who is appending to chain A, this chain will contain one block more than chain B where no additional block is found yet. When a miner who accepted chain B receives chain A with a block more, it will accept the longer chain as the correct one and will start mining on top of chain A. It is possible among the nodes which accept chain A that a new block is found at the same time as a new block is found among the nodes who accept chain B. This case is called a two block fork. This situation is feasible, but because the possibility to find a new block has only a certain chance, the probability of both chains growing at the same rate will decrease exponentially and one chain will eventually come out on top. In practice, the longest chain is implemented as the chain with the most amount of computational work put in to it. Each node can calculate the computational effort required to build the chain all the way back to the genesis block and compare different chains this way.

Computational power represents a resource which can be used to append a block to the blockchain. Naturally, miners need to be incentivized to spend computational resources. Most commonly the miner can reward himself with an amount of a cryptocurrency defined in the consensus protocol by adding a block to the chain. The amount of resources spent combined with the consensus protocol is the core part to protect against attacks and manipulation attempts. If a miner tries to reward himself with an amount of coins surpassing the amount defined by the shared consensus protocol and then invests all the resources to find a new block, the other nodes will reject the block and all the resources would have been wasted.

Despite the consensus protocols, blockchains can still be vulnerable to different attacks. The double spending problem occurs if a specific coin of a currency is spent multiple times, which can happen if a fork occurs. The currency can be spent for each branch of the fork. To prevent this, usually a certain number of confirmations is awaited to minimize the risk of double spending [8]. To spend a coin a second time on a fork of a PoW blockchain, one has to possess more computational power than every node who works on the main chain summed together such that a deliberately created fork can be validated. If the attacker does not have 51% of the computational power, the work put in to create the fork will not be rewarded therefore, rendering the attempt useless. Such an attack is called a 51% attack [57].

A 51% attack is a case of a Sybil attack. The goal of a Sybil attack is to create a large enough number of identities to gain a large enough influence to dominate the system [41]. In the case of PoW, the influence is represented by computational power.

By attempting a selfish mining attack on a PoW blockchain, the 51% threshold can be lowered down to 25% [57]. When attempting selfish mining, the attacker withholds found blocks to accumulate their advantage over the rest of the miners. By selectively releasing

the found blocks, the rest of the system is unable to catch up and is constantly wasting resources on blocks which in reality are already stale [6].

Compared to other consensus algorithms, PoW protocols usually have a smaller throughput. The throughput of Bitcoin stands at a maximum of 7 transactions per second [29]. As [4] show in their work, a throughput of more than 60 transactions per second can be achieved by adjusting the block size and interval.

10.5 Proof of X

Many other consensus algorithms emerged after the PoW algorithm. All of these "Proof of X" (PoX) consensus algorithms try to eliminate the problems of PoW by introducing a variation to Nakamoto's original protocol. One of the more popular introduced alternatives is the Proof of Stake (PoS) algorithm, which is not entirely problem-free. Similar to PoW, many alternatives to PoS have been published trying to eliminate its problems.

Thus, it's only natural to compare such PoX algorithms to either PoW or PoS and to examine how they have improved over them, which vulnerabilities and problems could not be eliminated and which new vulnerabilities have arisen with the new proposed consensus algorithms. This will be done in this section, but to get a better understanding of these consensus algorithms, PoS needs to be introduced and compared to PoW in a first step.

10.5.1 Proof of Stake

The goal of PoS is the same as of PoW: To reach a consensus over who is allowed to add the next block to the chain so that the state of the distributed ledger is identical across all nodes [63]. However, as seen in the previous section, the PoW algorithm has many underlying issues. The main issue that the PoS Algorithm is striving to solve is the excessive usage of resources due to the computation-bound principle of the PoW algorithm - that is to say, due to the energy consumption through mining hardware [24][64].

A user in a PoW-based blockchain is *mining* the next block by allocating processing power. Similarly, a user in a PoS-based blockchain is *minting* the next block by using, depending on the implementation, some or all of his own cryptocurrency [8][64]. The amount of cryptocurrency the user is willing to use for that purpose is called *stake* [64].

In PoW, the first user who solves the given inequality is the one who appends the next block to the blockchain. The higher the processing power or the more processing hardware a user possesses, the higher the chances of him mining the next block [64]. In contrast, in PoS the user who appends the next block is randomly selected. The chances of being selected is proportional to the amount of stake (in relation to the total amount of all cryptocurrency in the chain) the user possesses. If a user possesses 1% of all the circulating cryptocurrency in the blockchain, then his chances of minting the next block are 1% [8][63]. The PoS inequality that needs to be solved by minting can usually be generalized as the following formula:

$$h(a_1, \dots, a_n) < s(b_1, \dots, b_n) * d \quad (10.2)$$

where

- $h(\cdot)$: Hashing function,
- a_1, \dots, a_n : Input arguments for the hashing function,
- $s(\cdot)$: Some sort of a function that involves the stake,
- b_1, \dots, b_n : Input arguments for the stake function,
- d : Difficulty constant.

For the simplest case, assume the function $s(\cdot)$ would return the stake only. Since d is a constant, the difficulty of solving the inequality would depend on the amount of the stake. PPCoin, one of the first or even the first blockchain to introduce the PoS consensus algorithm [19][64], uses a concrete example of (10.2) [8][34][64]:

$$h(B_{previous}, A, time) < s(bal(A), age(A)) * d \quad (10.3)$$

where

- $B_{Previous}$: Some data of the previous block,
- A : Address of the stakeholder,
- $time$: Current time in seconds; usually in UTC,
- $bal(A)$: Balance in the stakeholder address,
- $age(A)$: Coin age.

Firstly, it is apparent that $time$ is an additional factor that influences this concrete inequality. Secondly, it is important to understand that the time unit being seconds restricts the hashing attempt of a minter to 1 attempt per second. A certain deviation interval is set for UTC. For example, $time$ can deviate ± 1 hour from the current UTC time. Since all the other parameters of $h(\cdot)$ are not changed as often, this would limit the amount of possible arguments of the hash function to 7200 [8], making the PoS consensus algorithm effectively more power efficient than the PoW algorithm due to the bound domain space [62].

It is noteworthy that the $time$ parameter could lead to a "Timedrift Exploit", where an attacker could calculate the hashes into the future to predict the probability of a certain user to mint the next block, effectively making it a long-range attack (See Section 10.5.1.2). However, this issue has been already solved. For more details please refer to [34]. Another detail of PPCoin is the initial use of PoW algorithm to distribute the first coins. Using PoS from start on without an Initial Public Offering (IPO) would not only increase the chances of a Sybil attack but it would also have economical impacts, like empowering early individual participants with a first-mover advantage [62].

10.5.1.1 Comparison to PoW

Both algorithms provide an incentive in form of cryptocurrency to motivate the users to participate in the consensus process [8][26]. However, since the domain space of the hashing function in PoW is infinite, the mining process becomes an "arms race": Each individual is competing against each other by trying to find ways to solve the inequality in a faster way [8]. While the choice of a memory-hard hash function can lower the cost-advantage of ASIC's [64], the computation-bound design principle of the PoW algorithm cannot be fully eliminated. On the other hand, that computation-bound design is responsible for the strong security and resistance against eventual malicious attacks. Therefore, the trade-off between the two algorithms occur in terms of power-efficiency vs security. More particularly, the security of PoS algorithms is by default so poor [47] (See Section 10.5.1.2) that it is necessary to come up with multiple ways of securing the algorithm against malicious attacks. Since almost no computational power is needed, the chances of having centralized resources diminishes in PoS. The decentralization (as defined in [69]) of PoS is therefore improved over PoW. At the same time, the disengagement from computational power allows for a faster throughput. In terms of performance and scalability, PoS implementations can vary widely. While the hybrid PoW/PoS algorithm of PPCoin has an estimated throughput of 8 transactions per second [54] in comparison to Bitcoins

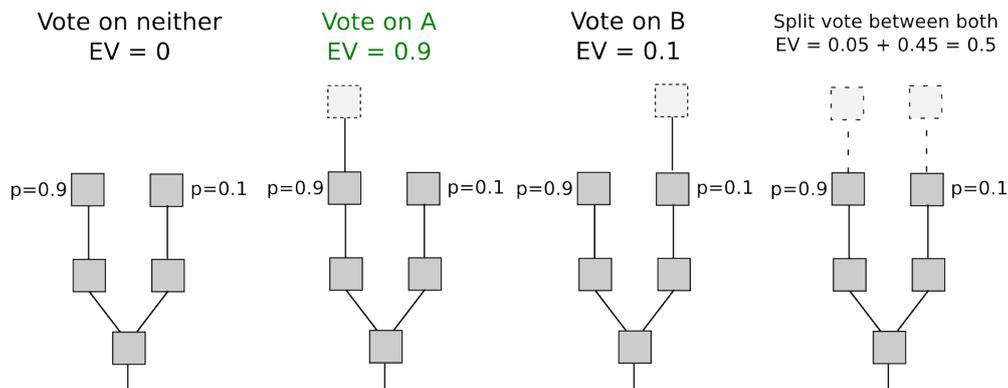


Figure 10.5: Shown are the 4 possible scenarios how a miner could act in case of a fork in a PoW system. The Expected Value "EV" changes depending on the miners choice. The green marked one is the rational choice. [68]

7 transactions per second, Ouroboros can achieve a throughput of 256.7 transactions per second (See Section 10.6). While PoS shows some improvements over PoW in regards to scalability, better scalability and security at cost of centralization can be achieved by a Delegated Proof of Stake (DPoS) algorithm (See Section 10.5.1.3).

10.5.1.2 Problem and Vulnerability comparisons between PoW and PoS

Both algorithms are potentially vulnerable to Sybil and 51% attacks. While the resistance to Sybil attacks in PoW comes from the needed computation power, the resistance against such an attack for a PoS system comes from an economical perspective: stakeholders with a high amount of stake are interested in keeping the network secure, since they don't want to sabotage themselves by attacking the system and risking lowering the value of their stake [34].

There are multiple ways to perform a double spending attack. The first method is with a long-range attack. If a PoS System is new, not yet established and the genesis block has just been created, a long-range attack can occur even if a stakeholder owns 1% of the total available coins [66]. At later stages, a higher stake amount might be needed [34]. In a first step, it is possible for an attacker to start minting on a forked new chain that is not the main chain. He can then spend some of his coins in the main chain and start minting on top of the forked chain in which he didn't spend the coins. In a next step, he can start minting into the future by executing a lot of calculations on the forked chain. Since he still owns 1%, the attacker will be faster in generating new blocks than the other minters in the main chain. When his chain gets validated, he will be able to spend his coins a second time [2][66][34]. There are multiple solutions to this problem, PPCoin implements two solutions to solve this problem. First, the input argument $B_{Previous}$ for the hashing function changes every 6 hours to make it impossible to foresee the future [34]. This solves the problem for both phases, when the chain is not yet established and when the chain is already established. And second (an indirect solution), PPCoin started using PoW and then slowly switched to PoS [62]. This especially solves the 1% attack. The second solution also solves the initial distribution problem of PoS, where some stakeholders would have too much influence over the system due to the first-mover advantage. Starting with a PoW algorithm is more competitive and certain miners do not possess more influence than the others [8][62].

The second way of performing a double spending attack is tied to the "nothing-at-stake" problem. Since there is only a limited amount of possible inputs for the hashing function of the PoS algorithm, forks are more likely to happen [34]. In case of a fork, a rational individual in PoW would continue to mine on only one forked chain without splitting its computation power, since that decision leads to the highest Expected Value (EV), as

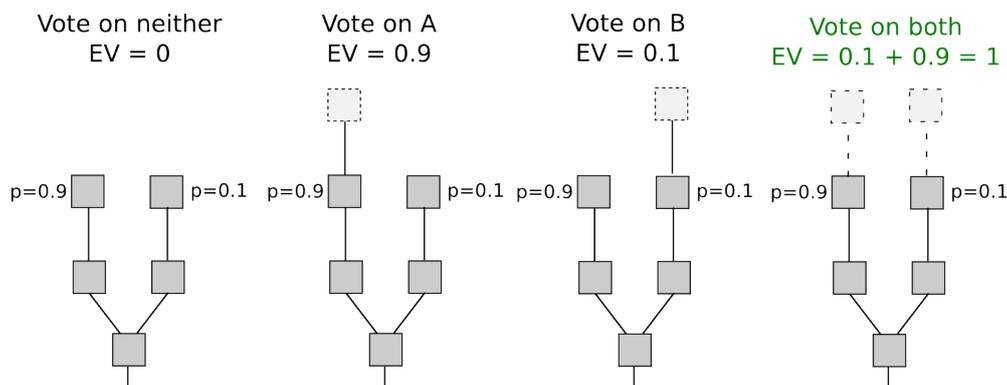


Figure 10.6: The same scenarios as in figure 10.5, but this time in case of a fork in a PoS System and with the highest Expected Value on the 4th scenario. [68]

seen in figure 10.5 [68]. In case of a fork in a PoS system however, a minter can decide to participate on any arbitrary number of forks without any penalty, since PoS is not computationally demanding and since there are no other (monetary) punishments. As seen in figure 10.6 [68], the minter increases his chances of being on the chain that will be chosen as the main chain in the future. The stakeholders EV increases accordingly. In this context, if a stakeholder decides to participate in a fork, this participation can be interpreted as a "vote" [26]. If enough participants act rationally by trying to increase their EV, they will vote for multiple chains and since they have voted for multiple chains, no consensus can be achieved [26][34][68]. Not only does it disrupt reaching the consensus, but this problem also facilitates a double spending attack. Similarly to the first method, a stakeholder with enough influence can effectively start minting at two forked chains, spend his coins at one of the chains and then leave that chain to mint on the other forked chain where the coins are unspent. Since that increases the probability of generating a longer chain on the second fork, the attack might succeed and the attacker could spend his coins a second time [39]. The nothing-at-stake problem can be easily solved by a DPoS algorithm [34][64]. See Section 10.5.1.3 for further details.

The third method to double spend is enabled through a bribe attack. For more details refer to [34]. A selfish mining attack seems to be possible in both [30][38]. In PoW, the selfish mining can waste the resources of other miners. Since there is no heavy computation involved in PoS, this is a non-issue. Selfish mining is used to increase the attackers own reward relative to the other participants rewards. This also applies to PoS and the result can even be worse in PoS systems [30]. Depending on the implementation, a grinding attack can also occur in a PoS System. In case of PPCoin, a minter might try to use different input arguments in such a way that the likelihood of him being selected increases [68]. This list of vulnerabilities is not complete. The amount of security issues with a standard PoS algorithm is large. For more types of attacks, refer to [2], [8] and [34].

10.5.1.3 Delegated Proof of Stake

Many of the sources used in this section are not official or not formal, as there don't exist a lot of them. Some of the issues of PoS can be overcome by a DPoS consensus algorithm while keeping the power efficiency achieved by PoS [11]. DPoS is based on a delegates system. Instead of minting by themselves, the stakeholders of a DPoS System can vote to select delegates who then will mint the new block, effectively centralizing the block generation. A stakeholder's voting power is proportional to their amount of stake. Since the delegated users have (temporarily) all the control over the chain, there would be nothing that would hinder them from misusing the nothing-at-stake problem from the PoS algorithm to double spend [8]. That's why it's especially necessary in DPoS to implement a method to punish users that are misbehaving. Casper for example, the

soon to be implemented PoS protocol of the Ethereum blockchain, punishes users who are trying to mint on multiple blocks by removing the malicious users complete stake [64]. In BitShares, a DPoS based system, the misbehaving block-producers can simply be "voted out" of the system [11].

The DPoS-based blockchain BitShares will be used as a concrete example. Based on self-governing democratic principles, it uses a *witnesses* and *delegates* system [11]. Both, the witnesses and delegates (also called committee) need to be selected by the stakeholders (the *workers* need to be selected too, but are of no relevance for this comparison) [11]. The amount of the witnesses and delegates is also determined by the stakeholders, depending on the wished decentralization level [10]. The more stake a stakeholder possesses, the more weight is assigned to his votes and he can split his voting power amongst the witnesses and delegates. It's possible for a stakeholder to proxy there votes instead of directly choosing the witnesses/delegates, meaning they can lend their voting-power to another stakeholder that will then perform the voting for them. Witnesses are the block-producers of the blockchain and they get rewarded for their work [11]. They are socially pressured to distribute as much of their rewards as possible to the voters, otherwise they would be replaced by other witnesses [65]. The elected witnesses are sorted in an ordered list. A time slot gets assigned to each witness during which they have to produce the next block. After the end of the list is reached, the witnesses get shuffled into a different order and the block-producing is repeated again and again. The committee members on the other hand are responsible for signing the blocks produced by the witnesses and changing the parameters of the blockchain, like block size and witness fee [10][11]. They do not get rewarded for their work [10].

The scalability and security are improved by increasing the risk of centralization [11][53]. In theory, according to the inventor of the DPoS algorithm and inventor of the BitShares and EOS protocol (also DPoS based), Daniel Larimer, it should be possible to reach a throughput of 10'000 transactions per second without optimization, 100'000 transactions per second with optimization [11] in BitShares and even 1'000'000 transactions per second in EOS [31]. In practice as of now, the all time high of BitShares in a test-environment was 3328.33 transactions per second [5] (claim could not be verified as source is not public) and 3996 transactions per second of EOS on a live environment. EOS is currently moving between 50 - 90 transactions per second [20] while BitShares is moving between 7 - 10 transactions per second [9]. The throughput can potentially reach higher numbers compared to many PoW and PoS implementations. If such high amounts are sustainable over a longer period of time in a real environment or if a higher throughputs can be reached in practice has not yet been proven.

Further on security, the nothing-at-stake problem is eliminated by design. Additionally, as a security measure in BitShares, the chain with the highest witness participation rate will be chosen as the main chain. This does not only improve the network stability in cases of latency and "communication breakdown" but also hinders a malicious witness from trying to build a fork, as he cannot increase the participation rate in his chain [10]. Since no alternative chain can be build successfully, a selfish mining attack is not possible. Compared to many PoS implementations, a long-range attack in BitShares can be performed if more than 67% (a BFT problem) of the witnesses with 51% of the total voting power cooperate to perform it [8][21]. This attack is more probable if, as in a real world representative democracy, less voters participate in the election system and if they vote by proxy. The witnesses could also corrupt the proxies to get more votes, performing a bribe attack [65]. In contrast to PoS implementations, the attack cannot be performed if a single voter owns 51% of the total coins due to the 2 separate delegate-type system [21]. However, a stakeholder can influence the system by owning more currency than the other voters [65]. For a grinding attack to happen, a witness must be able to influence his chances to be selected more often than the other witnesses per period. This is not

possible by design in BitShares. Also if a witness would try to cheat the system through fake network breakdowns and by increasing the latency, the algorithm would simply skip him and the stakeholders would vote them out for not behaving honest [10].

10.5.2 Proof of Elapsed Time

Another PoX algorithm that has been developed by Intel is called Proof of Elapsed Time (PoET) [45]. Generally, the way a PoET consensus algorithm works is simple: a user gets assigned a certain wait time. After the wait time, the user is allowed to produce the next block. For the user to be able to prove that he has waited the assigned time and that he does not cheat, a *trusted hardware* is needed [64]. Trusted hardware provide a safe environment in which certain instructions can be executed without getting disrupted by the user or by a malicious attacker [45]. Some Intel CPUs are in possession of the "Intel Software Guard Extensions" (Intel SGX), a set of instructions that allow applications to run in a trusted environment [70]. This is done by loading the data into enclaves in the memory [37]. The PoET algorithm can be run in such an enclave, thus preventing cheating and allowing the users to prove their wait time [57]. To further increase the security, there needs to be a central authority like Intel that maintains a list of trusted certificates that can check if the CPU is trustable [64]. Sawtooth Lake is a concrete blockchain-technology, also developed by Intel, that utilizes Intel SGX [36] and that can be used in a permissionless or permissioned network [70][32]. Concretely, Sawtooth Lake works by assigning a different random wait timer to each user. The wait timer starts counting down and the user with the smallest wait time left is chosen as the next block-producer [64].

Since almost no computation needs to be done, it's as demanding as power efficient as for example PoS or DPoS [70]. On the downside, a user can still increase his chances by buying multiple CPUs, making it an arms race like in PoW again [57]. This is the so called "stale chip problem" [57]. The throughput can go up to 1000 transactions per second [17] (claim could not be verified) and further scaling is possible. This kind of scalability is achieved by giving up decentralization and some security. Decentralization is essentially given up almost fully, since there needs to be a trusted authority like Intel [32]. Thus, it might be reasonable to use PoET in a private setting, where trusting a certain authority and decentralization are seen as less risky [32]. The centralization is a double-edged sword in regards to security: Since the application can run in trusted hardware, certain rules can be enforced to the user, like always running the newest version of the protocol [64]. This adds a lot of security to the blockchain, essentially making it resistant too many types of attacks. However, the trusted hardware is also its weakest point. If the trusted hardware has a security risk and gets exploited successfully, the malicious user could cheat and that would enable all sorts of attacks like double spending, selfish mining and grinding attacks [45]. According to Intel, this is a non-issue since it can be easily detected with a statistical analysis [57]. Other than that, the security risks of a PoET algorithm are largely unknown [45]. A successful attack can occur, if $\Theta(\frac{\log \log n}{\log n})$ nodes, where n is the total number of nodes in the system, are compromised. This is different compared to the concept of a 51% attack, since it depends on the size of the system. The system is less vulnerable the bigger n is [45].

10.6 Ouroboros

The following section adds a novel and interesting approach to the consensus algorithms already mentioned. Ouroboros is a provable secure algorithm that uses PoS in specific

timeframe and timeslot architectures with a protocol that provides an unbiased randomness to elect block generators, named slot leaders.

Ouroboros is a PoS algorithm used by Cardano which is a public decentralized open source blockchain and cryptocurrency project [13]. The development team of Cardano consists of a large collective of researchers and experts. Unlike many other open source projects Cardano set the focus on mathematical provable correctness and security through reviewed research of academics and developers. The modular structure is open for functionalities like delegation, sidechains, subscriptable checkpoints, efficiency for light clients, different forms of random number generation and even different synchronization assumptions [14][15][16]. With this flexibility and a research driven approach, Cardano aims to stay flexible, future-proof and with a test setup with 40 nodes and a slot frame of 5 seconds a median of 257.6 transactions per second could be achieved [2].

10.6.0.1 Basic Model

As in PoS, nodes with a positive stake may participate in running the protocol. An elected slot leader can generate new blocks through listening to transactions, generating the block, signing the block with the secret key and finally publishing it to the network. In Ouroboros, time is divided into epochs which are again split up into slots. A slot leader is similar to a miner in Bitcoin, the election process however is different. Each slot has only one leader which has a right to generate exactly one block within that time slot. It follows that because in each slot only at most one block can be generated, in each epoch at most n blocks can be generated, where n defines the maximum amount of slots. If the elected leader misses their slot, the right to generate a block is taken away until the same leader is elected again to generate one block [2].

The two principles, persistence and liveness as described below, as well as a robust transaction ledger, given that the maintaining ledger is divided into time slots which determine the order of transactions, are embedded in the ledger. These factors together provide a robust transaction ledger where honest transactions are adopted and become immutable as soon as the depth of the block is more than k blocks.

Ouroboros reaches persistence and liveness through stability. A transaction is seen as stable when the depth of the block is bigger than the security parameter of k blocks. So when it holds that one node agrees on a transaction to be stable, all other honest nodes agree on that transaction as well, persistence is achieved. If any honest node however would not agree upon a transaction, persistence would not be given. Bitcoin's measurement is comparable to the idea of persistence described where the longest chain of blocks is considered as the correct chain [56].

In Ouroboros, a transaction ledger reaches liveness when a transaction is accepted and added to all nodes after a certain amount of u timesteps. If the response is honest and confirmed by all nodes then the transaction will be reported as stable and the valid transaction will be added to the blockchain. In comparison to Bitcoin where an honest majority assumption and the common prefix property of the backbone protocol of Bitcoin lead to persistence (Lemma 24, Definition 4 and Theorem 16) and liveness (Lemma 25) [42], with a PoS algorithm this is not achievable in the same manner. In PoW, the competition of mining a block lies outside of the blockchain whereas in PoS the leader generating a block is located in the blockchain. This however leads to the need of countermeasures against grinding and nothing-at-stake attacks which could lead to loss of liveness or persistence. In PoW, a nothing-at-stake attack would cause to split the resources between every fork which would lead to also split the monetary value earned. So in PoW a natural incentive is given to only chose the longest fork instead of multiple forks. However, voting for multiple forks in PoS does not imply more costs nor a splitting of the reward. Under a rational assumption of an attacker a nothing-at-stake attack in PoS is executed only

when there are no penalties in the case when adversaries generate blocks on more than one fork. The consequence is that rational nodes would stake on the original chain as well as on the attacker's chain and if other nodes follow the attacker's chain, the persistence would be violated since the history could be rewritten. Ouroboros proves in their scientific paper that a nothing-at-stake attack would be infeasible [2] since each slot is assigned to a identified slot leader uniquely as well as randomly, and to be elected as a slot leader one must have at least 1% at stake [58]. But even if an adversary would try a nothing-at-stake attack, the chain selection rule used in Ouroboros suggests to ignore deep forks that differ from the last received block.

Grinding attacks or "Stake grinding" are a collection of attacks in which the validator has the intention of being elected to mint a next block more often than the random selection would allow [2][67]. In fact, such a grinding attack can also be used in order to operate a double spending attack. In case a grinding attack succeeds, and the electors could change the random election in their favor, liveness would be violated. In Ouroboros, a multiparty coin-flipping protocol is used to obtain the needed (dynamic) randomness for the leader election process which prevents grinding and double spending attacks [2] in each epoch, whereas in conventional PoS systems the random election is based on raw data. The blockchain itself is used as a broadcast channel where in each epoch a coin flipping protocol is executed. At the moment Ouroboros' slot leaders are declared publicly. There are different variations of Ouroboros as for example Praos which will not be discussed further in which only upon publishing a block, stakeholders are able to detect slot leaders. The committee has the power to determine stakeholders that have the permission to find the next group of stakeholders for the following protocol execution as well as the results of the leader election processes for the epoch and thus allowing that slot leader to add a block to the blockchain.

The multiparty computation (MPC) used in Ouroboros delivers the randomness which is needed for the leader election in every epoch. A MPC is required in a trustless environment where nodes do not need to trust any other node and thus can work on private inputs and messages sent from nodes to nodes are handled as black boxes. In the first step of a MPC a commitment is formed. During this formation the defined slot leaders do the coin flipping and generate the random numbers. Under the two assumptions that the outcome of the coin flipping is guaranteed (guaranteed output delivery) only if honest majority is given and R denotes the time of an epoch which contains $10k$ slots in order to simply relate to, in each time slot the following protocol is executed: In the commit phase which lasts $4k$ slots (Figure 13, Protocol DLS [2]) Stakeholder A generates the randomness and encrypts with the private key each generated share of the secret under the respective public key and posts it to the blockchain. The share cannot be opened at this point and remains a secret value to the other Stakeholders B. If however there is a Stakeholder C which controls more than half of the shares, that secret can be opened through reconstruction. The next phase is called Reveal Phase which also lasts $4k$ slots. Upon receiving a random secret from Stakeholder B with the same size as the secret from A, stakeholder A again sends a black box with a key to open the black boxes to Stakeholder B, now containing the key to open the first message that was sent in the Commit Phase as well as the secret. In case Stakeholder B compares the secret received in the Reveal Phase with the opened secret by the received key and equal values are found, the process proceeds. If the opened secret does not match each stakeholder terminates. Finally, in the Recovery Phase which starts at the $8k$ th slot and lasts $2k$ slots, the committee checks for stakeholders that have not revealed their secret yet. These stakeholders now submit their shares and as soon as at least half the shares are posted, all secrets can be reconstructed for each stakeholder. Obviously when less than 50% of the participants broadcast in the reveal phase due to either malicious behavior or simply not being online, the protocol can not reconstruct the secrets from the stakeholder and this process can not continue.

With all the secrets that are publicly known, an *XOR* computation results into a probability that is used as a seed for finding a satoshi, and therefore the owner for each slot in the next epoch can be elected. As soon as the slot leader is found and the next committee is formed, the next epoch starts.

Following the rules of the protocol as a participant which is proven to be an approximate Nash equilibrium is incentivized. Payoffs are offered for protocol action like being a committee member, endorsing a set of inputs or sending messages for the MPC protocol and thus it applies that for those stakeholders that act faithfully, the equilibrium holds when all the stakeholders are rational. This design choice eliminates the need for strong countermeasures against selfish mining and block withholding. However, for not rational attacks and stakeholders with strictly more than 50% of the stakes, this does not hold for Ouroboros since the requirements liveness and persistence are not given anymore. In that case it would be possible that the honest users will be skipped or left out [2].

It is notable that direct comparisons in throughput and performance as well as applicability between Ouroboros and any other already established, under real world conditions tested consensus algorithm can not be done as of yet. Although the research background is rigorously encouraged in Cardano, Ouroboros has not yet been tested in a real world scenario with hundreds or thousands of nodes and thus the scalability under those conditions can not be assessed yet. Nevertheless, Ouroboros is a good example of creating a consensus algorithm with focus on scientifically proving correctness and security. Therefore, Ouroboros is classified between being decentralized and secured, similar to Bitcoin. Due to the experimental testing of the scalability of Ouroboros, further evaluations need to be considered in order to be able to reclassify Ouroboros for a potential solution to the trilemma.

10.7 Open Representative Voting

Another exotic consensus algorithm is the open representative voting (ORV) combined with PoW, DPoS and an asynchronous block lattice architecture where each account has it's own account-chain in order to send and receive transactions as well as calculate it's balance.

Raidblocks launched in 2015 and rebranded in 2018 as Nano is a cryptocurrency which uses an innovative Direct Acyclic Graph (DAG) as an underlying structure and ORV to find consensus. Instead of one blockchain which has to be agreed by the global network like in Bitcoin, with block lattice only the sender and receiver have come to consensus in order to complete a transaction meaning that users are assigned to their own account-chains. Consensus is achieved by a balance-weighted vote of representatives on transactions that are conflicting, comparable to DPoS. PoW is additionally used to counter spamming attacks on the senders side. Considering the PoW sequence this leads to a maximum of 5 transactions per second that can be sent from the same account and about 10'000 transactions per second which mark the theoretical maximum bound of the current algorithm [19] using up-to-date hardware. Also mentionable is that the PoW for the transaction is pre-cached which means that the next transaction will be instantly. Nano's PoS protocol with ORV enables more usability for users as there are no fees involved in processing transactions. As of writing this paper, in comparison to a Bitcoin transaction that needs 400KWh [24] to transact one BTC, one Nano transaction needs less than 0.001KWh. With these features, Nano seeks to revolutionize the cryptocurrency field with fee-less real-time transactions [19].

10.7.0.1 Basic Model

Each account has its blockchain which represents the accounts transaction history. By only allowing the owner of the account to update its so called account-chain, the update is immediate and does not depend on other transactions on the block lattice. And since nodes record and rebroadcast, a wide spectrum of devices and hardware can be used allowing potential integration into Internet of Things. Several rather small projects, as for example an implementation of payment to a charging device for smartphones, have been realized[12].

Each transfer requires two transactions where the *send* defines the amount from the sender and the *receive* simply adds the amount to the receivers account. Handling the transfer of funds in this way brings several advantages. The first advantage and also a key design component is that incoming transfers of funds are asynchronous which allows the receiving account to decide in which order the incoming blocks are signed. Keeping the information slim to fit into UDP packets forms the second advantage, although in the future Nano considers to switch to TCP [3]. Because the running total balance is up to date at any time, the associative addition of further transaction amounts is not dependent on the order of the respective transactions. PoW as it can be seen in [7] where a C++ implementation of the PoW algorithm is shown, is only used as a countermeasure against spamming attacks. On high end consumer computers as of 2019, this PoW algorithm will exceed the threshold variable in the while statement in a fraction of a second [19], similar to Hashcash [1].

Incoming transactions can either be settled or unsettled. In settled transactions, the account has already generated receiving blocks, whereas in unsettled transactions, the cumulative balance is not added yet. So this implies that the nodes or their representatives must be online in order to receive transactions [27]. Moreover, during the sending process, the sender must have a balance and therefore an account with the address must be registered. The *send* message contains the fields previous, balance, destination, work, type and signature. In the field previous a hash of the previous block in that account-chain can be found and in the destination field, the destination of where the funds are being sent to is defined. After a block is being confirmed, the broadcast to the network starts and the funds are pending, assuming that the deduction is instant. Pending funds cannot be revoked by the sender. As soon as the receiver signs a *receive* block the sending process is completed and the receiving process starts. The receiver then creates the *receive* block and adds the hash of the *send* transaction to the source field. The creation of the *receive* block leads to the broadcast and after that, the balance is updated and the transaction is completed [19].

As in DPoS, representatives can be elected in block lattice in order to resolve conflicts by voting. Not every user might want to run a node and giving representatives the voting power might be favorable for the user for usability reasons as mentioned before, where a node must be online in order to receive transactions. At the time of creation of the account a representative must be chosen. The representative can easily be redefined through specifying this in the most common wallets and changed through a *change* request, similar to one of the transaction messages shown before. A *change* transaction consists of the fields previous, representative, work, type and signature. The transaction subtracts the voting weight from the previous representative and adds it to the newly defined representative. Weight of a node is the sum of all account balances that this node is the representative of [19]. In Nano, every node that has a stake of at least 0.1% can become a representative and the incentive to do so lies in not having to rely on a third party [50].

Forks are seen as either bad programming or an adversary double spending of an owner. In an adversary attack scenario double spending would inevitably lead to a fork in which two blocks refer to the same previous block. As soon as such malicious actions get detected,

a voting is broadcasted. The representatives weighted nodes will vote for a short period of time in order to find the winning block. Voting weights of the representatives are the sum of all accounts that have delegated their coins to them. After using the DPoS algorithm, the winning block is found as the most voted block and is then kept in the node's ledger where as losing blocks are removed. Representatives only vote when such double spending attacks or forking occurs. Then if the following conditions are given a transaction is seen as valid: First, there should be no duplicate transactions which means that there should not exist forks like mentioned. Second, each transaction is signed by the owner of that account in the creation of the *send*, *receive* or *change* transactions. And third, the accounts and especially accounts with a sending transaction must have a balance and on the receivers side the PoW threshold must be met.

Other attacks, as for example Sybil attacks where an adversary would create or maliciously conquer a vast amount of Nano nodes, would not lead to more delegational power nor incentive since the DPoS system relies on the delegated sum of stakes and not on the number of nodes. One other possible attack is flooding the network with either unnecessary but valid transactions or transactions where minimal amounts of currency are sent. Both of these types are to a degree countered by the PoW algorithm which limits the sending from an account to the power of the given hardware (Table 1 [19]). A third type of attack is precomputing blocks on the accounts chain with the malicious purpose of broadcasting and creating a Denial of Service (DoS) attack. Combining this with a 51% attack on the weighted system it would cause the system to break. At the moment, Nano is investigating ways to mitigate such a combination. The design of the consensus system itself would require the adversary to damage their investment since the ledger would be of no value anymore. Also considering that depending of the size of the network at the time of the attack, that investment of the node could not be recovered. Important to note and consider is that a 51% attack can be lowered to for example 33% in the case a DDoS is able to put another 33% of the representatives offline in case of a combination of attacks. The last attack regarding Nano is bootstrap poisoning which might remind of selfish mining. In fact, in bootstrap poisoning, the attacker tries to hold some old accounts private-key for a long time in order to waste time for the honest nodes in the network. By keeping that old representation of the network that adversary representative would try to get votes of nodes by presenting wrong information. Nano solves this problem by letting nodes pair themselves with a database of accounts with legitimate block heads [19]. Since this solution would involve using a database representation of information which should be stored on the blockchain, true decentralization would be questionable in the case of Nano.

Block lattice as the architecture behind consensus algorithms using PoW for spamming, DPoS for conflicts, ORV and DAG consisting of nodes of account-chains offers potential for very good performance. Nano picks each algorithm carefully in order to solve the problem which the algorithm was designed for. At the moment, the representative with the highest weight controls 26.43 % of the stake. In comparison with the leading representative, the second and third most voted representatives control 19.62 % of the votes combined. And although this distribution does not seem quite well, a voting in the form as described has not been needed yet [51]. Therefore, if the DPoS used to find consensus in Nano would be used, decentralization would be questionable as 8 out of 16 of the top representatives are official representatives owned by Nano. On the other hand, scalability in the block lattice architecture is seemingly given. Considering rational users and representatives, Nano with the ORV consensus algorithm can be seen as secure and scalable and thus does not provide a complete coverage for all of the three parameters in the trilemma.

Table 10.1: Comparison of consensus algorithms 1

Consensus Protocol	Working public implementation	Incentive	public/private blockchain	Transactions per Second
fpBFT	Stellar Consensus Protocol	Securing network	public	1000
pBFT	Neo	Monetary compensatoin	permissioned	1000
PoW	Bitcoin	Monetary compensatoin	public	7 [1]
PoS, PoW	PPCoin	Monetary compensatoin	public [2]	8 [3]
DPoS	BitShares	Monetary compensatoin	public	3328.33 [4]
PoET	Sawtooth Lake	depends on use case; monetary compensation possible	public permissioned private	1000
PoS Ouroboros	Ouroboros	Monetary compensatoin	public	256.7
DAG Block Lattice DPoS, PoW	Nano	not realiant on third party	public	10000

[1] Up to 60 in theory.

[2] In general, an implementation of a PoS algorithm can be private or public.

[3] These values depend strongly on the implementation and not only the consensus algorithm.

[4] On a stresstest only, current transactions per second is around 7 - 10.

10.8 Conclusion

The following tables serve as an overview of the different features analyzed for each algorithm. Table 1.1 gives an overview of the general features while table 1.2 focuses on the trilemma and the security aspects.

A wide variety of consensus algorithms and organizations that are using these different approaches have been analyzed. As seen in the Byzantine generals problem, not every node can be trusted in a public environment. Using a protocol which is well suited for public environments (*e.g.*, PoW) does not necessarily fit to a permissioned blockchain. Depending on the use case, some algorithms are better suited for permissioned networks, as for example PoET. The public permissionless Bitcoin blockchain, being a permissionless network, solved the Byzantine problem by sacrificing scalability while trying to achieve a high decentralization of nodes. PoW solves the Byzantine problem by reaching a consensus with miners competing in a *hash-war*. The task given to the nodes consists of a computational problem that is easily validated by the nodes but at the same time hard to solve since it requires a vast amount of energy.

To evolve from the wasteful PoW consensus to a more energy-efficient Byzantine-fault-tolerant algorithm, PoS was introduced. Using PoS, miners are incentivized to remain honest without the need for a *hash-war*. While PoS encourages decentralization and can improve scalability, the trade-off mostly occurs in terms of security. But as seen in Ouroboros, which has not been evaluated on a large network yet, security is guaranteed by the different protocols implemented around the PoS system assuming liveness and persistence are given. Considering the already successfully implemented PPCoin however, security issues are prevalent, as for example the nothing-at-stake problem. A more secure

Table 10.2: Comparison of consensus algorithms 2

Consensus Protocol	Trilemma			Resistance to Attack				
	Scalability	Security	Decentralisation	Double Spending Attack	Grinding Attack	Nothing-at-Stake	Selfish Mining	Sybil Attack
fpBFT	✓	✓	X	S	I	I	I	33%
pBFT	✓	✓	X	S	I	I	I	33%
Pow	X	✓	✓	S [1]	I	P	S	51% [2]
PoS, PoW	✓	X	✓	S	P	P	P	1% to 51%
DPOS	?	✓	X	S	S	S	S	67% of witnesses 51% of stakeholders
PoET	?	✓	X	S	S	S	S	<50% [3]
PoS Ouroboros	?	✓	✓	S	S	S	S	51%
DAG Block Lattice DPOS, PoW	✓	✓	X	S	I	I	I	51%

?: Questionable due to quantity of evaluations

S: Secured

I: Impossible by design

P: Prone

[1] 51% is required and its very expensive.

[2] Selfish mining by colluding nodes can lower the security threshold down to 25%.

[3] The percentage is variable, since it depends on n , the amount of nodes in the system.

The formula to calculate the percentage is: $\Theta(\frac{\log \log n}{\log n})$.

variant of PoS, DPoS, can be used to further achieve a trade-off between security and scalability versus decentralization.

None of the compared algorithms, except DPoS, could handle a 51% attack. In some cases, a combination of attacks as for example in Nano where a DoS combined with a 51% attack could lead to only needing 33% of the stakes in that respective representative. In case of DPoS, not only 67% of the witnesses would need to be malicious, but also 51% of the voters for a successful attack.

As of now, the trilemma has not been solved by a single consensus algorithm. It is important to evaluate carefully which trade-off to make before choosing a solution.

All in all, law restrictions as well as the efficiency of mining as of this moment lead to the need of more feasible consensus algorithms. Different laws and economical factors may have an influence on the algorithms.

Bibliography

- [1] Adam Back, *Hashcash - A Denial of Service Counter-Measure*. August 1, 2002, [On-line] https://www.researchgate.net/publication/2482110_Hashcash_-_A_Denial_of_Service_Counter-Measure, last visit May 25, 2019.
- [2] Aggelos Kiayias, Alexander Russell, Bernardo David, Roman Oliynykov: *Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol*; 37th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2017), California, USA, August 2017, LNCS, Springer, Cham, Vol. 10401, pp. 357-388.
- [3] Andy Johnso, *Weekly Update 4/15/19*. 2019, [On-line] <https://medium.com/nanocurrency/weekly-update-4-15-19-68e6be922699>, last visit May 25, 2019.
- [4] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, Srdjan Capkun: *On the Security and Performance of Proof of Work Blockchains*; 23rd ACM Conference on Computer and Communications Security (CCS 2016), Vienna, Austria, October 2016, pp. 3-16.
- [5] ash, *Current BitShares Testnet Stress-test Highlight: 3300TXs 14000OPs*. 2017, [On-line] <https://steemit.com/bitshares/@ash/current-bitshares-testnet-stress-test-highlight-3300tx-14000ops>, last visit May 25, 2019.
- [6] Ayelet Sapirshtein, Yonatan Sompolinsky, Aviv Zohar: *Optimal Selfish Mining Strategies in Bitcoin*; International Conference on Financial Cryptography and Data Security (FC 2016), Christ Church, Barbados, August 2016, LNCS, Springer, Berlin, Heidelberg, Vol. 9603, pp. 515-532.
- [7] Ben Green, *Code cleanup, add test*. January 8, 2018, [On-line] <https://github.com/numtel/node-raiblocks-pow/blob/70d26cde2ab9eed91a168e49136aa46c44a2d052/functions.cpp#L7>, last visit May 25, 2019.
- [8] BitFury Group, *Proof of Stake versus Proof of Work*. September 13, 2015, [On-line] <https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work/>, last visit April 11, 2019.
- [9] BitShares, *BitShares Block Explorer*. n.d., [On-line] <https://wallet.bitshares.org/#/explorer/blocks>, last visit April 16, 2019.
- [10] BitShares, *Delegated Proof-of-Stake Consensus*. n.d., [On-line] <https://bitshares.org/technology/delegated-proof-of-stake-consensus/>, last visit April 16, 2019.
- [11] BitShares Blockchain Foundation, *The BitShares Blockchain*. June 1, 2018, [On-line] <https://www.bitshares.foundation/articles/2018-06-01-bitshareswhitepaper>, last visit April 16, 2019.

- [12] Cami Albert, *Nano (NANO) IOT Charger Sends NANO Coin Price to the Moon*. August 25, 2018, [On-line] <https://cryptoglobalist.com/2018/08/25/nano-nano-iot-charger-sends-nano-coin-price-to-the-moon/>, last visit May 25, 2019.
- [13] Cardano, *Introduction*. n.d., [On-line] <https://cardanodocs.com/introduction/>, last visit May 25, 2019.
- [14] Cardano, *Ouroboros Proof of Stake Algorithm*. n.d., [On-line] <https://cardanodocs.com/cardano/proof-of-stake/>, last visit May 25, 2019.
- [15] Cardano, *cardanoroadmap*. n.d., [On-line] <https://cardanoroadmap.com/>, last visit May 25, 2019.
- [16] Cardano, *Why we are building Cardano*. n.d., [On-line] <https://whycardano.com>, last visit May 25, 2019.
- [17] Carlo Gutierrez, *Hyperledger's Sawtooth Lake Aims at a Thousand Transactions per Second*. March 13, 2017, [On-line] <https://www.altoros.com/blog/hyperledgers-sawtooth-lake-aims-at-a-thousand-transactions-per-second/>, last visit April 18, 2019.
- [18] Christian Cachin, Marko Vukolić: *Blockchain Consensus Protocols in the Wild*; CoRR, July 7, 2017, arXiv:1707.01873.
- [19] Colin LeMahieu, *Nano: A Feeless Distributed Cryptocurrency Network*. 2018, [On-line] <https://nano.org/en/whitepaper>, last visit March 3, 2019.
- [20] CryptoLions, *EOS Network Monitor*. n.d., [On-line] <https://eosnetworkmonitor.io/#>, last visit April 16, 2019.
- [21] Daniel Larimer, *DPOS Consensus Algorithm - The Missing White Paper*. 2017, [On-line] <https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper>, last visit April 18, 2019.
- [22] David Mazières, *The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*. April 2015, [On-line] <https://www.stellar.org/papers/stellar-consensus-protocol.pdf>, last visit May 25, 2019.
- [23] David Z. Morris, *Bitcoin Hits a New Record High, But Stops Short of \$20,000*. December 17, 2017, [On-line] <http://fortune.com/2017/12/17/bitcoin-record-high-short-of-20000/>, last visit May 26, 2019.
- [24] Digiconomist, *Bitcoin Energy Consumption Index*. n.d., [On-line] <https://digiconomist.net/bitcoin-energy-consumption>, last visit April 11, 2019.
- [25] Erik Zhang, *A Byzantine Fault Tolerance Algorithm for Blockchain*. n.d., [On-line] <https://docs.neo.org/en-us/basic/consensus/whitepaper.html>, last visit April 20, 2019.
- [26] Fahad Saleh, *Blockchain Without Waste: Proof-of-Stake*. February 27, 2019, [On-line] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3183935, last visit May 27, 2019.
- [27] Federico Matteo Benčić, Ivana Podnar Žarko: *Distributed ledger technology: blockchain compared to directed acyclic graph*; 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, July 2018, pp. 1569-1570.

- [28] Fred B. Schneider: *Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial*; ACM Computing Surveys, Vol. 22, No. 4, December 1990, pp. 299-319.
- [29] Ghassan Karame: *On the Security and Scalability of Bitcoin's Blockchain*; 23rd ACM Conference on Computer and Communications Security (CCS 2016), Vienna, Austria, October 2016, pp. 1861-1862.
- [30] Giulia Fanti, Leonid Kogan, Sewoong Oh, Kathleen Ruan, Pramod Viswanath, Gerui Wang: *Compounding of Wealth in Proof-of-Stake Cryptocurrencies*; CoRR, October 15, 2018, arXiv:1809.07468.
- [31] Greg Lee, TestZ, Josh Lavin, Daniel Larimer, Thomas Cox, Nathan Hourt, Qianli Ma, William Prioriello, *EOS.IO Technical White Paper v2*. April 28, 2018, [On-line] <https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md>, last visit April 16, 2019.
- [32] Hadja F. Ouattara, Daouda Ahmat, Tounwendyam Frédéric Ouedraogo, Tegawendé F. Bissyandé, Oumarou Sié: *Blockchain Consensus Protocols*; 9th EAI International Conference on e-Infrastructure and e-Services for Developing Countries (AFRICOMM 2017), Lagos, Nigeria, December 2017, LNICST, Springer, Cham, Vol. 250, pp. 304-314.
- [33] Hubert Kirrmann, *Fault Tolerant Computing in Industrial Automation*. 2005, [On-line] http://www.solutil.ch/kirrmann/FaultTolerance/20050418_HK_FT_Tutorial.pdf, last visit May 26, 2019.
- [34] Iddo Bentov, Ariel Gabizon, Alex Mizrahi: *Cryptocurrencies Without Proof of Work*; International Conference on Financial Cryptography and Data Security (FC 2016), Christ Church, Barbados, August 2016, LNCS, Springer, Berlin, Heidelberg, Vol. 9604, pp. 142-157.
- [35] Igor M. Coelho, Vitor N. Coelho, Peter Lin, Erik Zhang, *Delegated Byzantine Fault Tolerance: Technical details, challenges and perspectives*. March 14, 2019, [On-line] https://docs.neo.org/en-us/08_dbft.pdf, last visit May 26, 2019.
- [36] Intel Corporation, *Sawtooth v1.1.4 documentation - Introduction*. n.d., [On-line] <https://sawtooth.hyperledger.org/docs/core/releases/latest/introduction.html>, last visit April 18, 2019.
- [37] Intel Software, *Intel Software Guard Extensions*. n.d., [On-line] <https://software.intel.com/en-us/sgx>, last visit April 18, 2019.
- [38] Ittay Eyal, Emin Gün Sirer: *Majority is Not Enough: Bitcoin Mining is Vulnerable*; Communications of the ACM, Vol. 61, No. 7, July 2018, pp. 95-102.
- [39] James Ray, *Problems*. August 22, 2018, [On-line] <https://github.com/ethereum/wiki/wiki/Problems>, last visit April 15, 2019.
- [40] Jean-Claude Laprie: *On Computer System Dependability: faults, errors and failures*; 13th IEEE Computer Society International Conference (COMPCON 1985), California, USA, February 1985, pp. 256-259.
- [41] John R. Douceur: *The Sybil Attack*; Peer-to-Peer Systems, First International Workshop (IPTPS 2002), Massachusetts, USA, March 2002, LNCS, Springer, Berlin, Heidelberg, Vol. 2429, pp. 251-260.

- [42] Juan Garay, Aggelos Kiayias, Nikos Leonardos: *The Bitcoin Backbone Protocol: Analysis and Applications*; Advances in Cryptology - EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2015), Sofia, Bulgaria, April 2015, LNCS, Springer, Berlin, Heidelberg, Vol. 9057, pp. 281-310.
- [43] Karl J. O'Dwyer, David Malone: *Bitcoin mining and its energy footprint*; 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, June 2013, pp. 280-285.
- [44] Leslie Lamport, Robert Shostak, Marshall Pease: *The Byzantine Generals Problem*; ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, pp. 382-401.
- [45] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, Weidong Shi: *On Security Analysis of Proof-of-Elapsed-Time (PoET)*; 19th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2017), Massachusetts, USA, November 2017, LNCS, Springer, Cham, Vol. 10616, pp. 282-297.
- [46] Loi Luu, Viswesh Narayanan, Kunal Baweja, Chaodong Zheng, Seth Gilbert, Prateek Saxena, *SCP: A Computationally-Scalable Byzantine Consensus Protocol For Blockchains*. December 13, 2015, [On-line] <https://www.weusecoins.com/assets/pdf/library/SCP%20-%20A%20Computationally-Scalable%20Byzantine.pdf>, last visit May 25, 2019.
- [47] Mauro Conti, Ankit Gangwal, Michele Todero: *Blockchain Trilemma Solver Algorand has Dilemma over Undecidable Messages*; CoRR, January 28, 2019, arXiv:1901.10019.
- [48] Michael J. Fischer, Nancy A. Lynch, Michael S. Paterson: *Impossibility of Distributed Consensus with One Faulty Process*; Technical Report No. HIT/LCS/TR-282, September 1982; Department of Computer Science, Yale University, USA.
- [49] Miguel Castro, Barbara Liskov: *Practical Byzantine Fault Tolerance*; Third Symposium on Operating Systems Design and Implementation (OSDI 1999), Louisiana, USA, February 1999, USENIX Association, Vol. 99, pp. 173-186.
- [50] Nano, *The Incentives to Run a Node*. November 3, 2018, [On-line] <https://medium.com/nanocurrency/the-incentives-to-run-a-node-ccc3510c2562>, last visit May 26, 2019.
- [51] Nanode, *Representatives*. n.d., [On-line] <https://www.nanode.co/representatives>, last visit May 26, 2019.
- [52] Neo, *Blockchain Info*. n.d., [On-line] <https://neo.org/consensus>, last visit May 27, 2019.
- [53] Nichanan Kesonpat, *Consensus Algorithms: Proof-of-Stake & Cryptoeconomics*. June 9, 2018, [On-line] <https://www.nichanank.com/blog/2018/6/4/consensus-algorithms-pos-dpos>, last visit April 16, 2019.
- [54] Peercoin, *Comparison with other blockchain networks*. n.d., [On-line] <https://docs.peercoin.net/>, last visit May 27, 2019.
- [55] Samuel Eilenberg: *Automata, Languages, and Machines*; Academic Press, Inc., Orlando, Florida, USA, 1974.

- [56] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. October 31, 2008, [On-line] <https://bitcoin.org/bitcoin.pdf>, last visit May 26, 2019.
- [57] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, George Danezis: *Consensus in the Age of Blockchains*; CoRR, November 14, 2017, arXiv:1711.03936.
- [58] Spencer J. Hosack: *Use of the Proof-of-Stake Algorithm for Distributed Consensus in Blockchain Protocol for Cryptocurrency*; Honors Scholar Theses, University of Connecticut, USA, Supervisors: Yaacov Kopeliovich, 2018.
- [59] Stellar Development Foundation, *On Worldwide Consensus*. April 8, 2015, [On-line] <https://medium.com/stellar-development-foundation/on-worldwide-consensus-359e9eb3e949>, last visit April, 18, 2019.
- [60] Steve Walters, *Delegated Proof of Stake (DPoS): What is It? - Complete Beginners Guide*. August 20, 2018, [On-line] <https://www.coinbureau.com/education/delegated-proof-stake-dpos/>, last visit April 16, 2019.
- [61] Stuart Haber, W. Scott Stornetta: *How to Time-Stamp a Digital Document*; 10th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 1990), California, USA, August 1990, LNCS, Springer, Berlin, Heidelberg, Vol. 537, pp. 437-455.
- [62] Sunny King, Scott Nadal, *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*. August 19, 2012, [On-line] <https://decred.org/research/king2012.pdf>, last visit May 25, 2019.
- [63] Thomas Bocek, Burkhard Stiller: *Smart Contracts - Blockchains in the Wings*; in Claudia Linnhoff-Popien, Ralf Schneider, Michael Zaddach (Edts.) "Digital Marketplaces Unleashed", Springer, Berlin, Germany, 2018, pp. 169-184.
- [64] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, Ji Wang: *Untangling Blockchain: A Data Processing View of Blockchain Systems*; IEEE Transactions on Knowledge and Data Engineering, Vol. 30, No. 7, July 2018, pp. 1366-1385.
- [65] Tyler Jenks, *Pros and Cons of the Delegated Proof-of-Stake Consensus Model*. August 16, 2018, [On-line] <https://www.verypossible.com/blog/pros-and-cons-of-the-delegated-proof-of-stake-consensus-model>, last visit April 18, 2019.
- [66] Vitalik Buterin, *Long-Range Attacks: The Serious Problem With Adaptive Proof of Work*. May 15, 2014, [On-line] <https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work/>, last visit April 14, 2019.
- [67] Vitalik Buterin, *Proof of Stake FAQ - How does validator selection work, and what is stake grinding?*. March 20, 2019, [On-line] <https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ#how-does-validator-selection-work-and-what-is-stake-grinding>, last visit April 14, 2019.
- [68] Vitalik Buterin, *Proof of Stake FAQ - What is the "nothing at stake" problem and how can it be fixed?*. March 20, 2019, [On-line] <https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ#what-is-the-nothing-at-stake-problem-and-how-can-it-be-fixed>, last visit April 14, 2019.

- [69] Vitalik Buterin, *Sharding FAQ*. March 20, 2019, [On-line] <https://github.com/ethereum/wiki/wiki/Sharding-FAQ#this-sounds-like-theres-some-kind-of-scalability-trilemma-at-play-what-is-this-trilemma-and-can-we-break-through-it>, last visit April 15, 2019.
- [70] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, Dong In Kim: *A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks*; IEEE Access, Vol. 7, January 2019, pp. 22328-22370.

