University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Design and Implementation of an Integrated Water Quality Monitoring System and Blockchains

Sanjiv Subodhnarayan Jha
Zurich, Switzerland
Student ID: 15-728-074

Supervisor: Sina Rafati, Dr. Thomas Bocek
Date of Submission: January 25, 2018

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

ifi

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuhlestrasse 14, CH-8050 Zurich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

A trust based network is often liable of vulnerabilities. This study proposes a solution helps to keep track of changes made on the Internet of Things (IoT) data during transmission using distributed hyper ledger (blockchain) and LoRa technologies. Unauthorized data manipulation in pollution detection systems is a huge problem these days, which needs to be controlled. Through this study a solution is proposed to help store the data into blockchain and an end to end encryption using LoRaWAN. A trigger from a this reliable pollution monitoring system using Smart Contracts would help the pollution measuring labs to spend less energy and money in the management of the IoT devices.

This work proposes an IoT- and Blockchain-based, distributed system, for automated measuring, storing, and monitoring of water and air quality in environments such as lakes, mountains, urban areas, or factories. Comparable state-of-the-art solutions require, human interaction to access the data, or high power consumption, or space requirements, or they are based on centralized architectures.

The proposed pollution monitoring system here employs LoRa to address the high power consumption and long-range transmission challenges of IoT protocols with fully decentralized way of storing and retrieving the data recorded by IoT sensors. Thus, data integrity is provided without the need for a Trusted Third Party (TTP) and data is collected and captured automatically without any manual operations needed.

Observations on the four different types of sensors for measuring Potential Hydrogen (PH), Turbidity, Carbon monoxide (CO), and Carbon dioxide (CO2), revealed a high accuracy with the expected time-lines of measurements, non-falsified experimental values collected and used as a reliable evidence of presence of pollution in the environment (Air and Water).

Acknowledgments

At very first, I would like to thank Prof. Dr. Burkhard Stiller for letting me work under his close supervision on one of the promising and latest topics out there. I also want to thank him for helping me in gathering all the required resources for my experiment without doubting my expertise at all.

Further, I am grateful to Mr. Sina Rafati Niya, who helped me along the way with his functional and technical knowledge. I feel lucky to have him as my Advisor in this thesis, he always pushed me towards taking risks in experimenting with my ideas. He even helped me in writing and revising this document.

I am also thankful to Dr. Corinna Schmitt. She helped me in understanding the logic behind some of the concept and always encouraged me to ask questions.

Last but not least, I am extremely thankful to Mr. Gonzalo Casas, for answering my question from promptly and helping me in understanding the minor concepts of LoRa and TTN. He also helped me in getting the LoRa network running for my experiment in this thesis.

I am also thankful to my dear friends and family for supporting and understanding me throughout this time and even before that.

Contents

Chapter 1

Introduction

1.1 Motivation

Increasing population and industrialization leads to increase in pollution, the overall pollution scale is leading to the global warming and other major environmental hazards. People are trying all the possible ways to keep the contamination in control and limit the possibilities of pollution in all the forms, e.g., Carbon footprints in companies, using renewable energies and limiting the use of highly corrosive and environment endangering products. All most every countries and their government has always worked in this field for lowering the pollution quotient [4].

The major contamination can be seen are in water and air. Depending on the type of pollution there are many different techniques to handle it through identification and treatment procedures. For an exact treatment it is very important to know the issues clearly first. To differentiate among levels of contamination, there are some possible ways to verify the contamination before declaring them. These are often the limits of contamination declared by the municipal corporations of the area. Once the lab has checked on the contamination they are supposed to be updating the record, these records are then used for the treatment and in amendment of environmental laws. A central copy of data is often vulnerable of corruption.

There are some cases of untrue data put in the public use has come in news through out the years around the world. To minimize the risk of having an untrue central data, Blockchain (BC) [2] concept can be very efficient.The thesis provides a prototype of a similar Pollution Monitoring System (PMS) which allows the Labs and Internet of Things (IoT) to store measured data in form of transactions and can be accessed publicly.

1.2 Description of Work

In this thesis, two of the main cases of pollution *i.e,* Water and Air Pollution are considered. It caters the identification and verification of water and Air pollution. The present scenario has already been discussed in the earlier section 1.1.

The communication methods used in the transmission of the gathered data are not always trustworthy and if at all they keep the data consistent, it is very difficult to check for the actual data and track the data corruption. There have been many similar case around the world of data corruption in pollution control department, which pushes life of all resident of the whole world towards polluted future. Because we do not have a correct data it becomes highly risky to take decision for Human welfare. Small changes in the record can lead to worst situation in future, so, it is very important to keep the data consistent. This can be achieved if we have a centralized copy of the actual data for verification, but if someone makes changes to the centralized data then we come to the same risk all over again and this time the situation becomes even worst. The whole risks revolving around centralized data hints the idea of using decentralized and distributed data storage technique.

To achieve the benefits of a decentralized network without the trust issues in the network, the concept of peer to peer network [1] can be a very useful. The BC according to Kosba, Ahmed, et al.[2], can keep the correct data and can be verified at any point and from anywhere, without the need of any special permission. The distributed replication nature of BC, keeps the data corruption proof and far from identity theft.

A simple case can be seen as: A sensor is placed at a river bank, it has been enabled to send the data to the internet services or direct into a BC and this data can be used by labs and even updated after thorough check. The data is then used by the pollution control board, which goes through many different channels.

The Sensors used in the identification process are vary crucial part of the whole system and they can be classified depending on the Micropollutants [5] being identified using them. The generally identifiable micro pollutants are, Pathogens- Coliform and E-Coli Bacteria(Human and Animal waste) Inorganic Materials- Heavy metals(Arsenic, Mercury, Copper, Chromium, Zinc, Industrial waste) Organic Materials- MBTE (Carbon chemicals) Macroscopic Pollution- Contamination of water due to large object (paper, shipwrecks, plastic) breakdown into water. Fertilizers- Chemicals increase count of Algae, which reduces the oxygen and cloud the water. Multiple large scale sensors are available in the market for identifying pH, dissolved oxygen (DO), oxidation-reduction potential (ORP), conductivity (salinity), turbidity, temperature and dissolved ions (Fluoride (Fluoride (F-), Calcium (Ca2+), Nitrate (NO3-), Chloride (Cl-), Iodide (I-), Cupric (Cu2+), Bromide (Br-), Silver (Ag+), Fluoroborate (BF4-), Ammonia (NH4), Lithium (Li+), Magnesium (Mg2+), Nitrite (NO2-), Perchlorate (ClO4), Potassium (K+), Sodium (Na+), sort of contamination [5].

There are multiple other methods not using sensors explicitly to identify the pollutants especially in the Chemical pollution, such as Biological monitoring and Testing in laboratories. Biological monitoring such as using organisms for monitoring them on different pollutants. The changes in their behaviours give the indication of abnormality in water constituents. For example the behaviour of mussels in intoxicated water experiment in W. Slooff et al [6].

The proposed solution for the existing problem discussed in this section is elaborated in the next section 1.3.

1.3 System Design

The system consists of major three parts as IoT system, Monitoring system and the BC. Present day IoT system is quite efficient in gathering the data from different sources and push the data onto the cloud for further usages, which is then taken by different users for analysis and diagnosis. This approach is more centralized, which is difficult to scale and secure the data availability. Pushing the data to a BC might provide a better stability and more secure data with high availability. The ultimate system design for the project follows same philosophy, the data received from sensors are then accessed by the monitoring system which is accessible to only authenticated users. The data from laboratories and sensors are pushed to the BC. This system is also used for verifying the data already stored into the BC. This helps the people responsible for generating the problem solutions, to refer to the data publically but they cannot change the data as the permissions are given only to the laboratory staff (authenticated users).

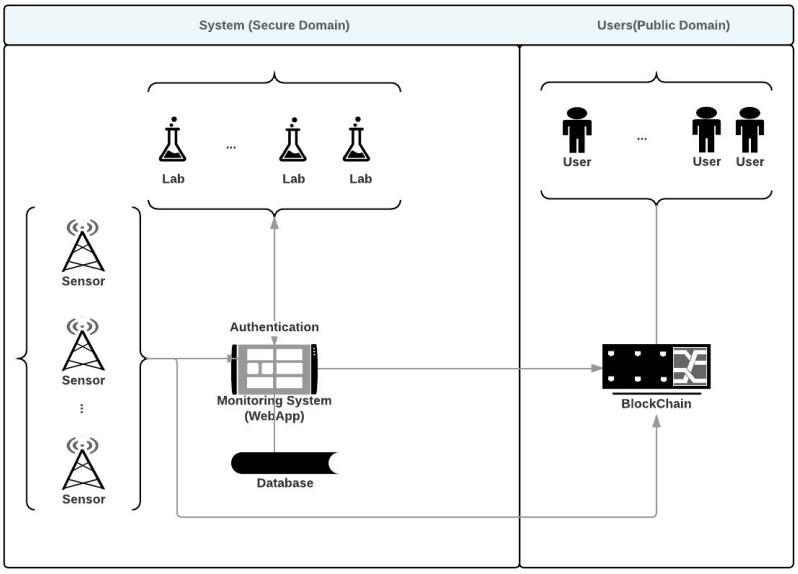

Figure 1.1: System Overview

The system overview in figure 1.1 shows two interconnected parts as System and users,which are connected through a monitoring system (web application). The data from the sensors are retrieved and stored in the Local database for easy and fast access. At the same time the data is stored onto a BC for keeping the data integrity intact. Hence, the data is made public yet secure through BC. The update / create operations can only be applied by the authenticated users who have the permission to do so.

A detailed description on the implementation of proposed system depending on the given overview is given in the implementation section 3.2.

1.4 Thesis Outline

The further chapters provide a deeper knowledge and analysis of BC available in the market and the sensors to detect the pollutants (discussed in the section 1.3).

Next section provides the details about the related work done to make the analysis for further sections. Here we are going to find the best possible solution for the communication between the BC and the IoT in terms of cost, efficiency, scalability, security, and reliability of the presented solution. This also describes the best BC available in the market for making cheapest and trustworthy connection with the sensors.

Depending on relative analysis result a prototype is proposed to detect pollution standard violations and store the report onto the BC and even locally for getting the real-time historical reports without being relied on the BC and internet connectivity.

Implementation section 3.2 shows the dependencies, design challenges and how to use the proposed solution prototype. The evaluation section 4 of the thesis digs deeper into the actual value of the solution which in turn helps to summarize the solution and draw a conclusion at the end in section 5.

Chapter 2

Related Work

In this section classification of different communication protocols, sensors and systems are given to decide on the design of proposed solution prototype, discussed in the section 3.2.

2.1 Classification of Wireless Communication Protocols

IoT has now spread across the globe and myriad of devices connecting to this network is increasing daily. These things are using some type of sensors for getting the state of pollution in the environment. It is a fastest growing industry, which faces security challenges about trust, identity [49] and device management.

In the modern laboratories for pollution monitoring, the data are collected using some sensors which are placed near the targeted area. This area can be very far from the laboratories and maintenance of these sensors on regular basis is not always feasible. So, using some low energy consuming communication protocol can save some extra expenditure.

For small area communication such as a room, a playground, *etc.*, it is possible to use sensor modules with WiFi / Ethernet but for keeping eye on far distance rural areas it is not advisable. These places are not fit of the heavy power greedy modules but it needs something which can be easily installed and used for a long duration. There are some ways to reduce the energy consumption by the sensor nodes such as using DARAL [50], *charging sensor batteries over WiFi network* but it adds the restriction on the communication range of the sensor nodes [51]. So, the Low Power Wide Area Network (LPWAN) protocol can be used instead. These protocols help modules to communicate over a long distance using a comparatively small amount of energy, increasing the battery life of the sensors while making them more independent and cost effective in maintenance.

Keeping this analogy in mind, the decision for this thesis was made, to use some low power wide area network protocol for the prototype. Apart from this the most important features of a pollution monitoring system are [27],

1. Continuous data collection
2. Provides location of the collection node
3. Portability
4. Energy efficiency
5. Accessibility
6. Compactness
7. Bi-directional communication
8. Easy maintenance
9. Easy deployment

A detailed comparison among selected technologies is shown in figure 2.1,

Ethernet does not need gateway compare to other protocols, and has a high number of sensors available in the market [52]. High data rate with continuous data flow drains the battery fast and it also requires entire infrastructure which is not very cost friendly and has limited range of communication [36].

WiFi Sensors provide high data rate but low range compared to the Ethernet but it has been used for a long time. Numerous sensor modules in the market support WiFi protocols, which makes it the best technology for in-house usage, but still, the problem of taking sensors outside remains intact.

Cellular Networks 3G/4G are well prepared for the long-range communication [37] by having base stations everywhere and GPRS and GSM communication protocols installed. The subscription for these services are easily available and security of communication is provided by the companies. Waspmote is one of the examples of 3G/4G activated sensors available in the market by the Libelium World [35]. It is a bit of redundant as the subscription needs to be renewed at a regular interval and the base stations should be nearby the sensor nodes. The complex communication protocol adds delays in the data transfer which is a crucial issue in IoT network.Some sensors use cellular networks (3G, 4G) for high data rates and continuous flow. These GPRS [30] networked sensors are capable of delivering continuous data, which drains the batteries, prohibiting the use of heavy sensors in most farther places. In monitoring remote areas, continuous battery change and the cellular subscription renewal is not an option.

ZigBee is a low power consuming [31] technology so, it is fit for IoT network but has a low range of communication. This lowers the chances of ZigBee being used for long distance data communications rather it is being used for remote controls and some in-house sensor controlling protocols to lower the energy usages. Example, SMCDW30-Z [26], temperature and humidity sensor, etc.

WiFi-ah or 802.11ah is 900 megahertz WiFi, which is ideal for low power consumption and long-range data transmission. It's earned the nickname *the low power WiFi* for that

	Ethernet	Wifi	3G & 4G	ZigBee	Wifi-ah	LoRaWAN / LoRa	Z-Wave	SigFox	LTE-m
Gateway needs	NO	NO	NO	Yes, Coordinator	NO	Yes	Yes	Yes	NO, Cellular network
Data rate	Up to 1 mbps	1-135 mbps	110 mbps-1 gbps	250kbps	Up to 347 mbitps	50 kbps	0.1 mbps	300 bps	1 mbps
Power Consumption	High	High	High	Low	Low	Low	Low	Low	Low
Range	100 meters	32 meters	10-15 km	10-100 meters	Extended WiFi range, >32 meters	15 km	100 meters	30-50 Km	2-5 Km
Accessibility	Licensed	Licensed	Licensed	Unlicensed	Licensed, Cellular providers	Unlicensed	Licensed	Unlicensed	Licensed, Cellular providers
Frequency band used	-	2.4GHz, 5GHz	2.6 GHz, 1800 MHz, 800MHz	2.4 GHz	868-921 MHz	EU 863-870 MHz and EU 433 MHz ISM frequency bands. US 902-928 MHz ISM band	868-921MHz	868 MHz in Europe and 915 MHz in USA	2.6 GHz, 1800 MHz, 800MHz,2.1GHz
Bi-directional Communication	Yes, high data rate	Yes, high data rate	Yes, high data rate	Yes, high data rate	Yes, low data rate	Yes, Low data rate	Yes, Low data rate	Yes, Lowest Data Rate	Yes, Low Data rate
Security	-	WPA / WPA2	AES 128 bit	AES 128 bit	WPA / WPA2	AES 128 bit, encrypting data at 3 levels	AES 128 bit	VPN + SSL encryption, devices with private key	AES 128 bit
Communication Module Cost	$10+	< $10+	Depends on network provider	< $4+	Not available in the market	> $6+	< $5+	>$2+	Depends on network provider (higher than LoRa and SigFox)
Applications	Wired to internet so, no wireless communications	Wireless communication possible in limited range	Regular subscription update required, Can be used for near base station communication	Used for in-house communications, short range M2M communication	Extended WiFi range usage possible, can be used with any WiFi oriented communication with easy upgrade	Precision farming, automation, monitoring	Door locks, window / Door sensors, Sirens, etc.	Predictive maintenance, demand forecasting	Object tracking, wearables, utility metering
Sensor Examples	GridConnect: Water Sensors, Temperature data logger	Monnit Cellular Temperature Sensors,	Waspmote Plug & Sense! Smart Water model	SMCDW30-Z temperature and humidity sensor	No support Available yet	Libelium Waspmote Plug & Sense! available in US for LoraWAN, LoRaWAN Temperature Sensor FMLRA-C-32L1-TEM	Aeotec LED Bulb, Aeotec Siren Gen5, Kwikset 910, Aeotec Smart Switch 6	Libelium Waspmote Plug & Sense! available in US for Sigfox, IDIAG Humidity, Atmel's ATA8520-EK3-E kit	Gemalto's Cinterion® EMS31, chip can be tailored on the IOT applications to use LTE-M network

Figure 2.1: Comparision of Communication Protocols

very reason [44]. AH is not available right now for public use but it shows the potential for being used in the IoT networks in the near future. It also has same lacking points like the WiFi protocols but it has a longer range (32 meters) and works in IoT network favorable frequency band *i.e,* 868-921 MHz.

*z-Wave*is not a very new technology but it is being enhanced according to the modern test-cases. It provides high data rate compared to other LPWAN technologies but lacks in the range, 100 m. z-Wave is an ideal technology to be used in the in-house devices such as Door locks, door sensors, Sirens etc. So, there are numerous products available in the market [45].

LTE-m is a cellular network based technology made especially for IoT. It has same dependencies as of the other cellular network technologies. Works with base stations and has

high data rates. The licensed protocol for distant sensors is very promising but cannot be used freely, it is useful for object tracking and utility metering. The installation of this technology on top of the sensor devices is easy using Gemalto's Cinterion EMS31chip. This chip can be tailored on top of the IoT application to use LTE-m network [53].

*SigFox*is an unlicensed communication providing technology, works in its own network. The modules connected to SigFox chips are directly communicating with the base stations of SigFox easily. Around the major continents, SigFox has its base stations, so anyone can connect to SigFox network. SigFox-network architecture can be seen in the figure 2.2. It has four main participants as objects, Sigfox stations (base stations), SigFox Cloud and Customers [54].

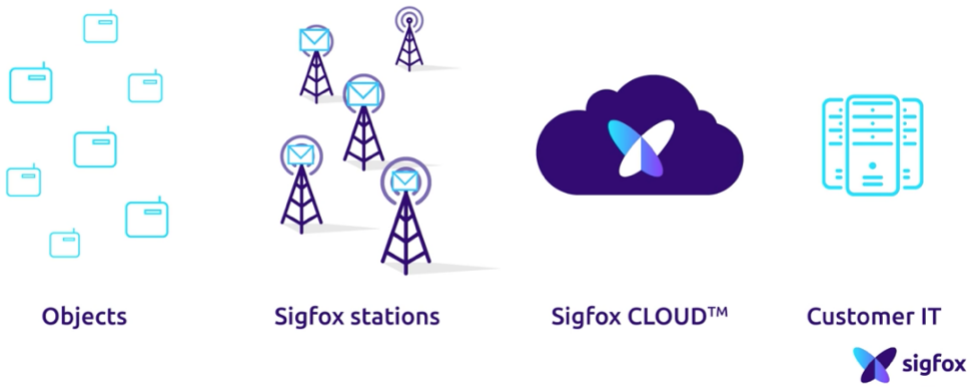

Figure 2.2: Architecture of SigFox Network

As the data is being sent from the objects (IoT devices), it gets received by the nearby base station. These stations are connected to the SigFox Cloud which helps to decode the message and send the data to the respective client. SigFox uses peer to peer encryption not a complete path encryption like other technologies. This helps to lower the data usage and ultimately increases the energy efficiency. In recent technologies SigFox is the most reliable and lowest power consuming network. Having these many positive points SigFox lacks at several other points, SigFox network takes around 10 seconds to send 10 bytes of data because of the extremely low data rate [55]. It does not have any collision avoidance mechanism which is responsible for inference during mobility [56]. Additionally every devices are required to register with the local SigFox network provider for communication, once the device is relocated on different geographical location.

LoRa / LoRaWAN is another evolving technology, the LoRa wireless RF technology provides a long range up to 15-30 miles in rural areas [28]. LoRa would also enable tracking applications in future without to replace additional GPS being used today (important to keep eye on theft issues). The tracking system unleashes additional features in the monitoring system, where one can identify the exact location of the pollution and keep data about it without having to spent extra money on GPS [24].

2.2 Pollutants and Sensors

Pollution monitoring using wireless sensor networks (WSN) is a good research area for a number of researchers because of the increasing potential applications and evolving technologies. WSN uses a number of small, low-priced detectors and which are helping users to see the data projected on-screen using Monitoring Systems in either analog or digital fashion. Constant contamination of air and water is leading to a hazardous atmosphere to live in. So, monitoring and controlling the pollution is a good attempt to make the world safe from malicious particles [62]. It can be predicted that which method would be the ultimate key to control the pollution index but one must have a reliable information source is crucial [61].

Air pollution is termed for the regular contamination of the air with combustion and particulate matters. A long-term exposure to the contaminated air is an important risk factor for lung cancer [64]. Rising CO2, CO, NO2, SO2, PM levels in the air is causing many health diseases which need to be carefully monitored and best-tackling methodologies should be applied on it. PSI provides an index value for these gases to conclude if the environment of the area is polluted or not [61]. Similarly, Water Pollution is another area where the pollution control and monitoring research is going on. To say that particular water body is polluted or not good for flora and fauna, PH and Turbidity are two major indicators to be considered. PH is an indicator of hydrogen ion present in the water, as the pollution increases the carbon level in the atmosphere causes variations and dramatic changes in the weathers, for example, Acid Rain. PH of a water body can fluctuate due to the increased pollution, which can be a great pollution indicator as one of the most devastating side effects of pollution is the change in PH level [65]. On the other hand, Turbidity shows the biological, geological and physical process of water bodies. Turbidity is measured based on the amount of light scattered from the suspended sediments in the water [60].

There are several smart sensors available in the market for example the range provided by Libelium (gas sensor pro and gas sensor) are capable of detecting the gases with highest accuracy possible but it only has the inbuilt memory to run the over the air programming (OTAP) [80], which is similar to the inbuilt memory of an Arduino Uno discussed in section 2.3. The memory and processing requirement for a BC node setup is much higher than the smart sensors which are being offered in the market. Each smart sensors need high compatibility and storage capacity for working with BCs, this can be done by adding a small computing module to them. This can eb achieved with the help of Raspberry Pi (RPI) discussed in section 2.3. The integration with the RPI can only be done if the smart sensor node is compatible to the RPI model. On the basis of BC availability to the sensors there can be two classifications,

1. Sensors with BC support

2. Sensors without BC support

2.3 Hardware System Components

Systems studied for this work involved Raspberry Pi (RPI) and Arduino. These are the systems used to connect the sensors with the LoRa and BC technology for optimizing the battery life of the sensor nodes.

Raspberry Pi 3 Model b is a small but full-fledged computer like system. It does not have inbuilt memory but one can install multiple storage devices (flash memory cards), this helps user to have multiple systems switching just by installing new SDCard. It is a light system and requires less power supply. Figure 2.3 is a RPI, which is not an open source device. This device runs especially designed Linux operating systems, SSH function is also possible in this independent device. The new RPI3 model B has inbuilt WiFi module which enhances the efficiency and independence of the device. As the inbuilt WiFi module works for the average users, it discarded the need of external Wifi module in the Pollution monitoring system. So, in the project no external Wifi module is used.

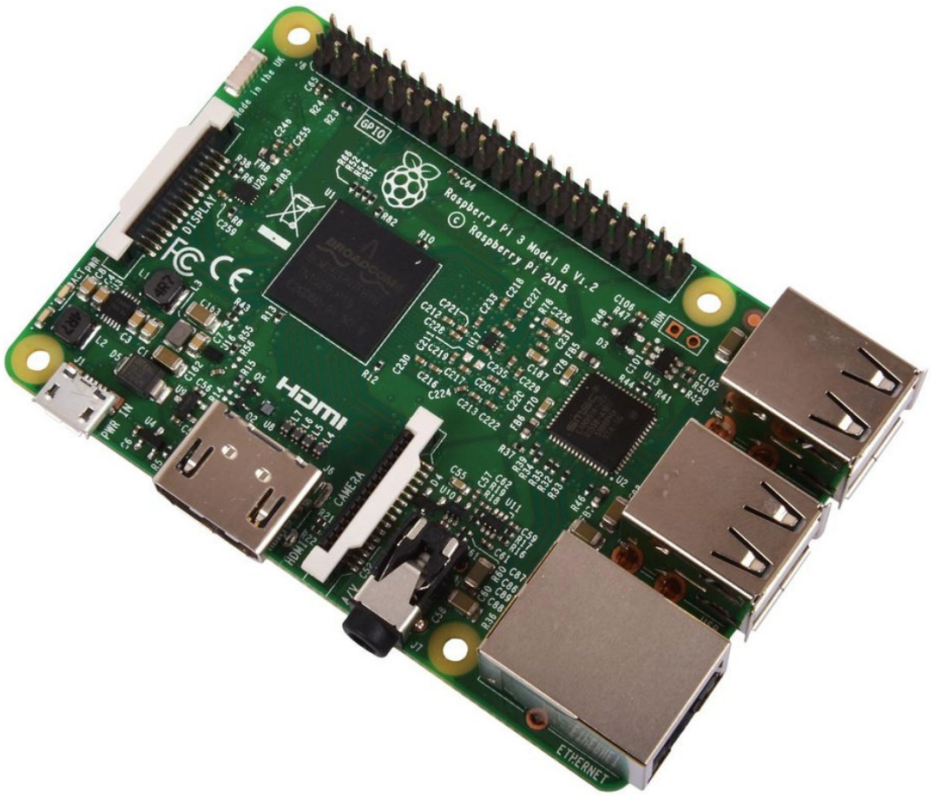

Figure 2.3: Raspberry Pi 3 Model B

Arduino Uno is a micro-controller based on ATmega328, interface between sensor and devices. It is not a full fledged computer but it can pass the sensor data to the devices for different use. This is a recommended system for gathering data from multiple sensors and also supports analog to digital adaptation. It needs external efforts to connect to the internet and needs to be programmed to behave in certain way either in C++ or Arduino.

It has an USB interface to communicate with the device as a serial port. It has 32KB of flash memory pre-installed on it to execute the code. Pin orientation of Arduino Uno can be seen in the figure2.4.

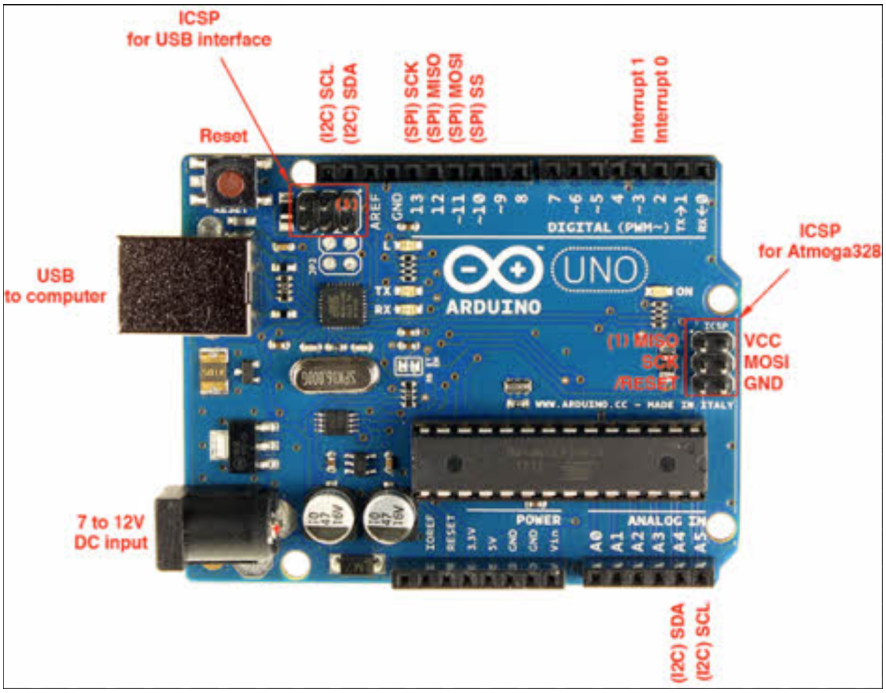

Figure 2.4: Arduino Uno v1.2

2.4 Blockchain Paradigm

Blockchain (BC) is a distributed ledger [1], which helps to hold the data by replicating it on the machines of every node connected to the network. This is also known for its peer to peer system behavior. The major need of BC initially was to stop from double spending. For this, the first company came up into the picture is Bitcoin, a cryptocurrency which helps to get rid of the middle parties and do the direct transactions.

Bitcoin introduced the term "Block-Chain" for the first time to the market. Since then, there have been many reincarnations and versions of the BC with different consensus algorithms. This peer to peer system is a trustless system where transactions are grouped into a block and these blocks are then connected to each other. The blocks are responsible to make the connection and store the hierarchies within them. A timestamp is used to carry on the "proof of work" [1]. These time stamps are used to help the whole network to make sure that no block has been betrayed by the other *i.e,*, no block can be placed on top of the other.

Each block has to be added after the already added once. As per the bitcoins analysis of the continuing work done using BC, there is a new block at every 10 mins [8]. This is good but this is a bad indication also as one can say that it takes 10 -30min [8] to confirm a transaction, which is right. To minimize this wait time and the cost of each transaction many other companies have been researching such as Ethereum, Ripple, NXT, Stratis, Monero, etc.

2.4.1 Working of BCs

A BC is consisting of blocks and hashes, blocks, as we have already seen, is a collection of transactions, a typical block used to contain 1 transaction before, which has now changed to approximately 2500 transactions [9].

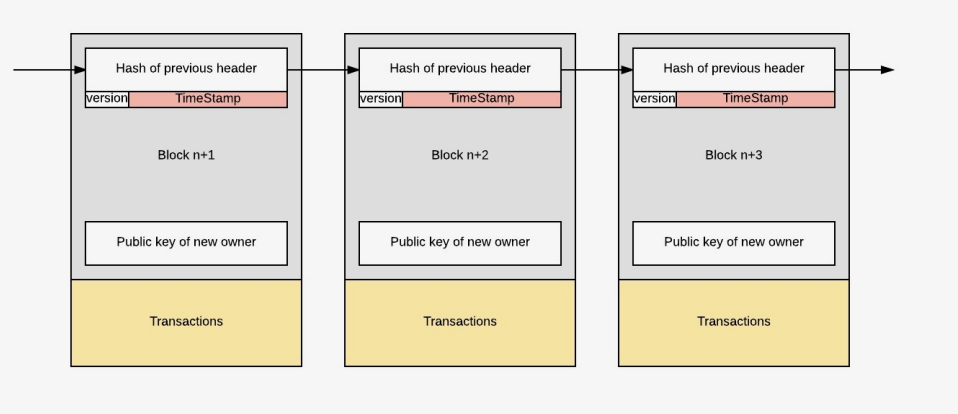

Figure 2.5: Block Header

The Figure 2.5 shows the details kept in a block. A block consists of a block header and the transactions. Each block header contains a hash (public key) address of the previous

block and a timestamp for its positioning in the network. It also contains a version of the block and public key of the new owner. The blocks are always signed by the previous owners to show the authenticity of the transactions. It stores transaction into the body which then can be referenced by the Merkel root reference available in the block header [1].

Hash Function

It is a function which takes input data for any size and creates an output sequence of a fixed length [10]. This function is used in the cryptographic algorithms to create passwords which results into a unique number for addressing a client. In Ethereum [11], hash functions are used to create the addresses of the accounts to enable the safe transactions. These sequences can be used to trace the account who made the transactions and it is so unique that anyone in the world can use this sequence as an address to send the currencies around the world. Example of the hash function can be as [10],

hash(”CoinDesk rocks”) => 7ae26e64679abd1e66cfe1e9b93a9e85

hash(”CoinDesk rocks!”) => 6b1f6fde5ae60b2fe1bfe50677434c88

the difference can be noticed when a small change is made into the original string. Hash functions in bitcoins are used in the bitcoin protocols into the block hashing algorithms called hashcash proof of work function [12].

Proof of Work

Proof of work (PoW) helps to determine attacks or malicious use of computing power in order to disturb the network. This concept was first approached by Hal Finney in 2009 to the bitcoin [13]. POW is not for all but used in many mainstream BC cryptocurrencies. To achieve a validity of a particular transaction BC works with the miners. These are the nodes which work on some given mathematical problems, this is required to be solved in order to add another block into the chain. Nakamoto says that the proof of work is there to make the peer to peer system trustworthy [15]. It is used to make the timestamped network distributed. The process of proving, scans the value which is hashed and produces a numeric expression, now the timestamp and the blocks hashed value must match in order to reach the consensus. Some CPU power is needed to do this work, which is why the transaction has to be paid. Once the proof of work is found the block is added to the chain.

Merkle Tree

Merkle tree is a collection of transactions within a block. This tree holds the hashes of the transactions which in turn linked to the transactions itself.

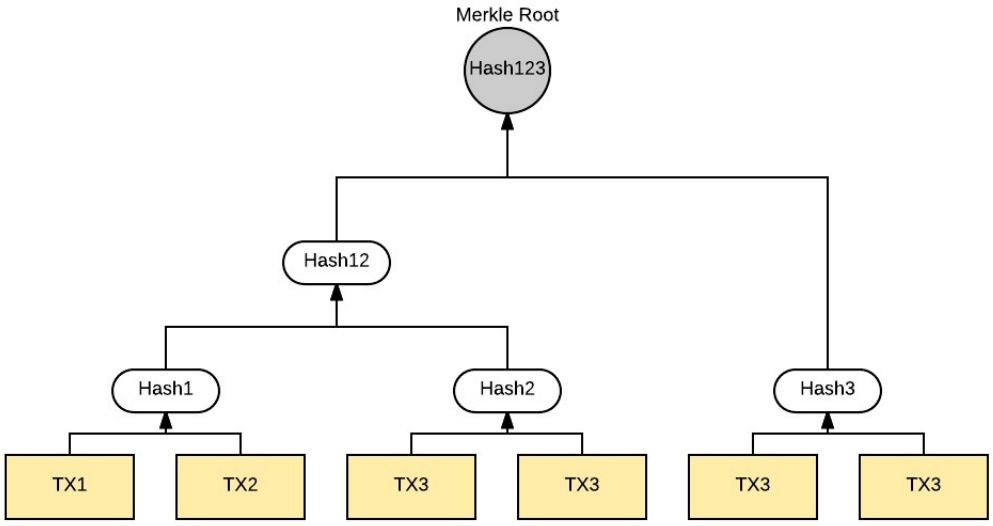

Figure 2.6: Merkle Tree

In the Figure 2.6, one can see that each transaction is hashed and then there hashed values are again hashed together to give a hashed root. This root is then referred into the Block header for accessing the real transactions.

Transaction

A transaction in BC means to transfer a cryptocurrency from one owner to another. This hashed transactions are made without involving the third party to hold the amount in between which saves the time as well as the cost of the complete transaction service. Example Banking sectors and International money transfer [16], which can be made cheaper by using BCs compared to the traditional way.

Smart Contracts

Smart Contracts (SC) are the channel which allows the applications to interact with the BCs. Ethereum accolades the users by allowing them to write their own contracts in a simplest language accepted across the globe with small changes to make the programming light, secure and widely accepted.

A SC 2.7, when deployed over the BC, generates a sequence of codes which is then referred as the address of the contract over the BC. Once it is deployed it can be accessed by calling the encapsulated functions residing inside the contract from the external media *i.e,*, the distributed applications. The ABI is the interface code produced at the same time of deployment. It necessary for the different APIs to allow the communication without facing an authorization issues. Each smart contract is bound with the number of gases

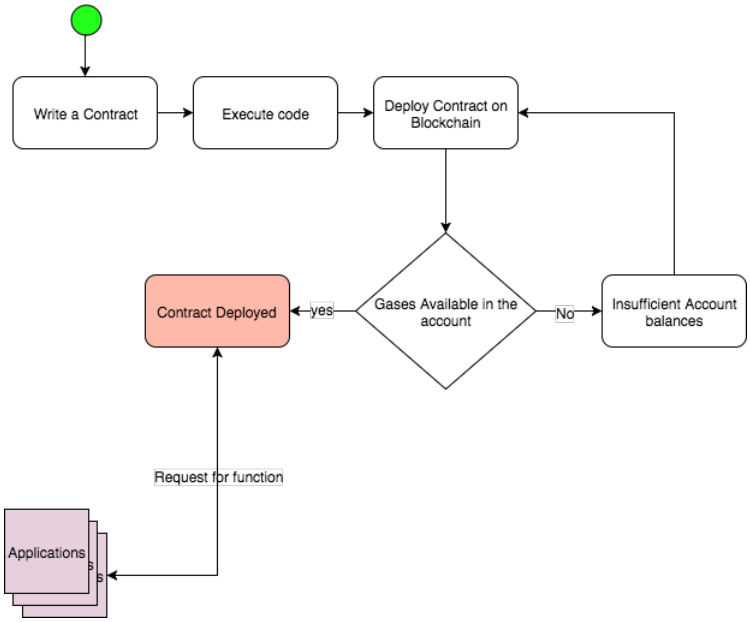

Figure 2.7: Smart Contract Life Cycle

available in the users account for deployment, an important strategy to keep the program light and distant from the vague users. These gases are then to be considered for the execution as it results into the Ether/Wei to be paid by the account to deploy the smart contracts. As per the formula used by Ethereum, transaction fee= gas*cost of gas. There are some standards of the Solidiy programs where one can lower down the gases being used by avoiding the use of complex data structures and overriding.

2.4.2 Blockchains

There are many BC teams trying to make the future of the internet much more trustworthy and low costing than now. Blockchains and Smart Contracts are making the current market more self-executable and trustless as it helps to remove the middle parties from between the consumer and the sellers. There are many other use cases where the BCs does not fit, for those use-cases, it is necessary to go with the traditional way [7]. The goal of this work is to provide a use-case to help explain, how the BC and IoT can work together without involving any trust issues. To understand the working, one first should know what BC means and why it is so important to use it.

Classification and Comparison of Blockchains

The blockchains available in the market, compete over the best transaction fees, security, scalability, and their use cases. There are some of the example listed below with the working and their contribution towards the IoT as a use case. As the BCs increasing in numbers, the race towards integrating IoT with BC has also increased. For using the BC into the IoT network, BC's transactions should be cheap and readily available

for use. Further in the section a detailed comparison over several BC available in the market is given. The comparison and analysis involves market share of the Blockchain cryptocurrencies, just to show the user's trust and future growth possibility of the BCs. All the data is taken in October 2017 from coinmarketcap.com.

- *Bitcoin*

  One of the most trusted cryptocurrencies, which is currently trading at more than $4000 having the market share of more than $67 billion [18]. As it has the most used BC so it has the delay of 10-30 min [14] to validate a transaction, not good for the IoT systems as the sensors are capturing data every second and we need to keep the count of all the data to keep the solution cheaper and faster. Bitcoin is the most secure and trusted BC available today but it is not helpful for the IoT system as it is slow and transaction fees is not cheaper at all, it costs nearly 1 USD ($) per transaction. However, there are many new initiatives coming in to the picture to make Bitcoin BC secure, fast and reliable, such as unbreakable botnet [17], which can solve the security issues of Bitcoin BC.

- *Ethereum*

  Being a decentralized platform, it allows the smart contracts to run, securing the reliability and work according to the program fed. It shares almost $28 billion of market share and trading at $400 [18]. Just like Bitcoin, Ethereum is also a top currency. Ethereum also works on the blockchain, additionally, the smart contract works in favor of the developers. This provides several APIs and programming platform supports for the programmers to build decentralized apps (DApps). DApps are used for communicating with the BC and it allows not only the Ether transmission but also provide support for data storage over the BC. Which helps IoT system to interact with the BC and perform a programmed function on certain triggers. As these functions and triggers are stored on the BC, it is very hard to cheat and hack the functionalities.

- *Litecoin*

  Trading at more than $55 [18]. Uses same BC concept as of the Bitcoin but open source protocol. Which makes the network more powerful and faster than the original core *i.e,* bitcoin.

- *Dash*

  Trading at more than $330 [18]. Same as Litecoin, Dash is also based on the Bitcoins core blockchain with many overcoming security and governance protocols. It has made the transaction much faster than the competitor Bitcoin.

- *IOTA*

  Trading at more mere $0.48 [18]. Apart from being traded at the small value, it is highly secure and fast [19]. According to the white paper released by the team IOTA, the transaction fees are nil. This makes the technology more desirable in the IoT network. It is able to provide a secure and free transaction over the network using its own take on Blockchain and Proof of Work. IOTA is using a network which

is not directly a Blockchain in itself. It is a network of peer connected blocks, that are confirmed by the honest decisions made by the transacting person. In IOTA transaction, there is no miner. Each one who is transacting is supposed to validate four random nodes. If someone is validating a wrong transaction, he will pay by not being verified by another person. It completely works on the trust and honesty. Thus, there is no miner and so the transaction is fast and charges less. It is an ideal condition for the IoT systems. IOTA is made for IoT networks with a specially developed architecture for IoT developers, which can be seen in the figure 2.8. IOTA offers two different architecture for the developers IoT device management

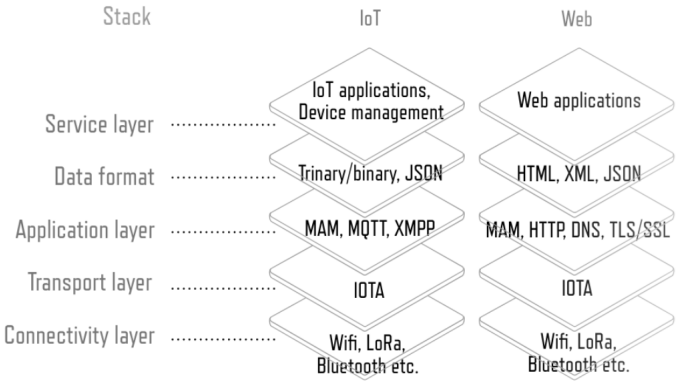

Figure 2.8: IOTA Stack Architecture

and Web application based architectures. Depending on the applications on top of the architecture the intermediate services change.

The recent development in the technology allows provide support for JavaScript API and different services to communicate with the IOTA network called Tangle. Team IOTA is working closely with IPFS for providing large file storage on the network [**?**].

- *Ripple*

  Trading at just $0.19 [18]. Uses the core as a BC with Ripple consensus algorithm. This consensus deals with the distributed database and ledgers, Ripple is a network of servers. An update in a transaction is called as last stored ledger and this ledger is made once every server has accepted the change. In ripple network, it is important to have a proposal to candidate pairing for getting the transaction validated. Each server in the network repeats the process of consensus until it reached above 80% of validity. Hence the Network works faster without involving miners and honesty principles. It is fast but not yet allowed to store the data over the network, which is our goal. However, it is very fast and cheaper than other BCs in the market for cross-border payment transfer [21].

- *Lisk*

  Trading at $5.55 [18]. Lisk is reaching too many now, because of its unique way of working. Lisk's Blockchain works on sidechains and a main-chain concept. The main chain works on the currency called LSK and the side chains are independent

and can be initiated using a JavaScript application. Lisk transactions are cheap as it limits the number of miners to confirm a transaction to 101 active main-chain delegates. The unique part of Lisk is that it focuses on the blockchain application development without using Smart Contract approach. It directly allows developers to store the values on a custom blockchain using a custom token to interact. This approach helps the main-chain to stay away from the spams and makes it more secure from poorly written sidechains. Lisk not only provides an SDK for frontend and backend development but also helps to showcase the application by giving a decentralized directory for applications. It is trying to accomplish what Google and Apple have already achieved by giving users that platform such as App Store and Play store. It has a marketplace where developers can find their own delegates to run the sidechain. In this way, Lisk is providing support for random usages including IoT support [22]. It is also seen that Lisk charges 0.1LSK per transaction which makes it much cheaper than other blockchains.

- *Stratis*

  Trading at $3.85 [18]. As we have seen benefits of some BCs, it would be a cheerful bonus to have an application which can communicate with any of the BC. This is what Stratis is doing. Stratis is a new blockchain platform compared to the other ones we saw. It provides a C# platform to develop the application for any use and it can be deployed on the private chain of the Stratis, keeping Public chain untouched but it should be registered there. Stratis provides a consultancy to the organizations who want to use the platform with s different blockchain, which makes it unique in a way. It uses a version of Bitcoin blockchain called NStratis Bitcoin Full Node with its own consensus algorithm. Minimum transaction charges for NStratis is about 0.1 STRAT, which is cheaper considering the market value of Stratis [23].

Chapter 3

Design and Implementation

3.1 Design Decision

This section explains different design decision taken for creating the PMS. The decisions are taken based on the availability, cost effectiveness, security and reliability of the technologies used.

3.1.1 LoRa

LoRa is a Semtech product and it allows anyone to connect it through different providers such as The Things network, Senet and Cayenne dashboard, etc. LoRa applications have a chip installed on the devices which helps to connect with the providers. LoRa stays under the physical layer of the OSI model the main protocol working in the background is the LoRaWAN protocol. LoRaWAN provides a network where LoRa applications can work. With the increasing number of demands in LoRa devices, the Things network team is continuously working in the area to connect the IoT devices through the LoRa to make them more energy efficient. Architecture of LoRaWAN node can be seen in the figure 3.1,

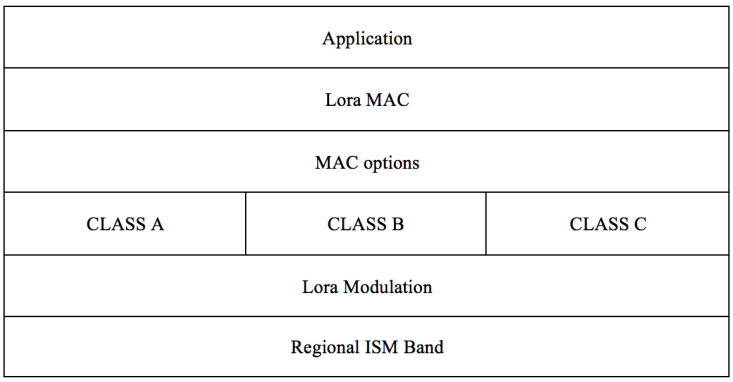

Figure 3.1: Architecture of LoRaWAN Communication Protocol

LoRaWAN has 6 layers as shown in the figure above, the application layer is the physical layer for LoRa devices and the MAC layer provides the LoRaWAN for its working. Three classes of the MAC options help in the trade-off between network latency and battery life [25]. Class A: Most efficient for optimizing battery life and used in almost all LoRaWAN devices. Supports a bidirectional communication in which after each up-link transmission there are two down-link receive windows. It uses random time interval protocol for regulating the uplink and downlink services. So, once a downlink is completed, the server has to wait for some time before another downlink. Class B: In addition to the class A benefits, class B provides additional receive windows at a scheduled time. In order to use this window, the device receives a time synchronized beacon form the gateway. Class C: the receive windows of class C end-devices have a continuous open window, only closed when transmitting, this can lower the battery life of the end devices.

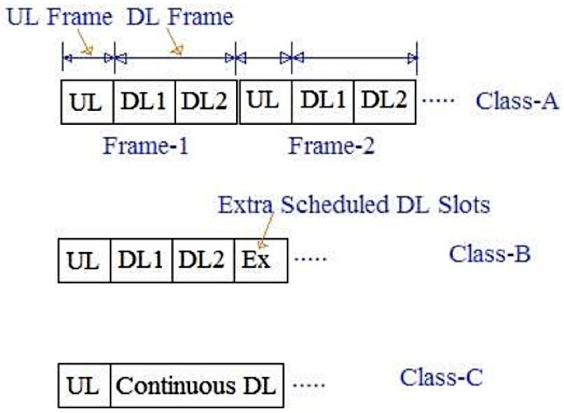

Figure 3.2: Frame Structure of LoRaWAN Communication Protocol [78]

LORA appliances are programmed for energy efficiency so, they deliver data periodically in low frequency, which works in favor of the IoT devices. The LoRa appliances are also low power consuming devices, which helps to reduce the battery usage so the device once installed would not require battery change for nearly 20 years (in a standard weather condition) [24]. LoRa comes under LPWAN systems, which also features SigFox. SigFox is a strong competitor to LoRa, but LoRa has a unique feature in its favor, as it can be paired with any provider without subscription update unlike SigFox network[57]. SigFox is tied with providers and whenever somebody wants to use it he needs to be connect through the base stations of that provider. The number of data permitted to transmit is totally bounded by the subscription rules. [34].

In LoRaWAN, an end to end encryption using AES-128 provides more secure transactions. The future enhancement in LoRa *i.e,* location detection, is a strong point as it is more powerful than GPS in some typical situations [34] such as mobile end-devices.LoRa provides 3 device classes for Bi-directional communication [25], where random time schedule protocol plays an important role in regulating the traffic while transmitting and receiving the data. To match up the competition with the LoRa, the cellular network providers are working hard on different other IoT supporting technologies which will come out in future and some are being used now, for example, LTE-M provided by AT&T in the US [29].

Compare to ZigBee [31], LoRa has lower power consumption and uses a higher quality

of data modulation schemes such as CSS [32], FSK or GFSK. ZigBee's network [26] consists a coordinator and routers to communicate with end devices, whereas LoRa uses its gateway to propagate data directly to the network server. Each class of devices uses different frame structure in LoRa [78] as seen in the figure below, whereas ZigBee has only one generic structure, which lowers the bandwidth usage for Zigbee. LoRa uses RF for data transmission as it has low data rate than the other wireless technologies. Using RF technology enables LoRa to use freely available frequencies to communicate and low data rate helps in using low bandwidths easily, which in return helps to save the battery life of the sensor. ZigBee based systems are not secure as of WiFi-based systems, whereas WiFi-based systems are less secure than LoRa's end to end encryption policy.

In comparison with the cellular network, WiFi network is cheaper with high certification cost and low range [33]. LoRa in comparison uses lower bandwidth than the Cellular network, which helps it to be even cheaper and can be used independently. Considering the plus points of using LoRa chips, Libelium, a sensor distributor is giving upgrades to all of its products to fit the smart city's needs. LoRa powered Waspmotes [35] provide bi-directional data transmission and high-security feature.

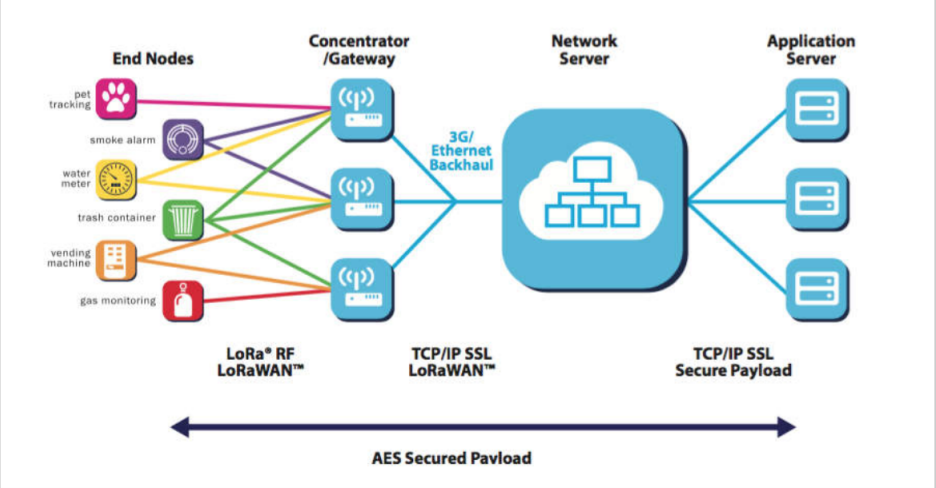

Figure 3.3: LoRa Network Architecture [74]

3.1.2 Sensor Decision

Detecting the pollution indicators such as CO, CO2, PH and Turbidity is sufficient to set an alarm for the monitoring people to start counter mechanisms. Detecting these factors in the remote areas is not possible without using WSN. The proposed solution supports the sensors with low power consuming communication network which is LoRaWAN. Now to make the network more reliable author has used some technologically advanced and low-priced sensors. The classification of the sensors is shown in the figure **??**;

Table 3.1: Pollutant and Sensors Matrix

Pollutants	**Types**	**Minimum outdoor Range**	**Maximum outdoor Range**	**Detecting Sensor modules**
Carbon Monoxide (CO) [109, 111]	Air	9 PPM	>35 PPM	MQ-7,MQ-2,NAP-505
Carbon Dioxide (CO2) [110, 111]	Air	<250 PPM	>350 PPM	MQ-135, MG-811, CozIR LP, TGS4161
Potential of Hydrogen (PH) [108]	Water	6.5	8.5	PH Sensor (SEN0161)
Turbidity [107]	Water	0 NTU	5 NTU	T urbidity Sensor (SEN0189)

As the smart sensor discussed in the section 2.2 is using small sensor modules for getting the data into the system, using same approach was a good decision. The prototype for the solution consists of a mounting board (Arduino) where a small amount of computing could be done. The table shown in the table **??** classifies the sensors depending on the pollutants which can be detected by them. CO detecting Sensors are MQ-7, MQ-2, and NAP-505. MQ-7 is a semiconductor sensor, which works on low and high-temperature variation. In Low temperature, it detects the CO gas present in the atmosphere around it and during high temperature it clears off the other gases sensed by it. It gives a very accurate data about the CO level present in the environment and it is a long life, cheap costing sensor [98].

MQ-2 is useful for gas leakage detection as it can sense many other gases including CO. It is so sensitive that measures can be taken as soon as the change in the atmosphere takes place. The output voltage of the sensor increases with the concentration increase of gases in the environment [67].

NAP-505 comes under new low-cost 3-electrode electro-chemical gas sensor range of sensors. Compact and leak proof and it works specifically for CO detection [66]. CO2 detecting sensors are MQ-135, MG-811, CozIR LP, TGS4161 few to name. Where MQ-135, MG-811and TGS4161 are easily available and being used by many sensing devices present in the market today, CozIR LP is Low power consuming and low-priced sensors

for detecting the CO2 level present in the atmosphere. MQ-135 is a smoke and another gas detecting sensor whereas MG-811 is a specialized sensor for detecting CO2 level in the atmosphere. CozIR LP is an ultra-low power co2 sensor present in the market. GSS team claims that it is the lowest power consuming CO2 sensor present in the market to the date. It runs on a compact battery and only 8 cm in size. It works on the only 3.3V of a power supply, can be used for wide range applications [69].

TGS4161 provides varied voltage output depending on the concentration of CO2 in the atmosphere around it. It needs to be heated at a proper level for getting the accurate value of CO2 level like other semiconductor gas sensors. A 10 min of power supply helps to achieve high precision in the data received from TGS4161 gas sensor [70].

In Water Pollution monitoring systems using a low cost and compact sensors for data collection is not possible. For example, conductivity of drinking water is supposed to be under 800 μS/cm, which is very untrue for swimming pool's conductivity that is greater than 4000 μS/cm for a safer bath. These changes in water quality variables depending on the places encourages the testing teams to collect the samples from the places and then test them inside the nearest laboratories. These logistic costs more than the actual water testing process, which encourages the researchers to think about the compact sensors for detecting the potential of water pollution. Water pollution sensors are normally bigger in size than that of gas sensors and so it needs more power to be operated [71].

For the proposed solution two of the low cost and small sized water sensors were selected, to indicate the potential of water pollution at specific places of the installation. *PH Sensor (SEN0161)* is an easy to install and a pen-sized low-cost PH value sensor. PH2.0 interface to connect with the Arduino modules [72], which makes it an autonomous device to transmit sensed data using LoRa communication module. *TURBIDITY Sensor(SEN0189)* is a Turbidity sensor compatible to Arduino and low-cost small sized water sensor [73].

3.1.3 The Things Network (TTN)

TTN is a middle ground for the data coming from sensors and going to the applications. TTN provides a platform for decryption of data and manages number of different data for applications by routing them towards the calling applications. TTN does the routing functions in decentralized fashion, to do so, TTN has a Router, a Handler and a Network Server all connected via a Broker. Network server is an important feature to communicate with LoRaWAN handling LoRa's specific functionality. A very important part in TTN is Handler, the data going to be downloaded are first decrypted by it. The inner functionality of the TTN can be studied through the figure 3.4. Here G denotes the external LoRaWAN

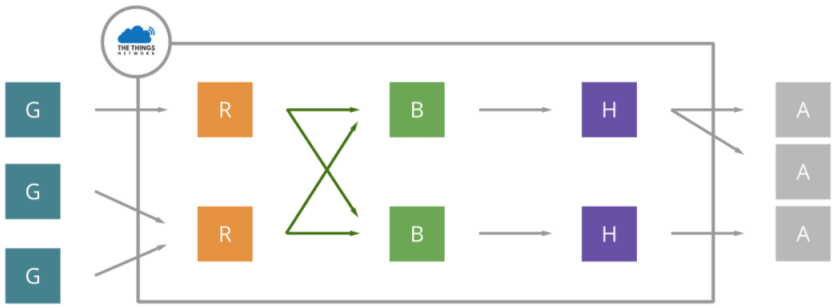

Figure 3.4: TTN Public Community Network [79]

gateway, responsible for collecting the data transmitted from the nearby nodes within 15 KM of range. On the right hand side there are applications A, to visualize the data at user's end. The components shown in the middle rectangle is the back-end of TTN. Which consists of the Router (R), Broker (B) and Handler (H) under a network server. As this configuration is for public domain, private network configuration is also possible by using singular number of components in the network.

TTN works for LoRaWAN by managing the gateways and scheduling in compliance with the European duty cycle. Schedulimg is important so that the load of handling *a transmission at a time* feature of LoRaWAN id distributed among different gateways equally. Next TTN keeps track of the nodes addresses as it is not unique and also the frames and keys of each node. Handler needs to interpret binary data and utilize higher-layer protocols like MQTT (Message queuing telemetry transport).

Limitations of TTN are as it limits the uplink time on air for frequency 868.1 MHz to 30 seconds per day, per device and at most 10 down-link messages per day. It is suggested to have payload restricted to 12 Bytes and interval between the transaction should be kept at least few minutes.

Table 3.2: Comparison of Blockchains

Name	**Average Transaction fee (USD)**	**Average Block-time (m)**
BitCoin [77]	2.433	10.36
LiteCoin [77]	0.0689	2.63
Dash [77]	0.172	2.64
Ethereum [77]	0.821	0.13
IOTA [19]	0	No Blocks
Lisk [22]	0.493 for delegates for running the nodes	0.10
Stratis [23]	0.4690	1 (Can be customized by the side-chain owner)
Ripple [21]	$\approx 2.10^{-6}$	0.035

The table 3.2 is showing a comparison among the enlisted blockchain providers depending on their average block formation time and the cost per transaction in USD. As per the definition of IOTA, the tangle network does not work with blocks and transaction fees. It is free and each individual transaction is verified by the peers.

Stratis has its own method of dealing with the Blocks and Block time, it allows the node owners to customize the block size and block time. A simple comparison among the enlisted blockchains depending on the block sizes is given in the figure 3.5

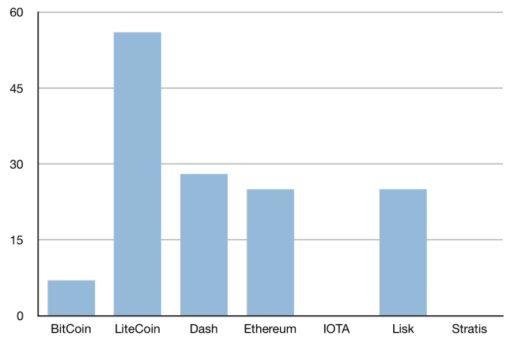

Figure 3.5: Comparison of Blockchains in terms of BlockSize [76]

As Ripple is currently having the highest block size i.e. 10,000 transactions per block, it is not shown in the figure 3.5 to have a clear picture among other BCs. From the figure 3.5, it is clear that Litecoin has the lead and Bitcoin is lagging behind of all other BCs. However, bitcoin scalability improvement program such as Sigwit and lightning network are assumed to increase the block-size up to 50,000 transactions per second in near future [76].

Performance and Security of Different Types of Blockchains

There are two major types of blockchains as Public, Private and Permissioned, depending on the cases each of them is used.

Public Blockchain used by almost all of the cryptocurrencies, anyone can mine and access the transactions. These are good for having high transparency and validation but low security.

Private/Consortium Blockchain is implemented by many organizations where the consensus are controlled by one node and read permission is only given to people belonging to the same organization but it can be made public. The write permission is restricted strictly to the members only. The synchronization time of a node with the network in private BC is much faster than the public BC, which saves delays. The transaction cost is also cheaper than public BC.

The main issue with the public blockchain is the scalability, currently, Ethreum and BitCoin, for example, can only process 3-20 transaction per second and the delays are much higher. The Block-size of BitCoin is now proposed for extending from 1MB to 8MB for incorporating more transactions in one block.

The performance of Private blockchain is much higher and more secure compared to the public blockchain. An experiment entioned in *Xu et al* [75], using Ethereum private BC, showed that around 15000 transactions were processed in 41 seconds. That is 366 Transactions per second, which is around 18 times faster than the public BC (3-20 transaction per second).

The privacy in Public BC is not guaranteed, so it is important to have some external privacy system on top of the Public BC, such as encryption of data before transmission. Finally,as in comparison to all of the other BCs seen in section 2.4.2, Ethereum has shown consistent growth and has a better developer and technical support. So, Ethereum was chosen for this prototype.

3.2 Implementation

3.2.1 System Implementation Models / Approaches

The proposed solution to the problem discussed in the section 1 involves both BC and LoRa technology. The data flow diagram in seen in diagram 3.6, shows the overview and different approaches that can be used to implement such a system. Different approaches mentioned in this data flow are elaborated further in this section.

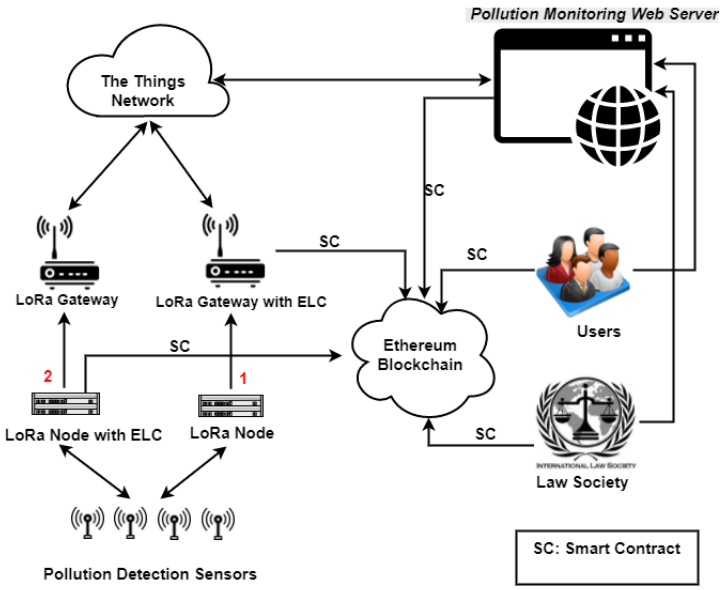

Figure 3.6: Data Flow Diagram for the Proposed System

1. *Approach 1: IoT sensors connected to LoRa boards (LoRa sensor nodes) connected to BC with an ELC installed on the LoRa sensor nodes*

   The solution architecture in figure 3.7 shows a detailed overview of the system. Here the data about the surrounding environment is first captured by the sensors mounted on the LoRa nodes. These nodes are capable of communicating over LoRaWAN using installed LoRa shield and precised antenna.

   The Data is encrypted and sent over the LoRaWAN, which is then received by the nearest LoRa Gateway. For this prototype, a multi-channel gateway is used. Only one channel can work at a time, it receives the packets and then forwards them to an external LoRa network service called The Things Network (TTN). TTN decrypts the data and makes it available for downloading or to be forwarded to somewhere else (as it does not stores the data for a longer period). For reading and visualizing the data over the web application called Pollution Monitoring System (PMS) data is downloaded and stored into the local database MongoDB.

   While receiving the data from TTN each data is filtered through and stored on the Ethereum BC before storing them into the local database. The check on data is made by executing a Smart Contract on the BC, this helps users to check whether any

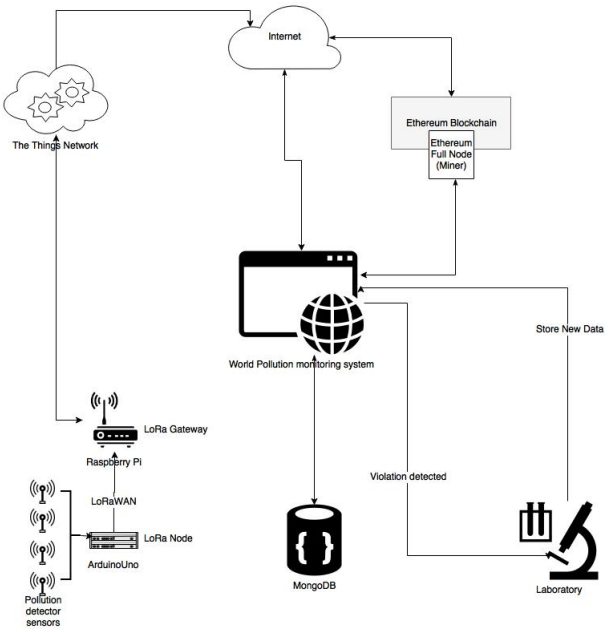

Figure 3.7: System Overview (with LoRaWAN)

of the sensors has detected some violations or not. Once the violation is detected, the Pollution Control Laboratory staffs can then decides to collect and test the real world data samples and then updates it into the system. This data is again filtered and stored over the BC and local database. The web application allows user to monitor the changes in the environment by using inbuilt chart and maps. The application is secured by the user authentication so that no person can edit the local data without permission. As the data is stored over distributed network changes can be tracked down easily.

2. *Approach 2: IoT sensors connected to Full nodes via WiFi or LoRa*

   There is another variant to the system which uses the Ethereum Light client and Wifi module for data transmission as WiFi technology is also evolving to support IoT devices. Since the LoRa network added with TTN has limited Air Time, this solution is more credible with continuous data updates from the sensors with minimum delay time from ethereum BC. For the prototyping purpose a private network using Light client node and a Full node as a miner is established which allows for faster and cost less transactions. The system design can be seen in the figure 3.8

   The Design decision is taken on the basis of Scalability and Reliability of the system. Arduino Uno is used to communicate with the sensors which provides the flexibility for increasing the number of sensors at any point of time with small input and output modification in the Arduino Sketch code. In the case of WiFi module the data is read through the serial port into the BC via NodeJS and Web3 packages installed on the RPI 3 model B system. Whereas the data is transmitted directly form Arduino to the Lora Gateway through the LoRa Shield installed on the Arduino Uno module (called LoRa node). In both the cases the data is highly secure as it travels from one device to another.

   In first scenario the data is Handed over to the BC to convey it to the Web Appli-

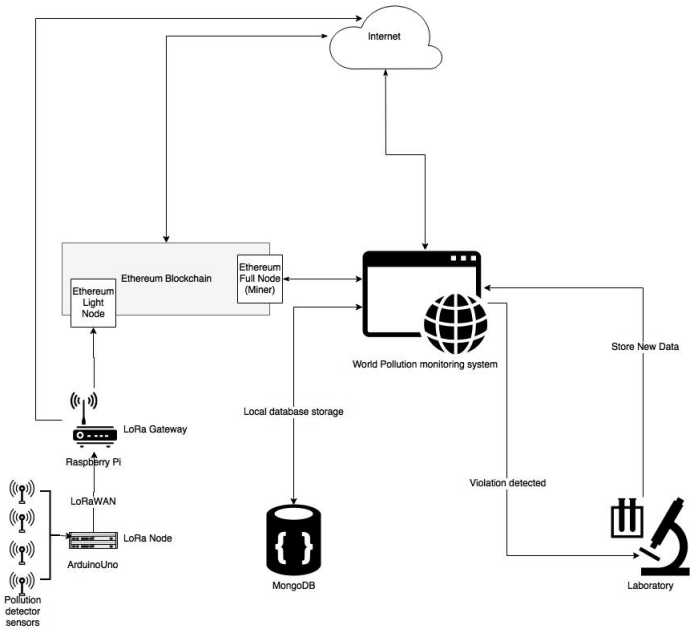

Figure 3.8: System Overview (with WiFi Module)

cation by the Wifi network which is using WPA/WPA2 encryption as seen in the table shown in figure 2.1. For the second case LoRaWAN (supports AES 128bit 3 level data encryption) is used to convey the data to the TTN network and thereafter TTN network provides a highly secured tunnel to download the data into the Local Storage. Since the BC and the WiFi and LoRa Network are trusted and widely used technologies the complete system is fully reliable in terms of the performance.

3. *Approach 3: LoRa sensor nodes connected to BC with an ELC installed on the LoRa gateway*

   The design shown in figure 3.9 shows an accumulated approach where the data security is provided through out the process. At first the data is protected through the LoRaWAN and thereafter by the BC. The TTN network works for decrypting the values received from the LoRa node in terms of Payload. In this design the data comes form the sensor node via LoRaWAN and being accepted by the LoRa Gateway. Once this distance is traveled, The system employs TTN network for decryption which is then again received by the LoRa Gateway and stored on the BC using the smart contract already deployed over the network. The Web Application is able to reflect all the changes made on the blockchain if the transaction is correctly executed from the LoRa Gateway (Ethereum light client).

3.2.2 Sensor Node Setup

The Setup can be seen in the figure 3.15. Firstly the sensor nodes are attached to the breadboard in serial on the breadboard. The power and ground (GND) pins of the sensors are connected to the positive and negative pin of the breadboard so that the battery can be connected to the circuit for providing power to all of the sensors from one point. This

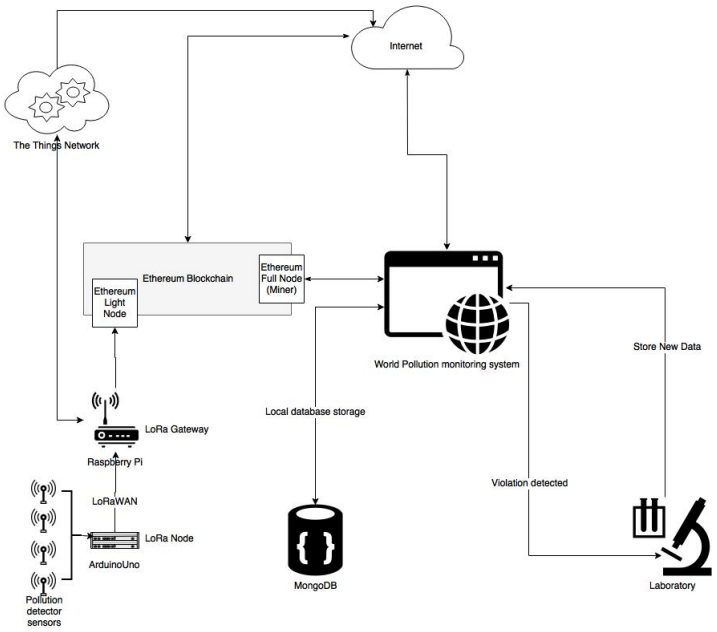

Figure 3.9: System Overview (with WiFi Module and TTN)

arrangement makes the battery replacement and monitoring easy. The analog pins of the sensors are connected to the A0, A1, A2 and A3 IO pins of Arduino Uno. The 5v pin on the Arduino board is connected to the positive pin and GND is connected to the negative pin of the battery (9v) via the breadboard. All connections between the sensors and the Arduino board are made using jumper wires. Once the connection was done an Arduino sketch was uploaded onto the sensor node to send the data to onto the LoRa network. The data is sent over the network in Bytes, the conversion from *integer* to *Byte array* can be seen in the code snippet 3.1. The sketch code is build using the IBM LMIC(LoraMAC-in-C) library [113]. The decryption function stored on TTN can be seen in the code snippet 3.2

Listing 3.1: Payload Data Conversion (*integer* to *byte array*)

```
1    int16_t COval = (int16_t)CO;
2    int16_t CO2val = (int16_t)CO2;
3    int16_t turbidityval = (int16_t)turbidity;
4    int16_t PHval = (int16_t)ph;
5    //shift bits and store the value in bytes
6     byte data[9];
7     data[0] = CO2val>>8;
8     data[1] = CO2val & 0xFF;
9     data[2] = COval>>8;
10    data[3] = COval & 0xFF;
11    data[4] = turbidityval>>8;
12    data[5] = turbidityval & 0xFF;
13    data[6] = PHval>>8;
14    data[7] = PHval & 0xFF;
```

Listing 3.2: Payload format on TTN to get (*integer* from *byte array*)

```
1   function Decoder(bytes, port) {
2     // Decode an uplink message from a buffer
3     // (array) of bytes to an object of fields.
4     var CO2=(bytes[0]<<8)|bytes[1];
5     var CO=(bytes[2]<<8)|bytes[3];
6     var turbidity=(bytes[4]<<8)|bytes[5];
7     var PH=(bytes[6]<<8)|bytes[7];
```

```
8       return{
9           Carbondi:CO2,
10          Carbonmono:CO,
11          Turbi:turbidity,
12          PHvalue:PH

13      };
14  }
```

3.2.3 Implementation of the Web Application

Web Application is built on top of the NodeJS libraries and it majorly uses Meteor framework for the front-end. The back-end is built using MongoDB, which uses JSON to store the data locally. Once the Meteor installation is done on the system a local DB is automatically generated which is secured by the Meteor library internally. NodeJS supports Web3 Library for the communication with Ethereum network. The application uses libraries for the Graph and World Map generation. The Application allows the read operation without any permission but for updating the received data and pollution standards users need to authenticate themselves. The world map shown on the front-end figure 3.10 is made using the JVector JavaScript library [81], the files are added to the system after downloading them, to make sure that no future enhancement from the owner of the library affects the current setup. The tooltip functionality of the library is used for showing the pollution level violations of the country. If the tooltip background color is "RED" means the country is facing the violation for some pollutants but if the color is "GREEN", it indicates that the country is keeping the pollution under control and no violation is found. The values of recorded data is directly liked with the BC, so that the free users can get the accurate data without prior authentication. A click on the country in the map, asks for the region selection and returns the average pollution value calculated recently for that particular region.

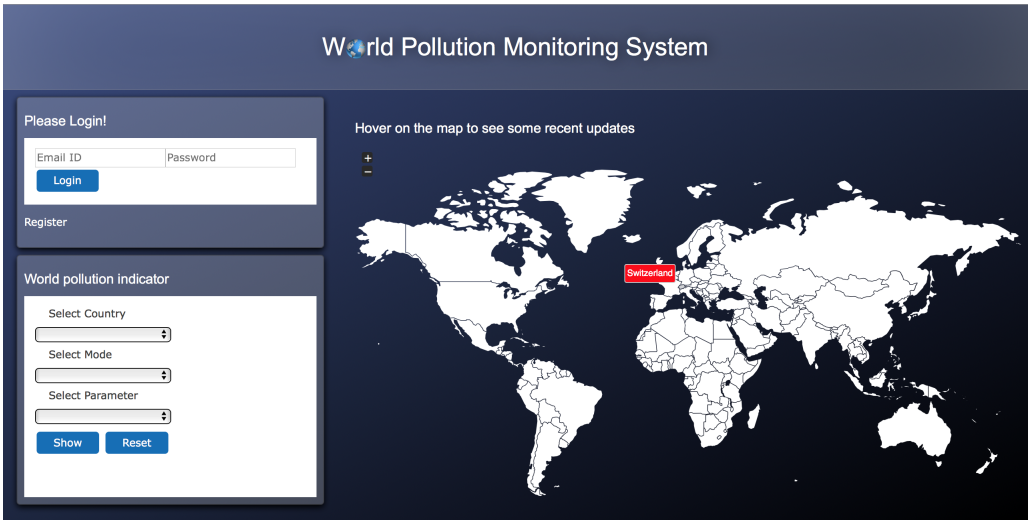

Figure 3.10: The Public Page of Front-end

The chart shown in the figure 3.11 is made using the public library chart.js, the usage and examples are explained on its website [82]. The chart is used to show the historical

data about the pollutant of selected country present on the left hand side of the web page. The Histogram makes it clear for users to take action depending on the analysis presented to them. Natural and artificial changes in the environment over time can be judged through this analytic representation. The following figures 3.12, 3.13, 3.14 shows the tabs presented to the users once they have authenticate themselves. The manipulations and amendments to the existing data can be done through these tabs. The actions are stored on BC as well as in the LocalDB for faster and secure transactions.

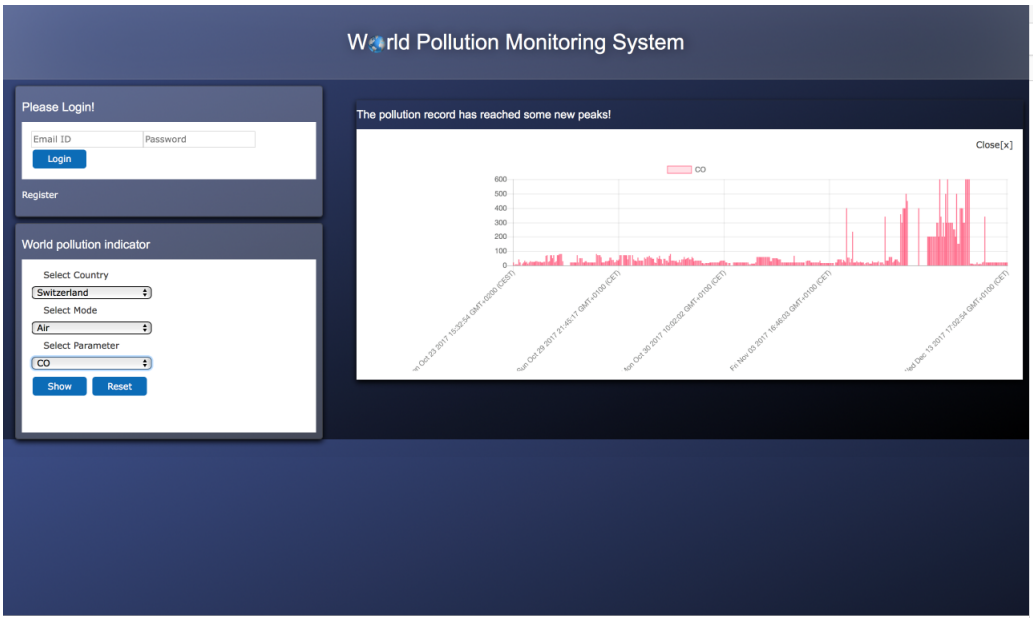

Figure 3.11: The public Page of Front-end Showing Historical Data

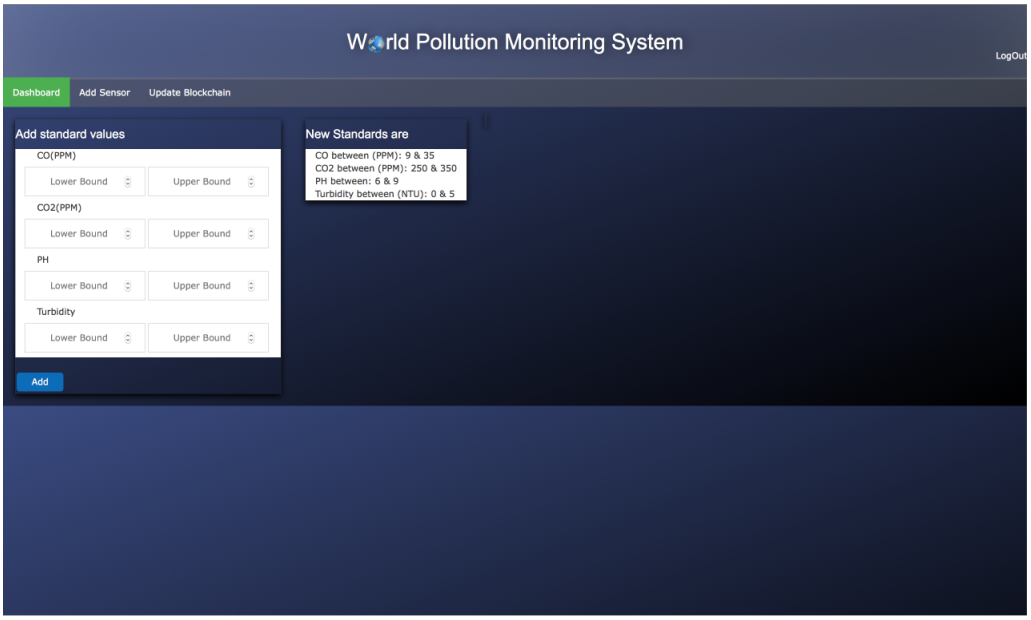

Figure 3.12: update the pollution standards

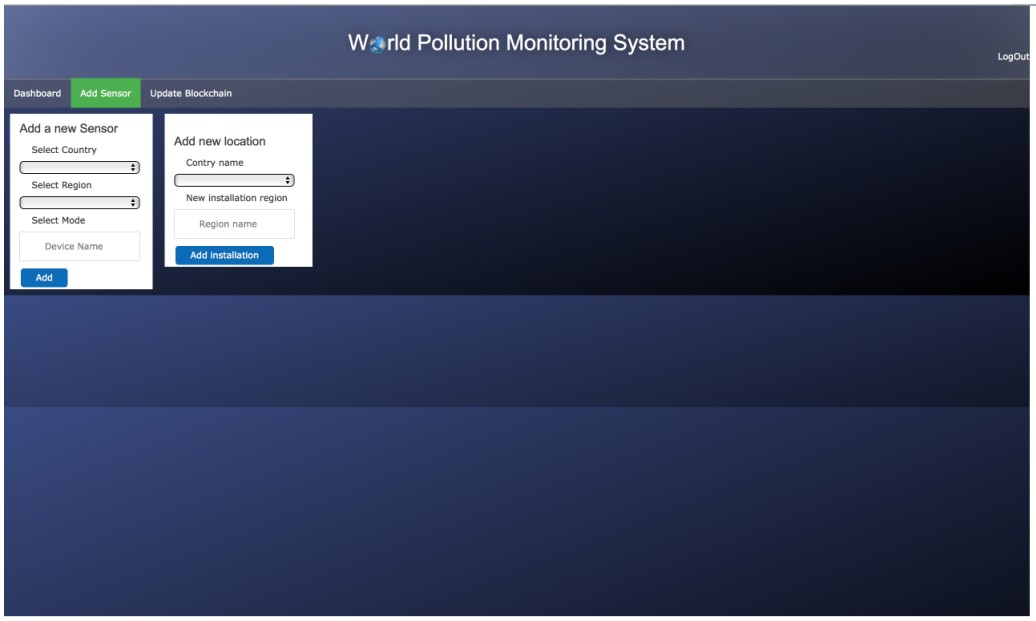

Figure 3.13: Add new installation details

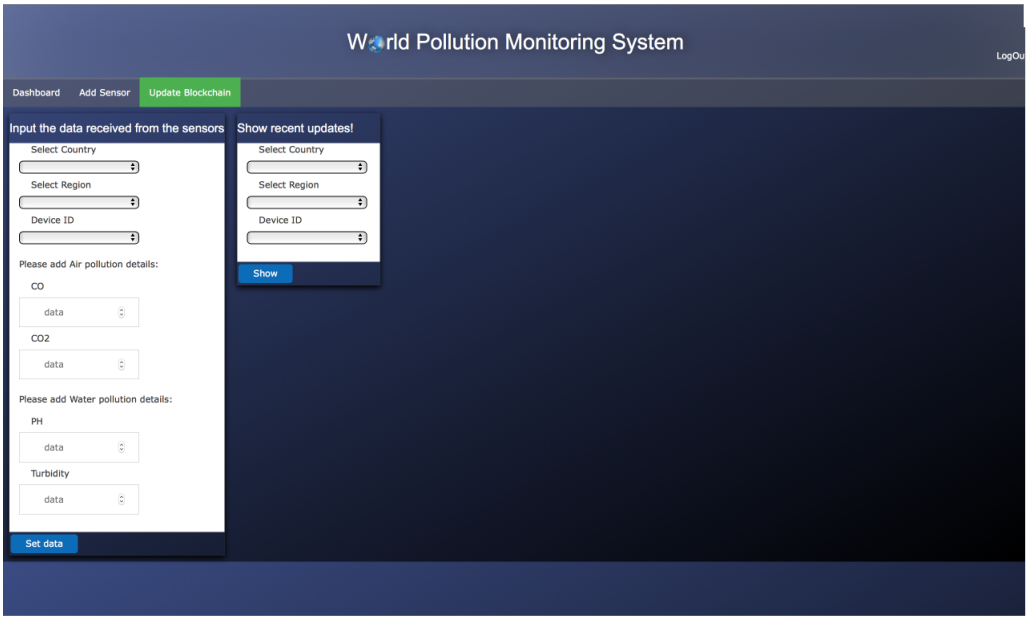

Figure 3.14: Update the data over BC and LocalDB manually

3.2.4 Setting Up a Private BC for Ethereum Light Client

For prototyping the complete solution a private BC network is used, so all the data received from the sensor nodes are directly transacted onto the BC. In the case of using Public BC where the transaction cost is high and not affordable in some cases. In this scenario only read operation is suggested to be performed and the comparison can be done on the node level only before sending the data onto the BC through the Ethereum light client. A small update in the JavaScript file running on the Light Client node (RPI) can update the system for the public economically expensive scenarios. Now the system would only sends the data onto the BC which has violated Governments Pollution Standards updated in the deployed Smart Contract.

For setting up the private network for Ethereum, the publicly available library [83] and install instructions were followed to install Etehreum node over the RPI. During download of the Geth repository for Linux platform, ARMv7 architecture was used. The very small difference between installing light client and full node is the mode of synchronization. If the Geth node is called using the –testnet, –rpc, etc. the node runs with full synchronization which means it needs the data storage capacity comparatively more than the capacity of the RPI. Hence, the installation over the RPI must be called using the mode of synchronization as –light. On the full node which is going to serve the client (light) node it is mandatory to use the –lightserv parameter with the number of clients it should respond to as the number is high more attention towards the light node is paid by the full node.

The APIs on the light nodes does not include the Miners and that is how the load on the on the light client is shaded away. The mining task is done by the full node connected to the network and listening to the light client. So whenever the Transaction is sent from the light client it needs to be mined by the listening full node (peer). The private network is setup by adding the peers to the nodes and the network id should be kept same for accessing the data deployed by any of the peers in the network. Once the private network is setup the Smart Contract needs to be deployed.

3.2.5 Smart Contract Deployment

The Smart Contracts in terms of Ethereum are written in solidity, Ethereum provides a free platform [84] for deploying the contracts over the network. Remix platform was used to deploy the smart contract over the private network in this solution. The Contract includes 4 major functions to read and update the pollution standards and data received from the sensors.

Listing 3.3: The Pollution Monitoring SC

```
1 pragma solidity ^0.4.10;

2 contract setStandards{
3     uint16 Colow;
4     uint16 Cohigh;
5     uint16 Co2low;
6     uint16 Co2high;
```

```
7     uint16 Turblow;
8     uint16 Turbhigh;
9     uint16 PHlow;
10    uint16 PHhigh;
11    bytes32[] public message;
12    uint[] public index;
13  struct receiveddata{
14     bytes32 Country;
15     bytes32 Location;
16     bytes32 deviceID;
17     uint16 CO;
18     uint16 CO2;
19     uint16 Turbidity;
20     uint16 PH; }
21  struct countryflag{bool flag;}
22  function setStandards() public{
23        Colow=9;
24         Cohigh=35;
25         Co2low=250;
26         Co2high=350;
27         Turblow=0;
28         Turbhigh=5;
29         PHlow=6;
30         PHhigh=9;}
31  function updateStandards(uint16 data11,uint16 data12,uint16 data21,uint16 data22,
    uint16 data31,uint16 data32,uint16 data41,uint16 data42) public returns(bool success)
32   {
33         Colow=data11;
34         Cohigh=data12;
35         Co2low=data21;
36         Co2high=data22;
37         PHlow=data31;
38         PHhigh=data32;
39         Turblow=data41;
40         Turbhigh=data42;
41         return true;}

42    mapping(bytes32=>countryflag) public cdata;
43    mapping(uint=>receiveddata) public data;
44  function retrievestandards() public constant
    returns(uint16 ,uint16 ,uint16 ,uint16, uint16 ,uint16 ,uint16 ,uint16 )
    {
45   return(Colow,Cohigh,Co2low,Co2high,Turblow,Turbhigh,PHlow,PHhigh);}
46  function showflag(bytes32 Country) public constant
    returns(bool j)  {return cdata[Country].flag;}
47  function compare(uint16 x11,uint16 x21,uint16 x31,uint16 x41) public returns(bool
    success)
48    {
49         bool i=true;
50         message.length=0;
51         if((x11<Colow||x11>Cohigh) &&x11>0)
52         {     message.push("CO");
53            i=false;}
54         if((x21<Co2low||x21>Co2high)&&x21>0)
55         { message.push("CO2");
56            i=false;}
57         if((x31<Turblow||x31>Turbhigh)&&x31>0)
58         {message.push("Turbidity");
59            i=false;}
60         if((x41<PHlow||x41>PHhigh)&&x41>0)
61         { message.push("PH");
62            i=false; }
63         return i;
64    }
65   function save(bytes32 Country,bytes32 Location,bytes32 deviceID,uint16 x11,uint16
    x21,uint16 x31,uint16 x41) public returns(bool success)
67    {
68         bool i=compare(x11,x21,x31,x41);
69         if(i==false){//comment this line for storing all the data
70         data[index.length].deviceID=deviceID;
71         data[index.length].Country=Country;
```

```
72          data[index.length].Location=Location;
73          data[index.length].CO=x11;
74          data[index.length].CO2=x21;
75          data[index.length].Turbidity=x31;
76          data[index.length].PH=x41;
77          cdata[Country].flag=i;
78          index.push(index.length);
79          //}//comment this line for storing all the data
80          return true;
81          }
82   function retrievedata(bytes32 Country,bytes32 Location,bytes32 deviceID) public
    constant
    returns(bytes32 x,uint16 y,uint16 z,uint16 k, uint16 j,bytes32[] f)
83     {
84          x=deviceID;
85         for(uint i=0;i<index.length;i++){
86           if((data[i].Country==Country)&&(data[i].Location==Location)&&(data[i].deviceID
    ==deviceID))
87          {y=data[i].CO;
88          z=data[i].CO2;
89          k=data[i].Turbidity;
90          j=data[i].PH;}}
91          compare(y,z,k,j);
92          return(x,y,z,k,j,message);  }}}
```

The data received from the sensors are subjected to comparison with the standards available at that time and a message about the violation is released. For deployment of the contract shown in the code snippet 3.3 over the network, one needs to first deploy connect the remix portal to the IPC address on which the Geth is running and then the miners needs to be running on the node. Once the smart contract is ready, it can be deployed on the network through the remix portal online (The account should have enough balance for creating the contract on the BC). In some other case offline SolC compiler is also used. In the presented solution the remix portal was used for the smart contract deployment. The contract is extracted into the web application using the Web3.js module. This allowed to read and write the data over the Smart Contract by following the regulation mentioned on the Web3JS website [85]. Web3 gives an interface between the NodeJS server and the Ethereum node.

3.2.6 Sensor Node Setup

The figure 3.15, shows the sensor node installation done for the project. This node consists of 4 sensors and these are mounted on a breadboard. The breadboard is powered using the batteries (atleast 9V). The sensor module is connected to the Freeduino Uno module which runs the Arduino sketch for regulating the IO of the sensors. The Analog data is read by the Arduino Uno and it is then converted to a byte stream for LoRa transmission. The LoRa Shield is embedded on to the Freeduino Uno for further transmission of the data over the LoRa network. The sensor node comprises of several independent modules which are brought together to work in a synchronized fashion. The modules are numbered in the figure 3.15 according to the following list.

1. CO Sensor
2. CO2 Sensor

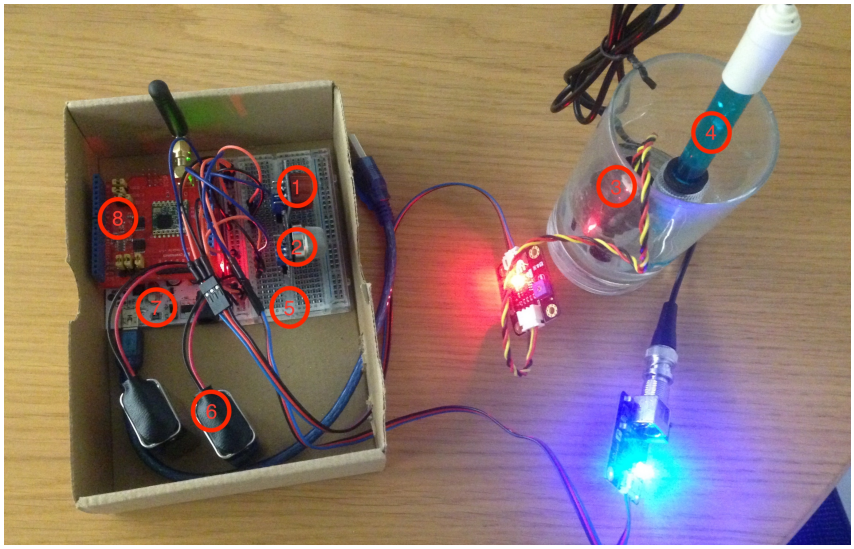

Figure 3.15: LoRa Embedded Sensor Node

3. turbidity sensor
4. PH sensor
5. Breadboard
6. Battery (9V)
7. Freeduino Uno
8. Dragino LoRa Shield

The total cost for the Sensor node setup can be seen in the table 3.3.

Table 3.3: LoRa Node and Sensor Cost

Name	**Supplier**	**Price (CHF)**
MQ-135 (CO2 sensor module) [97]	PLAY-ZONE	16.90
MQ-7 (CO sensor module) [98]	PLAY-ZONE	15.90
Turbidity Sensor module [100]	Mouser.ch	9.68
PH Sensor [99]	Mouser.ch	28.84
Jumper Wires m-m [106]	BoxTec AG	9.75
Dragino Lora Shield -868 MHz V1.4 [102]	Bastel garage	22.90
Arduino Uno R2 ATMEGA328P [103]	YAMPE Gmbh	14.80
Bread Board Clear 8.2*5.3 cm [101]	BoxTec AG	6.20
Primery battery 9V 6LR61/9V + Cable connection [104, 105]	DISTERELEC AG	4.81 + 0.84
Total Cost		**130.62**

3.2.7 LoRa Network Setup

- *LoRa Node Setup*

  LoRa node can be seen in the figure 3.15, uses the LoRa Shield for sending data over the LoRa network. The setup is done by following the manual provided by the Draguino team [102]. The Draguino shield is simply mounted over the Freeduino Uno and the Arduino software module is used for uploading the Arduino sketch into the Freeduino Uno.

- *LoRa Gateway Setup*

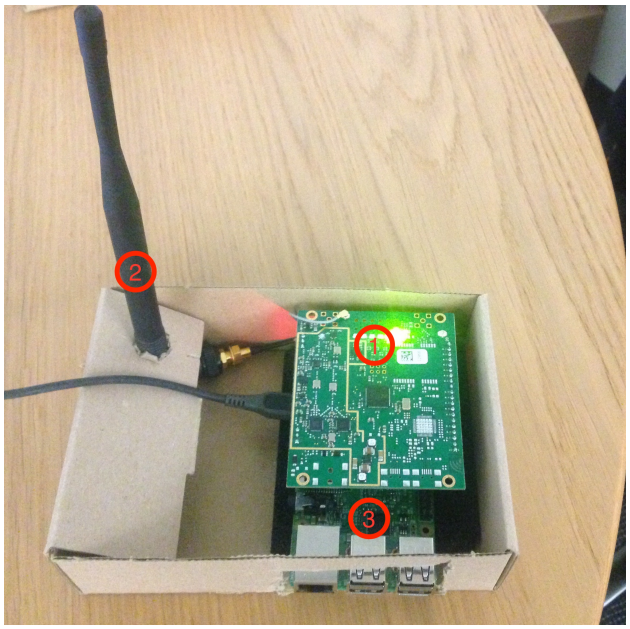

Figure 3.16: LoRa Gateway

The Zero to LoRaWAN in a weekend project [87] was followed for installing the gateway for LoRaWAN. The gateway requires the WiFi module called Backhaul was replaced with the inbulit Wifi module of RPI3 discussed in section 2.3. This WiFi module is used for connecting the gateway to TTN for data decrypting and further accessibility. During gateway installation, it is recommended that the antenna should be connected to the IC880A LoRa Concentrator correctly before installing onto the RPI model 3. The modules are numbered in the figure 3.15 according to the list above. The total cost for the Sensor node setup can be seen in the table 3.4. The Gateway comprises of three major modules namely

1. IC880A LoRa Concentrator
2. Antenna (868.1 MHz for Europe)
3. RPI model 3

Table 3.4: Cost of LoRa-Gateway Setup

Name	**Supplier**	**Price (CHF)**
iC880a concentrator board [90]	IMST web shop	178.86*
Pigtail for antenna [92]	IMST web shop	7.50*
Raspberry Pi 3 model b [93]	Pi Shop	38.90
Antenna [91]	IMST web shop	7.50*
Sandisk MicroSD card [94]	DISTERELEC AG	8.05
RPI to ic880a interface [95]	Tindie	4.88
EDUP 802.11 b/g/n 150 Mbps Wireless USB adapter [96]	Pi Shop	11.90
Total Cost		**247.10**
* Shipping and Extra cost applicable		

- *TTN Integration*

  TTN provides many different types of web application integration for data migration [88]. As TTN only stores the data for few minutes (until the page is refreshed), it is important to get the data stored elsewhere. For migrating and accessing the data "Data storage" integration is used in this prototype. The Data storage integration works synchronously with the Swagger Storage services. TTN supports this storage by allowing to store the data only for 7 days. An HTTP call is made from the web application to this service by passing the authorization key as a part of header. The recent data stored by TTN is called through this HTTP call and shown on the dashboard of the front-end 3.12. Each time the data is checked by the user, it gets stored into the LocalDB with the time when it was received from the LoRa gateway. The time entry helps to keep the data accurate and free from unwanted replications.

3.2.8 Setup Ethereum Light Client on LoRa Gateway

The LoRa gateway is used for transmitting the received data from the node over TTN, once the gateway is registered with TTN network. For the solution prototype the Ethereum node was installed on the RPI used in building the LoRa gateway. As the Lora gateway works for receiving the data from the node, it is mostly busy with the Uplink and Downlink messages, but the RAM and the installed memory on the RPI is not used completely so any other system with less RAM consumption can also be run on the system simultaneously.

To make the prototype more efficient a light client of etherum BC is installed over the RPI of lora gateway. The Ethereum node installed on the RPI with –light mode synchronization leads increment into the RAM usage by additional 10% without affecting the

gateway's original work. The ELC is a light mode synchronization of the Ethereum BC, the BC setup is mentioned in the section 3.2.4.

The received data once sent over TTN then decrypted and sent back. This data is retrieved by the gateway again using the NodeJS server installed on the gateway. A JavaScript file (interface) built using TTN-NPM [89] and Web3 libraries. It is written for sending the data received the gateway to the BC. Whenever the transactions are sent, like a normal Etehreum light client, they await for the miners to get validated (mined). Once the SC is updated on the BC, it is then can be updated through the dedicated web application (PMS).The transactions after mining are updated and validated to the BC depending on the functions written in the deployed SC.

Chapter 4

Evaluation

4.1 Comparison among the Proposed Approaches

The evaluation is done on the basis of 3 major criteria (Security, Reliability and Cost) and by answering few questions about the usability of each approaches from the section 3.2. The very first decision criterion of all the BC based prototypes is the cost effectiveness, for which the solution uses private network which obviously need some tokens to log a transaction into the BC the tokens can be managed through the miner nodes by sending the tokens to the light nodes after checking the available account balance on regular basis. The reliability of the solution approach 2 (figure 3.8) is higher than other approaches as comparatively more amount of data coming form the sensor nodes are being logged to give broader analysis on the front-end. This enhances the possibility of getting correct value each time the data is sent to the BC. The reliability also depends on the communication duration, which is limited in other two approach because of TTN Fair Access Policy [112]. The cost effectiveness and reliability is added with high security for the next approach by using LoRa network for data transmission. In this case the number of data received by the gateway is very low, LoRa is good to give the complete security to the data coming from the sensors. Since the data is only transmitted whenever the receive window is available for transmission then the received data is transacted to the BC, which makes the approach less reliable compare to the first approach. Because the data is transacted from the gateway itself, this approach ensures the integrity and accuracy of the data made available in the public domain.

Comparing the two approaches involving Ethereum light client on the Sensor node to the Ethereum light client on the LoRa gateway, the high reliability is achieved in the first approach but the communication channel used is WiFi, which has shorter range and the battery usage is higher than the LoRa network. The whole sensor node setup uses +9V of battery power to start transmiting the correct data from all the sensors over LoRa network. The third approach (figure 3.9) uses Ethereum light node installed on the LoRa gateway provides a better solution. Since the model only executes couple of transactions because of the TTN Fair access policy [112], it is important to have more trustworthy data transmission for correct analysis at the user's end. To achieve the desired transparency,

the data is directly transacted to the BC from the LoRa gateway itself (Secures complete data transmission).

The prototype showed high degree of scalability, as the front-end is supporting the implementation of the solution all over the world. The publicly available website can be used by any of the labs in the world and register themselves on the web application. The number of the sensors available in the sensor node can be easily increased depending on the requirements by altering number of data fields in the present model without affecting the core logic of the current setup.

4.2 System Evaluation

4.2.1 Test Environment Setup and Verification

The system was prepared for the evaluation by following the steps enumerated below.

1. LoRa node was powered up through two 9V batteries added in series.
2. Lora gateway was powered up through stationary wall socket and the connection was verified over TTN console. Ethereum light client and web interface (main.js) were also started.
3. Ethereum full node is started into on the full capacity machine and paired with the light node already started in the LoRa gateway.
4. The blockchain peers were verified for creating a private network as seen in the figure 4.1.
5. the web server at the users terminal is started.

Once the system was running then the following tests were performed.

4.2.2 Sensor Integration Test

This was the first part of the complete testing of the system. In this phase the data received on the TTN console was verified with the data received at the gateway. The data can be seen through the output of the interface service (main.js, continuously running on this gateway). Each time the data is received by the gateway it was being sent to the BC, as a result of this process incoming data (from TTN) and outgoing data in the form of transactions could be seen in the output figure 4.2.

Figure 4.1: Added Peers

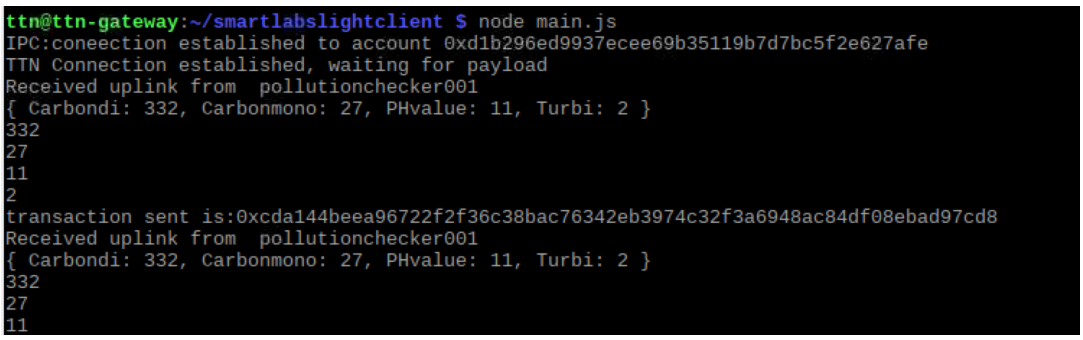

Figure 4.2: Data Received from TTN

4.2.3 BC Integration Test

As the data was being sent over the network, Unauthenticated users could (also) see the recent update through the alerts generated on the maps (RED/GREEN flags). The drop-down menu resulted into showing the current average data recorded by several installations done in the chosen city (for the prototype testing only one installation was used hence the result was not an average) as shown in figure 4.3 & 3.10.

4.2.4 BC Update Test

After Authentication, the Pollution standards form was filled and submitted, this updated the Smart Contract values for current Pollution Standards. This was verified through the read view provided on the right of the form in the dashboard page (figure 3.12). The update tab was able to update the current data of the BC, which was verified by completing the read form available in the tab (Update) (figure 3.14). The data received from the TTN

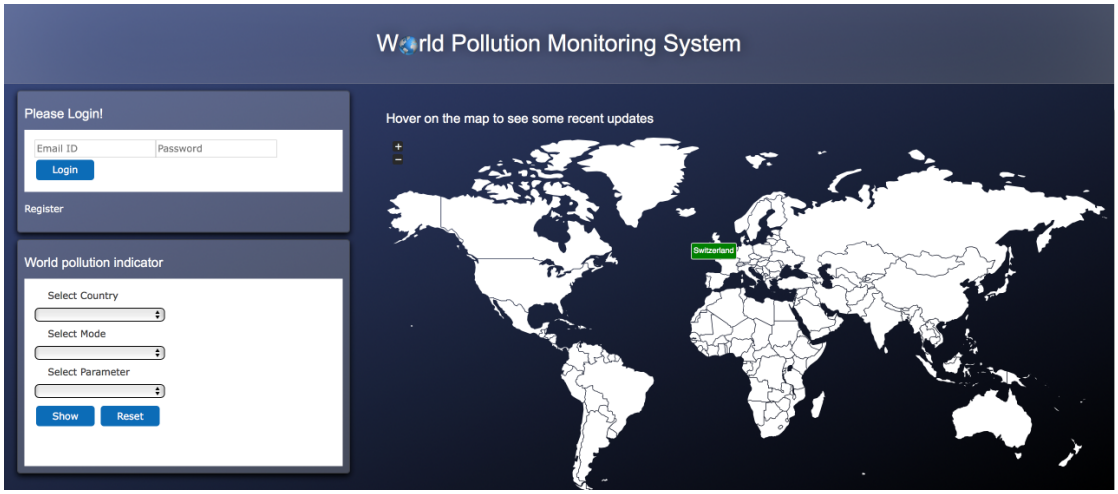

Figure 4.3: Alerts Generated on the Maps (RED/GREEN Flags)

network through the storage integration could also be seen on the dashboard. User needs to refresh the page for getting new data shown on the dashboard once it is received. The data received by the LoRa network or updated on the BC is always updated into the BC and this data is used to show the historical data about the specific pollutant over time. This statistics was found updated every time the data was received or updated on the BC. The data can be verified in the storage account on Swagger data storage integration page, as in the figure 4.4

4.3 Findings

In the first design proposal (With LoRa) (figure 3.7), since it is using LoRaWAN for the communication purpose it is highly secure but not reliable enough as it has an Air-Time limit and the data gathered could be insufficient for the formal conclusion in some cases. Where as the Communication protocol used in the second proposed design (figure 3.8) the communication is not highly secure as compared to LoRaWAN but it is giving continuous data communication both ways. In both the cases the security of data as discussed in earlier is after all taken care by the BC technology which is most renowned solution for the security and reliability as of now.

The other end i.e, Web Application is secured by the authentication gateway, only authorized members can only pass the gateway and update the BC. The local government's publicized standards (example shown in table 3.1) for the pollution index can also be updated from here. The public viewing of the data update can be done without authentication.

The System is scalable as it can be implemented in several cities in several countries and the data can be updated by adding the country, city and device name (PMS) in the system. The current system is made exclusively for Zurich, Switzerland, so it is hard coded in the system for the prototype purpose. If the system is deployed on larger scale and the pollution is monitored among different cities / countries from a central location

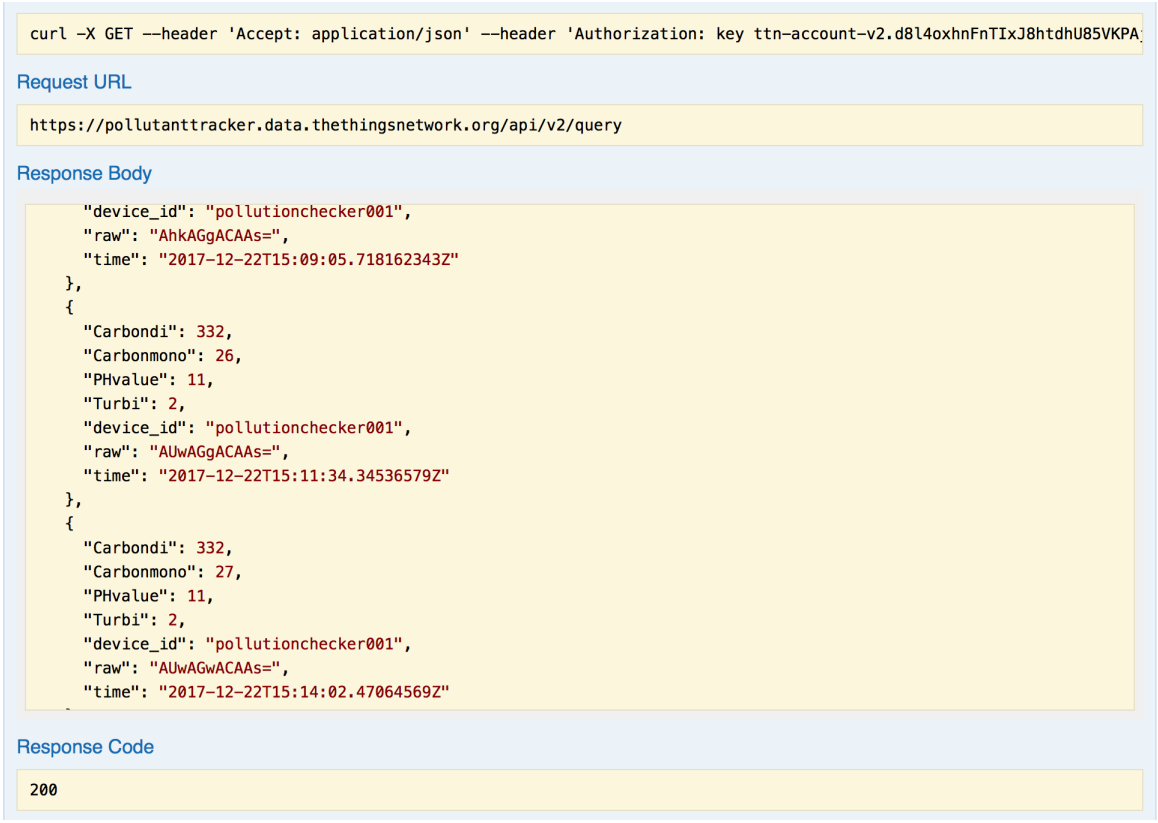

Figure 4.4: Data Gathered by Swagger Data Storage Integration Service of TTN

a GPS system embedded to sensor node can be beneficial for tracking the objects and getting the correct location data into the system automatically

The solution also has some drawbacks as each of the installation is required to use the same BC used in the rest of the solution, which makes the system dependent on a common service provider (Ethereum). The number of data that can be sent over the LoRa network is restricted with the air-time regulation of the TTN network (30 sec per day), this restricts the user from monitoring the systems behaviour over the longer period of time. The data (uplink and downlink) received can be seen in figure 4.5 & 4.6.

The restricted payload size (12 bytes) also restricts the number of pollution indicating sensors to be added to the LoRa node. The current solution has 4 sensors attached and only the pollution data is collected and sent over the network (payload size 8 bytes of real data + 4 bytes of LoRa headers =12 bytes in the provided solution). If one needs to have these many sensors, battery and location information to be transmitted over the network then using LoRa network is not a good enough, as the payload size would exceed the allowed payload size limit, thus the data would not be transmitted to the gateway.

With the number of data, the size of the Smart Contract to be deployed over the network also increases which makes the Smart Contract more cost heavy and each transaction too. This can act as a deal breaker for some organizations depending on their need and cost bearing capacity.

The connection with the light client in the private BC network is fragile and done manually.

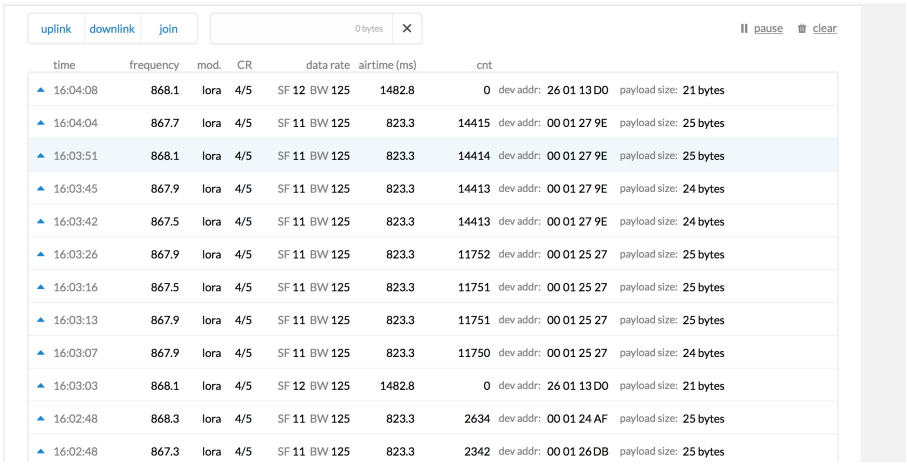

Figure 4.5: Uplink Traffic on TTN Gateway Console

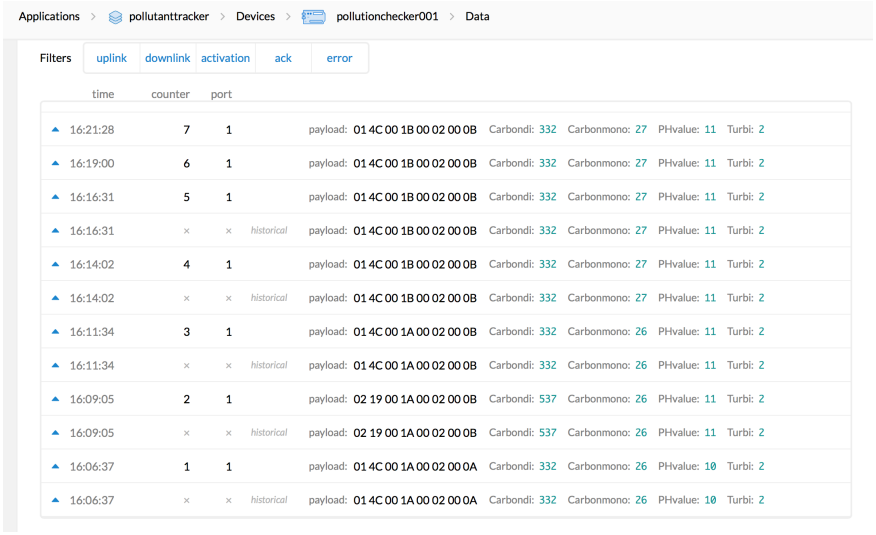

Figure 4.6: Downlink Packets Received on TTN Application Console

As the clients may loose their peers at each time they change the network, this manual work is necessary for mining the transactions sent by the light clients onto the common network.

Regarding the overall coverage of LoRa and TTN, it should be mentioned that the gateway installed for this solution is an indoor gateway and it needed to be placed near to a window for better connectivity from the outside world. The outdoor coverage of the setup was better than the indoor without many obstacles between LoRa node and gateway in indoor locations. Communications of the LoRa node and gateway was fast when they were placed directly in front of each other. So, it can be deduced that the node should be placed in an open area and the path to the gateway should not have many obstacles in between.

The light client of Ethereum needs transaction fees to send data to the BC, keeping eye on the token balance of the light node through the full node on regular basis is important for uninterrupted flow of data.

Regarding power consumption of the proposed PMS, the whole sensor node setup uses with 4 sensors, power generated by ≈ 18 V of battery (9 V batteries in series) to start up and transmit the data over LoRa network. Total power consumption of the sensor nodes needs to be divided to power consumption in transmitting data and sensing (gathering) data. Regarding the data transmission, LoRaWAN enables the communication to be power efficient as discussed in the Section **??**. Which narrows down the major power consuming parts of the PMS as Sensors and Computing board (Arduino Uno), the major part of power consumption directly relates to the number of sensors and their power consumption.

The time between when the data is received by the LoRa gateway from TTN network and updating actual BC depends on the miners availability. If the miners are available, the transaction is added to the BC as soon as it is received. This keeps the real time perception of the users intact, ensuring high level of user experience. The transactions submitted by the light client waits for the miners to get accepted, the queue of transactions to be added to the BC can be seen in figure 4.7.

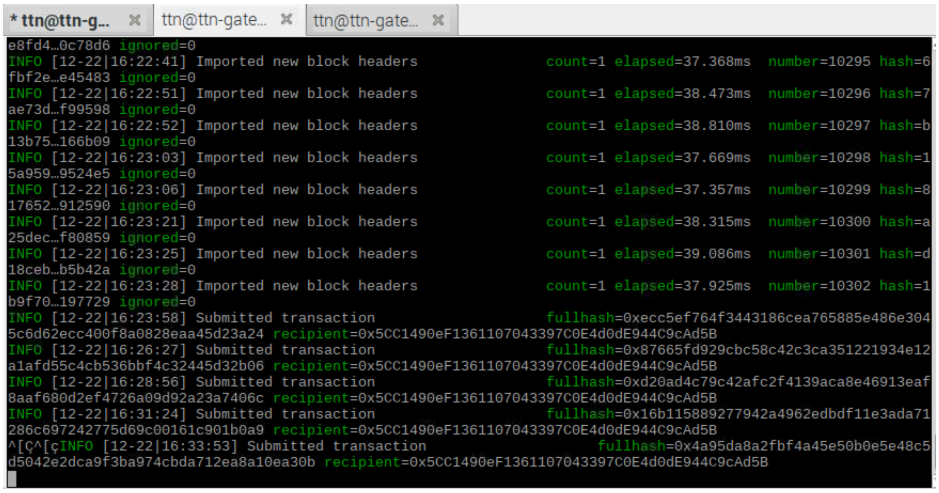

Figure 4.7: Submitted Transaction by Light Client Waiting for m Miners

The frame count at the TTN console increases depending on the frame count of past payloads received. This stops gateway from receiving the down-link packets with lower frame-counts. Resetting the frame count helps gateway to receive all the data coming from the sensor node.

Chapter 5

Summary and Conclusions

The major parts of this master thesis was to give a solution for the security and trust issues in the present (real world) pollution monitoring systems. The introduction section 1 gives details about the present issues and its importance. The solution provided (PMS) works with BC for maintaining the accuracy of the data from sensors to the end users. The data collected is used for analysis to make decisions about the environment and its behaviour, so the accuracy is very crucial in this case. Furthermore the thesis provides an overview of all the communication protocols which can be used and their comparisons to each other. As a result a solution design is proposed for tackling the issues. A detailed comparison among the presently available BCs resulted into the selection of BC (Ethereum) for storing data and get the highest accuracy possible through the distributed system. The the proposed solution was realized by creating a web application for pollution monitoring using smart contracts and observing its benefits over the system without BC involved. In the further sections the necessary protocols and technologies are discussed. The final design for the solution was declared by comparing different possibilities of placing BC between the sensor nodes and the web application. The three different models mentioned in section 3.2 were implemented separately during the implementation phase of the thesis and depending on the drawbacks of the models a new idea was proposed. The iterative way of designing resulted in a robust solution for providing high cost effective and security to the data travelling through the system. The evaluation phase showed the drawbacks and future expectations of the system discussed in section 4. The PMS web application is developed such that the users from different regions of the world can be benefited and unauthenticated users would be able to check the data for free. This platform is scalable as it can support multiple installations in different regions and multiple LoRa gateways can be added to the system to increase the data collection at end devices.

To conclude, it can be said that the choices made for the final system setup were accurate and the solution provided in this thesis works well in the real world scenarios. The current solution can be used for mare than one premises of environment monitoring with use of different sensors at the sensor nodes and small programming amendments. The system has many benefits with some drawbacks as well. For example the air time of TTN is 30 seconds and the transaction fee in Ethereum BC is high. The fair access policy of the LoRa does not allow for continuous down-link messages that withholds the users from accessing continuous real-time data on the monitoring system. However the benefits of the solution

overshadows the drawbacks. As the main work of the system to make the users aware of the changes in the environment is fulfilled, the limited time of data collection is adequate to detect the pollution standard violations in the environment. By implementation of the current system the staff members of the pollution control labs would be successful in keeping the record transparent and the awareness about the pollution control would increase. The evaluation tests ensures that using smart contracts with the LoRaWAN provides the high security and accuracy to the system.

Bibliography

[1] Nakamoto Satoshi. "Bitcoin: A peer-to-peer electronic cash system." 2008.

[2] Kosba, Ahmed, et al. "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts." Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 2016.

[3] Christidis, Konstantinos, and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things." IEEE Access 4 2016.

[4] Chen, Y. How Can Blockchain Technology Help Fight Air Pollution? https://media.consensys.net/how-can-blockchain-technology-help-fight-air-pollution-3bdcb1e1045f, Accessed 23 Sept. 2017.

[5] Kim, Moon-Kyung, and Kyung-Duk Zoh. "Occurrence and removals of micropollutants in water environment." Environmental Engineering Research, vol. 21, no. 4, 2016, pp. 319-332., doi:10.4491/eer.2016.115.

[6] Slooff, W., Dick de Zwart, and J. M. Marquenie. "Detection limits of a biological monitoring system for chemical water pollution based on mussel activity." Bulletin of Environmental Contamination and Toxicology 30.1 (1983): 400-405.

[7] "What Are the Applications and Use Cases of Blockchains?" CoinDesk, 15 Mar. 2017, www.coindesk.com/information/applications-use-cases-blockchains/. Accessed 23 Sept. 2017.

[8] "Bitcoin Fees for Transactions." Bitcoin Fees for Transactions, bitcoinfees.21.co/. Accessed 23 Sept. 2017.

[9] "Bitcoin Block Explorer - Blockchain." Bitcoin Block Explorer - Blockchain, blockchain.info/. Accessed 23 Sept. 2017.

[10] "Bitcoin Hash Functions Explained." CoinDesk, 19 May 2017, www.coindesk.com/bitcoin-hash-functions-explained/. Accessed 23 Sept. 2017.

[11] Buterin, Vitalik. "Ethereum white paper." (2013).

[12] "Block hashing algorithm." Block hashing algorithm - Bitcoin Wiki, https://en.bitcoin.it/wiki/Block-hashing-algorithm. Accessed 23 Sept. 2017.

[13] "RPOW - Reusable Proofs of Work2004." RPOW - Reusable Proofs of Work | Satoshi Nakamoto Institute, nakamotoinstitute.org/finney/rpow/. Accessed 23 Sept. 2017.

[14] "Bitcoin Fees for Transactions." Bitcoin Fees for Transactions, bitcoinfees.21.co/. Accessed 23 Sept. 2017.

[15] "Bitcoin White Paper: Beginner's Guide - Bitcoin.Com." Bitcoincom, 17 Sept. 2017, www.bitcoin.com/guides/bitcoin-white-paper-beginner-guide. Accessed 23 Sept. 2017.

[16] "Blockchain cross border payments." B2B Pay powered by Barclays, 3 Aug. 2017, www.b2bpay.co/using-blockchain-cross-border-payments. Accessed 23 Sept. 2017.

[17] "This Bitcoin Botnet is Vying to Be Future of Secure IoT." CoinDesk, 27 Mar. 2017, www.coindesk.com/this-bitcoin-botnet-is-vying-to-be-future-of-secure-iot/. Accessed 23 Sept. 2017.

[18] "Cryptocurrency Market Capitalizations | CoinMarketCap." Cryptocurrency Market Capitalizations | CoinMarketCap, https://coinmarketcap.com/. Accessed 23 Sept. 2017.

[19] "Whitepaper Â· IOTA." IOTA, https://iota.readme.io/docs/whitepaper. Accessed 23 Sept. 2017.

[20] "Introduction." Sandbox Documentation, https://dev.iota.org/sandbox/. Accessed 23 Sept. 2017.

[21] Ripple. "Ripple/Rippled." GitHub, 1 Sept. 2017, https://github.com/ripple/rippled. Accessed 23 Sept. 2017.

[22] Lisk. "What Is Lisk? And What It Isn't. - Lisk Blog." Lisk Blog, Lisk Blog, 26 July 2016, http://blog.lisk.io/what-is-lisk-and-what-it-isnt-e7b6b6188211. Accessed 23 Sept. 2017.

[23] Stratis-Whitepaper, https://stratisplatform.com/files/Stratis-Whitepaper.pdf. Accessed 23 Sept. 2017.

[24] "Semtech Environment Air pollution Application Brief." 2016, https://www.semtech.com/wireless-rf/internet-of-things/downloads/Semtech-Enviro-AirPollution-AppBrief-FINAL.pdf.

[25] Lora-Alliance, https://www.lora-alliance.org/technology.

[26] Zigbee Alliance, https://www.zigbee.org/.

[27] Al-Ali, A. R., et al. "A Mobile GPRS-Sensors Array for Air Pollution Monitoring." IEEE Sensors Journal, vol. 10, no. 10, 2010, pp. 1666-1671., doi:10.1109/jsen.2010.2045890.

[28] "Company." Semtech Supplies Analog and Mixed-Signal Semiconductor Products for Use in Computers, Portable Devices, Communications and Industrial Equipment., https://www.semtech.com/Press-Releases/2017/MI-Products-Leverage-Semtech's-LoRa-Technology-to-Help-Improve-Public-Safety.html.

[29] Dallas, Texas May 18 2017. "AT&TNewsroom." AT&T Launches LTE-M Network a Step Forward to 5G, 2017, https://about.att.com/story/att-launches-lte-m-network-a-step-forward-to-5g.html.

[30] "GPRS Network Architecture Tutorial." GPRS Network Architecture | GGSN SGSN PCU | Radio-Electroncis.com, https://www.radio-electronics.com/info/cellulartelecomms/gprs/gprs-network-architecture.php.

[31] "RF Wireless World." LoRa vs Zigbee | Difference between LoRa and Zigbee, https://www.rfwireless-world.com/Terminology/LoRa-vs-Zigbee.html.

[32] "RF Wireless World." LoRa Modulation Basics | CSS Modulation | Advantages,Properties, www.rfwireless-world.com/Terminology/LoRa-modulation-vs-CSS-modulation.html.

[33] Ray, Brian. "WiFi Vs. Cellular: Differences; Uses For M2M Applications." Link Labs: Low Power Wide Area Networks (LPWAN) and LTE-M for IoT, www.link-labs.com/blog/wifi-vs-cellular-differences-for-m2m.

[34] Wixted, Andrew J, et al. "Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks." 2016 Ieee Sensors, 2016, doi:10.1109/icsens.2016.7808712.

[35] "Libelium World." Libelium Connecting Sensors to the Cloud RSS, www.libelium.com/lorawan-waspmote-868-europe-900-915-us-433-mhz-asia-lora/.

[36] "10 Internet of Things (IoT) Design Considerations: Cost and Network." Grid Connect Blog, 2015, gridconnect.com/blog/general/10-internet-things-iot-design-considerations-cost-network/.

[37] "3G Vs 4G." 3G Vs 4G - Difference and Comparison | Diffen, www.diffen.com/difference/3G-vs-4G.

[38] "EMS31." Gemalto, www.gemalto.com/m2m/solutions/modules-terminals/industrial/ems31.

[39] "FMLR Sensors." FMLR - LoRaWAN Sensors - Miromico, www.miromico.ch/fmlr-lorawan-sensors.html.

[40] "How to Use the SIGFOX Technology to Connect to the Internet of Things." How to Use the SIGFOX to Connect to the IoT | DigiKey, www.digikey.com/en/articles/techzone/2016/apr/how-to-use-the-sigfox-technology-to-connect-to-the-internet-of-things.

[41] "IDIAG Humidity." Sigfox Partner Network, partners.sigfox.com/products/idiag-humidity.

[42] Lou Frenzel 1 | May 16, 2017. "Long-Range IoT on the Road to Success." Electronic Design, 2017, www.electronicdesign.com/embedded-revolution/long-range-iot-road-success.

[43] "Low Power, Wide Area." EEJournal, www.eejournal.com/article/20150907-lpwa/.

[44] Ray, Brian. "Examining The Future Of WiFi: 802.11ah HaLow, 802.11ad (Others)." Link Labs: Low Power Wide Area Networks (LPWAN) and LTE-M for IoT, www.link-labs.com/blog/future-of-wifi-802-11ah-802-11ad.

[45] Smith, Ms. "EZ-Wave: A Z-Wave Hacking Tool Capable of Breaking Bulbs, Abusing Z-Wave Devices." CSO Online, CSO, 2016, www.csoonline.com/article/3024217/security/ez-wave-z-wave-hacking-tool-capable-of-breaking-bulbs-and-abusing-z-wave-devices.html.

[46] "Top 7 Technologies for IoT Connectivity 2017." Flespi, flespi.com/blog/top-7-technologies-for-iot-connectivity-2017.

[47] "Make Things Come Alive in a Secure Way." Sigfox, www.sigfox.com/en/technology/security.

[48] "Sigfox's Ecosystem Delivers the World's First Ultra-Low Cost Modules to Fuel the Internet of Things Mass Market Deployment." Sigfox, www.sigfox.com/en/news/sigfoxs-ecosystem-delivers-worlds-first-ultra-low-cost-modules-fuel-internet-things-mass.

[49] Ben Dickson, T. (2017). How blockchain can change the future of IoT. [online] VentureBeat, https://venturebeat.com/2016/11/20/how-blockchain-can-change-the-future-of-iot/ [Accessed 13 Oct. 2017].

[50] Estevez, Francisco Jose, Peter Gloesekoetter, and Jesus Gonzalez. "DARAL: A Dynamic and Adaptive Routing Algorithm for Wireless Sensor Networks." Ed. Ignacio Bravo. Sensors (Basel, Switzerland) 16.7 (2016): 960. PMC. Web. 13 Oct. 2017.

[51] Barrett, B., Barrett, B., Pardes, A., Staff, W., Null, C., Pierce, D. and Staff, W. (2017). Wi-Fi That Charges Your Gadgets Is Closer Than You Think. [online] WIRED, https://www.wired.com/2015/06/power-over-wi-fi/ [Accessed 13 Oct. 2017].

[52] Gridconnect.com. (2017).Environmental and Security Sensors for AKCP sensorProbes & securityProbes | Grid Connect, https://gridconnect.com/sensors-io/environmental-sensors.html [Accessed 13 Oct. 2017].

[53] Gemalto.com. (2017). EMS31:Industry-first LTE Cat. M1 module for Highly Efficient LTE-M connectivity optimized for IoT, http://www.gemalto.com/m2m/solutions/modules-terminals/industrial/ems31 [Accessed 13 Oct. 2017].

[54] Sigfox.com. (2017). Sigfox Technology Overview | Sigfox, https://www.sigfox.com/en/sigfox-iot-technology-overview [Accessed 13 Oct. 2017].

[55] IoT Daily. (2017). Sigfox Pros and Cons, https://iot-daily.com/2015/03/13/sigfox-pros-and-cons/ [Accessed 13 Oct. 2017].

[56] Rfwireless-world.com. (2017). Advantages of SigFox | Disadvantages of SigFox, http://www.rfwireless-world.com/Tutorials/advantages-and-disadvantages-of-Sigfox-wireless-technology.html [Accessed 13 Oct. 2017].

[57] Anon, (2017), https://www.sigfox.com/en/coverage [Accessed 13 Oct. 2017].

[58] Rfwireless-world.com. (2017). LoRaWAN Classes | Class A,Class B,Class C | RF Wireless World, http://www.rfwireless-world.com/Tutorials/LoRaWAN-classes.html [Accessed 13 Oct. 2017].

[59] Kane International Ltd. (2017). What are safe levels of CO and CO2 in rooms? | Kane International Ltd, https://www.kane.co.uk/knowledge-centre/what-are-safe-levels-of-co-and-co2-in-rooms [Accessed 13 Oct. 2017].

[60] Hu, Yi, et al. "A highly sensitive in-situ turbidity sensor with low power consumption." Photonic Sensors 4.1 (2014): 77-85.

[61] Khanna, Neha. "Measuring environmental quality: an index of pollution." Ecological Economics 35.2 (2000): 191-202.

[62] Pavani, Movva, and P. Trinatha Rao. "Real time pollution monitoring using Wireless Sensor Networks." Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual. IEEE, 2016.

[63] Akyildiz, Ian F., et al. "Wireless sensor networks: a survey." Computer networks 38.4 (2002): 393-422.

[64] Pope III, C. Arden, et al. "Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution." Jama 287.9 (2002): 1132-1141.

[65] Kazmeyer, M. (2017). Cite a Website - Cite This For Me. Sciencing.com, https://sciencing.com/water-ph-amp-pollution-5850000.html [Accessed 14 Oct. 2017].

[66] Nemoto.eu. (2017). The NAP-505 Low cost Carbon monoxide sensor from Nemoto, http://www.nemoto.eu/nap-505.html [Accessed 14 Oct. 2017].

[67] Team, S. (2017). Grove - Gas Sensor(MQ2) - Seeed Wiki. [online] Wiki.seeed.cc, http://wiki.seeed.cc/Grove-Gas-Sensor-MQ2/ [Accessed 14 Oct. 2017].

[68] Wiki.eprolabs.com. (2017). Gas Sensor MQ7 - ePro Labs WiKi, https://wiki.eprolabs.com/index.php?title=Gas-Sensor-MQ7 [Accessed 14 Oct. 2017].

[69] Gassensing.co.uk. (2017). CozIR LP ambient CO2 sensor, https://www.gassensing.co.uk/products/ambient-air-sensors/cozir-lp-ambient-air-co2-sensor/ [Accessed 14 Oct. 2017].

[70] Microcontrollershop.com. (2017). Carbon Dioxide (CO2) Sensor, TGS4161, 350 to 10000, http://microcontrollershop.com/product-info.php?products-id=6782 [Accessed 14 Oct. 2017].

[71] Lin, Wen-Chi, et al. "Multifunctional Water Sensors for pH, ORP, and Conductivity Using Only Microfabricated Platinum Electrodes." Sensors 17.7 (2017): 1655.

[72] Dfrobot.com. (2017). PH meter(SKU: SEN0161) - DFRobot Electronic Product Wiki and Tutorial: Arduino and Robot Wiki-DFRobot.com, https://www.dfrobot.com/wiki/index.php/PH-meter(SKU:-SEN0161) [Accessed 14 Oct. 2017].

[73] Digikey.ch. (2017). SEN0189 DFRobot | Entwicklungsboards, -kits, Programmierer | DigiKey, https://www.digikey.ch/products/de?keywords=SEN0189 [Accessed 14 Oct. 2017].

[74] Alliance, L. (2015). A technical overview of LoRa and LoRaWAN. White paper, Nov.

[75] Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A. B., & Chen, S. (2016, April). The blockchain as a software connector. In Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on (pp. 182-191). IEEE.

[76] Steemit.com. (2017). Cite a Website - Cite This For Me, https://steemit.com/cryptocurrency/@cyberblock/top-9-market-cap-blockchains-ranked-in-order-by-transaction-speed-lets-see-where-steem-fits-in [Accessed 27 Oct. 2017].

[77] (2017). Bitcoin, Ethereum, Litecoin, Dash Avg. Transaction Fee chart. [online] Bitinfocharts.com. Available at: https://bitinfocharts.com/comparison/transactionfees-btc-eth-ltc-dash.html#3m [Accessed 27 Oct. 2017].

[78] Rfwireless-world.com. (2017). LoRaWAN Classes | Class A,Class B,Class C | RF Wireless World, http://www.rfwireless-world.com/Tutorials/LoRaWAN-classes.html [Accessed 27 Oct. 2017].

[79] Thethingsnetwork.org. (2017). Home - The Things Network Wiki, https://www.thethingsnetwork.org/wiki/Backend/Home [Accessed 13 Dec. 2017].

[80] Thethingsnetwork.org. (2017). Home - The Things Network Wiki, https://www.thethingsnetwork.org/wiki/Backend/Home [Accessed 13 Dec. 2017].

[81] Jvectormap.com. (2017). Download. [online] Available at: http://jvectormap.com/download/ [Accessed 23 Dec. 2017].

[82] Chartjs.org. (2017). Chart.js Â· GitBook. [online] Available at: http://www.chartjs.org/docs/latest/ [Accessed 23 Dec. 2017].

[83] GitHub. (2017). ethereum/go-ethereum, https://github.com/ethereum/go-ethereum/wiki/Setting-up-private-network-or-local-cluster [Accessed 23 Dec. 2017].

[84] Remix.ethereum.org. (2017). Remix - Solidity IDE, https://remix.ethereum.org/ [Accessed 23 Dec. 2017].

[85] Web3js.readthedocs.io. (2017). web3.js - Ethereum JavaScript API-web3.js 1.0.0 documentation, https://web3js.readthedocs.io/en/1.0/ [Accessed 23 Dec. 2017].

[86] Anon, (2017). [online] Available at: http://wiki.dragino.com/index.php?title=Use-Lora-Shield-2B-Arduino-set-up-a-Lora-Node [Accessed 23 Dec. 2017].

[87] GitHub. (2017). ttn-zh/ic880a-gateway. [online] Available at: https://github.com/ttn-zh/ic880a-gateway/wiki [Accessed 23 Dec. 2017].

[88] The Things Network. (2017). Storage Integration, https://www.thethingsnetwork.org/docs/applications/storage/ [Accessed 23 Dec. 2017].

[89] The Things Network. (2017). Quick Start, https://www.thethingsnetwork.org/docs/applications/nodejs/quick-start.html [Accessed 23 Dec. 2017].

[90] IMST GmbH. (2017). iC880A-SPI - LoRaWAN Concentrator 868 MHz, http://shop.imst.de/wireless-modules/lora-products/8/ic880a-spilorawan-concentrator-868-mhz [Accessed 14 Oct. 2017].

[91] SMA Antenna for iC880A-SPI, WSA01-iM880B and Lite Gateway, http://shop.imst.de/wireless-modules/accessories/19/smaantenna- for-ic880a-spi-wsa01-im880b-and-lite-gateway [Accessed 14 Oct. 2017].

[92] U.fl to SMA - Pigtail cable for iC880A-SPI, http://shop.imst.de/wireless-modules/accessories/20/u.fl-to-sma-pigtail-cable-foric880a-spi [Accessed 14 Oct. 2017].

[93] Raspberry Pi 3 Model B, https://www.pishop.ch/raspberry-pi-3 [Accessed 14 Oct. 2017].

[94] Ultra microSDHC 16 GB 10 / U1,SDSQUNC-016G-GN6IA, SanDisk, https://www.distrelec.ch/en/ultra-microsdhc-16-gb-10-u1-sandisk-sdsqunc-016ggn6ia/p/30033465?q= SanDisk+MicroSD+Card+%2816+GB%29&page=1&origPos=1&origPageSize=25&simi=86.93 [Accessed 14 Oct. 2017].

[95] Reddy, V., B., Y., G., M., D., . . . D. (2017, October 12). IMST iC880a LoRaWAN backplane (Kit) by Gnz on Tindie, https://www.tindie.com/products/gnz/imst-ic880a-lorawan-backplane-kit/,[Accessed 14 Oct. 2017].

[96] EDUP 802.11b/g/n 150Mbps Wireless USB Adapter. https://www.pi-shop.ch/edup-802-11b-g-n-150mbps-wireless-usb-adapter [Accessed 14 Oct. 2017].

[97] MQ-135 Sensor (Air Quality), http://www.play-zone.ch/en/mq-135-gas-sensor-allg-luftqualitat.html [Accessed 14 Oct. 2017].

[98] MQ-7 Sensor (Carbon Monoxide / CO), http://www.play-zone.ch/en/mq-7-gas-sensor-kohlenstoffmonoxid-co.html [Accessed 14 Oct. 2017].

[99] Mouser Electronics. (2017). SEN0161 DFRobot | Mouser, https://www.mouser.ch/ProductDetail/DFRobot/SEN0161/?qs= /ha2pyFaduhfEf0KXb0HTnAc8SBBS7ZOD%2bdgQDlVLS4= [Accessed 26 Dec. 2017].

[100] Mouser Electronics. (2017). SEN0189 DFRobot | Mouser, https://www.mouser.ch/ProductDetail/DFRobot/SEN0189/?qs= /ha2pyFaduj1KeQ3FcKM1Ly0omfLehZTrOlQOUnMhq0= [Accessed 26 Dec. 2017].

[101] Bread Board Clear - 8.2*5.3cm, https://shop.boxtec.ch/breadboard-clear-8253cm-p-40199.html

[102] Dragino LoRa Shield - 868MHz v1.4 - Arduino., https://www.bastelgarage.ch/index.php?route=product%2Fproduct&search=Dragino%2BLora%2BShield&product-id=306

[103] GmbH, Y, http://www.yampe.com/product/details.jsp?curren=1&la=3&type=detail&product-id=2800&product-type-id=38 &person-id=0&brand-id=34 [Accessed 14 Oct. 2017].

[104] Buy Primary battery 9 V 6LR61/9V, GP Batteries, 1604AUP-B10/6LF22/9V ULTRA PLUS, https://www.distrelec.ch/en/primarybattery-6lr61-9v-gp-batteries-1604aup-b10-6lf22-9v-ultraplus/ p/30023538?q= Primary%2Bbattery%2B9%2BV%2B6LR61%2F9V%2B&page=1&origPos= 1&origPageSize =25&simi=97.59 [Accessed 14 Oct. 2017].

[105] Buy Cable connection for 9V battery, OKW, A9160003, https://www.distrelec.ch/en/cable-connection-for-9v-battery-okwa9160003/p/15056836?mainId=30023538 [Accessed 14 Oct. 2017].

[106] Breadboard Jumper Wire m-m (100pcs), https://shop.boxtec.ch/breadboard-jumper-wire-100pcs-p-40615.html [Accessed 14 Oct. 2017].

[107] Water.ncsu.edu. (2017). Water Resource Characterization DSS - Turbidity, http://www.water.ncsu.edu/watershedss/info/turbid.html [Accessed 29 Dec. 2017].

[108] Mr. Brian Oram, P. (2017). Water Research Center - pH. Water-research.net, http://www.water-research.net/index.php/ph [Accessed 29 Dec. 2017].

[109] Carbon Monoxide Kills. (2017). Permissible levels of Carbon Monoxide - Carbon Monoxide Kills, http://www.carbonmonoxidekills.com/are-you-at-risk/carbon-monoxide-levels/ [Accessed 29 Dec. 2017].

[110] Engineeringtoolbox.com. (2017). Carbon Dioxide Concentration - Comfort Levels, https://www.engineeringtoolbox.com/co2-comfort-level-d-1024.html [Accessed 29 Dec. 2017].

[111] Kane International Ltd. (2017). What are safe levels of CO and CO2 in rooms? | Kane International Ltd, https://www.kane.co.uk/knowledge-centre/what-are-safe-levels-of-co-and-co2-in-rooms [Accessed 29 Dec. 2017].

[112] Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the limits of LoRaWAN. IEEE Communications Magazine, 55(9), 34-40.

[113] Matthijs Kooijman. "LoraWAN-in-C library, adapted to run under the Arduino environment." https://github.com/matthijskooijman/arduino-lmic. Accessed 23 Sept. 2017.

Abbreviations

AAA	Authentication, Authorization, and Accounting
ELC	Ethereum Light Client
AES	Advanced Encryption Standards
WPA	WiFi Protected Access
BC	Blockchain
IoT	Internet of Things
PoW	Proof of Work
TTN	The Things Network
LPWAN	Low Power Wide Area Network

Glossary

Internet of Things It is a network of things, capable of communicating with each other.

Blockchain It is a data stored in a chain fashion to be accessed and maintained by the users themselves.

Ethereum Ethereum, is one of the several companies who is working closely with the Blockchain technology.

LoRa Long Range, Low Power wireless platform.

LoRaWAN Medium for LoRa device communication.

Cryptocurrencies Tockens generated by the blockchains to regulate the Proof of Work.

Uplink Sending data to the Node from your Application.

Downlink Sending response from your Application to the Node.

Geth Geth is the the command line interface for running a full ethereum node implemented in Go.

List of Figures

Listings

List of Tables

Appendix A

Installation Guidelines

Installation can be done by following the steps on Mac environment mentioned below:

1. Install Geth (https://geth.ethereum.org/install/) and Meteor (https://www.meteor.com).
2. Clone the git hub repository (https://github.com/jhasanjiv5/smartlabs.git)
3. Change the directories according to the downloaded repository location and run meteor (meteor)
4. After installation of Geth create an account by using phrase ”pollution monitoring system” in terminal 1, run (geth –port 30303 –networkid 6666 –datadir=”data” –rpc –rpcport 8545 –rpcaddr your local ip address –rpccorsdomain ”*” –unlock 0 –password=”password.sec” –rpcapi ”admin,eth,net,web3,personal,miner” –lightserv 70)
5. Open a new terminal 2 with same directory location run (geth attach ”http://your IP address:8545”)
6. Run (miner.start()) in the terminal 2, One can see the synchronization started once miner is started
7. Create the contract (copy the code from the repository) via connected Remix portal over the IPC
8. Once the web application is up and running tun on the sensor node and LoRa Gateway
9. Open the LoRa Gateway using VNC listener (password: raspberry)
10. Run Geth (geth –light –port 30303 –networkid 6666 –datadir=”data” –rpc –rpcport 8545 –rpcaddr your gateway IP address –rpccorsdomain ”*” –unlock 0

    –password=”password.sec” –rpcapi ”admin,eth,net,web3,personal”)

11. Run TTN-BC interface (node main.js, for updated version of the file use the git hub repository

    (https://github.com/jhasanjiv5/smartlabslightclient.git))

12. Check for added peers (admin.peers) if not found then connect the peers on Full node and Light node (admin.addPeer())

13. Once the peers are connected data synchronization started can be seen on the light client

14. Console in TTN gives detailed information about incoming traffic

    (https://console.thethingsnetwork.org)

15. Login to the Web application to update the data and standards using simple forms added on the web pages. The incoming data updates can be seen without authentication

16. Once the system setup is done, verify the working of the system by following the steps mentioned in section 4.2.1

Appendix B

Contents of the CD

The enclosed Compact Disk (CD) contains following contents:

1. Latex and PDF files of the Thesis documentation
2. Zipped file containing all the digital files (codes) required to replicate the system on any local machine (computer)