University of Zurich[UZH]

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

Improving Trust and Transparency Through Usability and User Experience in Decentralized Applications

Paulin Roth
Zurich, Switzerland
Student ID: 19-916-212

Supervisor: Daria Schumm, Thomas Grübl, Prof. Dr. Burkhard Stiller
Date of Submission: May 1, 2025

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

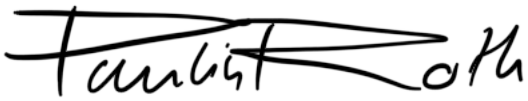
ifi

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

Declaration of Independence

I hereby declare that I have composed this work independently and without the use of any aids other than those explicitly declared in the Acknowledgments section, (including generative AI such as ChatGPT). Generative AI tools such as Writefull (Premium) for grammar improvement, the JetBrains AI Assistant for code restructuring and optimization, Claude 3.7 Sonnet for generating mock credential data, and ChatGPT-4o for creating landing page illustrations were used. All AI-generated content was critically reviewed, refined, and fully integrated by myself to ensure its quality, relevance, and originality. I am aware that I take full responsibility for the scientific character of the submitted text myself. All passages taken verbatim or in sense from published or unpublished writings are identified as such. The work has not yet been submitted in the same or similar form or in excerpts as part of another examination.

Zürich,



Signature of student

Abstract

In der heutigen digitalen Welt gewinnen Self-Sovereign Identity (SSI) und Decentralized Identity (DI) zunehmend an Bedeutung, da sie den Nutzenden mehr Kontrolle über ihre persönlichen Daten ermöglichen und gleichzeitig Datenschutz und Sicherheit stärken. Trotz ihrer Vorteile sind diese Systeme bislang kaum massentauglich, da sie häufig mit komplexen Konzepten, unklaren Anwendungsfällen, mangelnder Transparenz sowie schwacher Benutzerführung verbunden sind. Vorhandene DI/SSI-Anwendungen weisen oft gravierende Usability- und UX-Mängel auf, was deren Verbreitung erheblich hemmt.

Während die Forschung bereits einzelne Probleme identifiziert hat, fehlt bislang ein umfassender, praxisnaher Ansatz. Diese Arbeit schließt diese Lücke, indem sie einen Katalog von Richtlinien und konkreten UX-Elemente entwickelt, die speziell auf SSI-Applikationen ausgerichtet sind. Im Zentrum steht der Prototyp Mask Identity, der diese Merkmale umsetzt und durch Nutzerstudien validiert wurde. Anders als frühere Arbeiten bildet der Prototyp alle drei Rollen des Triangle of Trust (Holder, Issuer, Verifier) ab und deckt einen grossteil des Credential-Lifecycle ab, vom Antrag bis zur Verifikation und Widerrufung. Darüber hinaus liefert er konkrete, wiederverwendbare UI-Lösungen, wie visuelle Vertrauenssymbole, transparente Hinweise und intuitive Navigation.

Zur Umsetzung wurden bestehende Forschungsergebnisse zu Usability, UX und Vertrauen ausgewertet, daraus ein Feature-Katalog extrahiert und auf drei bestehende SSI-Anwendungen angewendet. Die identifizierten Schwächen bildeten die Grundlage für die Gestaltung eines neuen Prototyps namens „Mask Identity“. Anschließend wurde dieser Prototyp in einer Nutzerstudie mit qualitativen und quantitativen Methoden mit bestehende Lösungen verglichen.

Die Ergebnisse zeigen, dass der Prototyp in allen Bereichen, insbesondere bei Vertrauen und Transparenz, klar überlegen ist. Dies belegt, dass nutzerzentriertes Design ein entscheidender Hebel für die Akzeptanz von SSI-Systemen sein kann.

Zukünftige Arbeiten sollten unter anderem alternative Backup-Methoden, eine echte dezentrale Backend-Integration, sowie effektivere und interaktive Möglichkeiten zur Nutzeraufklärung untersuchen, etwa durch kontextbezogene Hinweise, Tooltips oder wiederholbare Tutorials. Auch die verbesserte Visualisierung der Vertrauenswürdigkeit von Issuern ist ein vielversprechender Ansatz. Langzeitstudien mit einer vielfältigeren Nutzergruppe könnten zudem helfen, das Verständnis von Vertrauen und Nutzung über längere Zeiträume hinweg zu vertiefen.

In today's digital world, Self-Sovereign Identity (SSI) and Decentralized Identity (DI) are gaining increasing importance, as they give users greater control over their personal data while enhancing privacy and security. Despite these advantages, such systems lack mass adoption, often due to their conceptual complexity, unclear use cases, lack of transparency, and poor user guidance. Existing DI/SSI applications frequently suffer from major usability and UX issues, which significantly hinder their acceptance.

While previous research has identified individual challenges, a comprehensive and practical approach is still missing. This thesis addresses this gap by developing a catalog of guidelines and concrete UX elements specifically tailored to SSI applications. At the core of this work is the prototype Mask Identity, which implements these features and validates them through a user studie. In contrast to earlier projects, the prototype supports all three roles of the Triangle of Trust (holder, issuer, verifier) and covers a large part of the credential lifecycle, from request to verification and revocation. Furthermore, it provides reusable UI patterns, such as visual trust indicators, transparent data handling cues, and intuitive navigation.

To develop these elements, existing research on usability, UX, and trust was analyzed, a feature catalog was extracted, and applied to three existing SSI applications. The identified weaknesses influenced the design of Mask Identity, which was then compared to existing solutions through a user study using both qualitative and quantitative methods.

The results show that the prototype outperforms the evaluated applications in all key areas, particularly in terms of trust and transparency. This demonstrates that a design focused on user needs can greatly enhance the uptake of SSI systems.

Future work should explore alternative backup methods, real decentralized backend integration, and more effective, interactive user education strategies, such as contextual tooltips or repeatable onboarding tutorials. Improving how issuer credibility and credential provenance are visualized also holds promise. In addition, long-term studies with a broader and more diverse user group could deepen our understanding of trust and user behavior over time.

Acknowledgments

Acknowledgments

First and foremost, I would like to thank my supervisors Daria Schumm and Prof. Dr. Burkhard Stiller for their guidance, feedback, and continuous support throughout the development of this thesis. Their insights and suggestions have been very valuable in helping me go into the right direction and for the quality of this work in general.

I am also thankful to the participants of the user study. They took the time to help test the application and gave honest and helpful feedback. Their responses helped me validate many of the assumptions and design choices I made during the project.

Use of Generative AI Tools

Several generative AI tools were used responsibly to support the development of this thesis, and all uses are transparently declared here:

- Writefull (Premium) was used to improve the grammar and clarity of the writing of this thesis.
- The JetBrains AI Assistant supported the restructuring and optimization of code. The models powering these improvements included Claude 3.7 Sonnet and ChatGPT o3 mini.
- The mock credential data used in the prototype (Mask Identity) was created with the help of Claude 3.7 Sonnet by Anthropic.
- The illustrations on the landing pages were generated using ChatGPT-4o.

All AI-generated content was critically reviewed and refined by the author to ensure quality, relevance, and originality.

Finally, I am grateful to the Communication Systems Group (CSG) at the University of Zurich for providing the environment and resources necessary to complete this work.

Contents

Chapter 1

Introduction

1.1 Motivation

Self-Sovereign Identity (SSI) and Decentralized Identity (DI) are emerging approaches to enhancing data privacy, especially in the digital domain. The core idea is to store data on decentralized systems like blockchains, in contrast to centralized servers, to enable users to maintain control over their personal data. For instance, when a third party needs to verify certain credentials, such as proof of age, possession of a drivers license, university degree or citizenship, they can do so without accessing more information than the user agrees to expose. When credentials are issued, they are signed with a digital signature by a trusted authority (e.g. a government) and verified by a verifier with the use a public key. This process ensures authenticity and prevents forgery. With the rise of artificial intelligence and its growing ability to mimic and forge identities, these concepts become increasingly important. Another aspect to consider is that many individuals living in developing countries, especially the elderly, lack official identity documentation. This significantly restricts their opportunities to participate in society, e.g. hindering them from voting, getting access to education, healthcare, or bank loans [49]. These populations could benefit from a SSI/DI solution. In addition, in countries with corrupt or inefficient governments, a centralized identity system might not be desired. It is also worth mentioning that empowering a hand full of big tech corporations by granting them control over your data (e.g. sign in to services using Google, Meta or Apple) might impose some risks on society.

Despite those and other advantages, mass adoption of Decentralized Identity technologies has not yet occurred and likely continues to struggle. While the European Union or the Swiss government are working on such solutions, global implementation could take many years. A key barrier is the general complexity and immaturity of the technology, which requires significant technical understanding and coordination between organizations. Additional challenges include the absence of widely accepted standards, interoperability concerns, and the lack of intuitive user interfaces [25].

This thesis seeks to address these challenges and bring SSI/DI technology closer to the user. By focusing on Usability and User Experience (UX) principles, the goal is to make

the technologies more accessible and trustworthy. The User Interface (UI) serves as the bridge between the user and the underlying technology, and its design must combine functionality with aesthetic appeal. Without those qualities, the users are not likely to adopt or continue to use the solution [26].

1.2 Thesis Goals

The thesis aims to create usability and UX guidelines to design trustworthy applications that enhance user engagement with SSI and DI technologies. It also identifies specific features that can be implemented to foster trust and improve usability. To validate these guidelines and features, a functional prototype will be created to show their application in a practical context.

1.3 Methodology

The study begins by outlining existing research on usability and UX principles, as well as trust in blockchain-based applications and other applications in general. From these foundations, guidelines and concrete best-practice usability features will be extracted. They will then be used to evaluate three existing SSI/DI solutions and to create a problem statement to identify usability and UX issues. As a next step, a prototype for an SSI application will be designed and implemented which incorporates the proposed guidelines and usability features and aims to be a good example of a trustworthy application. Following its development, the prototype will be evaluated by users through usability tests and compared with the above-mentioned applications. These evaluations will assess the effectiveness of the proposed guidelines and features.

1.4 Thesis Outline

The rest of this thesis guides the reader from the basic ideas of decentralized identity to the design, development, and testing of the created prototype. Chapter 2 (Fundamentals) explains the background by defining important terms such as usability, user experience, and trust, and by introducing technical terms like decentralized identifiers and verifiable credentials. It reviews previous research to extract general guidelines and concrete usability and UX features, which also serve as evaluation criteria. Based on these criteria, the next chapter analyzes three existing SSI applications and develops a problem statement that defines the main challenges and forms the basis for the prototype design. The problem statement also outlines the specific contributions of this thesis in more detail. Chapter 3 (Design) builds on these ideas to create a user-focused prototype called Mask Identity. It describes the choices of visual design and shows the main screens for holders, issuers, and verifiers, while referencing the design features where they were applied. Chapter 4 (Implementation) shows how the design was translated into a working product. It

presents selected code snippets to provide a conceptual impression of the implementation, rather than a complete technical walkthrough. Further technical details and the most important system components can be found in the appendix. A Chapter 5 (Evaluation) presents the results of a user study that compares Mask Identity with three existing SSI applications. It combines quantitative data (SUS, trust, transparency and UX scores) with qualitative observations to see where the prototype works well and where it can be improved. Finally, Chapter 6 (Final Considerations) sums up the main results, discusses the limitations, and suggests ideas for future research and development. The appendices include the system documentation, the pre-questionnaire, the user study questionnaire and the task descriptions for the study participants.

Chapter 2

Fundamentals

2.1 Background

2.1.1 Fundamentals of Usability, UX and Trust

Usability and UX significantly influence how effectively users can interact with a system. Those concepts are particularly important for digital applications, such as mobile applications and websites. While usability concerns the functional aspects of the interaction, UX focuses on the overall emotional experience a user has while engaging with the system. According to the International Organization for Standardization, usability is defined as the extent to which a system, product, or service can be used by specified users to achieve specific goals with effectiveness, efficiency, and satisfaction in a specified context of use [21]. Usability is the ease of use and acceptability of a product, shaped by the attitudes of users toward the system and their ability to perform tasks successfully [6]. From a user point of view, the ultimate goal of usability is to ensure acceptable levels of effectiveness, efficiency, and satisfaction [5, 21]. UX, on the other hand, is defined as a person's perception and response resulting from the use and/or anticipated use of a product, system, or service [22]. UX can be viewed as an extension of usability, including learnability, accessibility, and safety, which contributes to the overall user experience [4]. Studies [4] further suggested that UX can be measured as user satisfaction and pleasure while achieving a pragmatic or hedonic goal. A study [26] about mobile applications shows that poor user experience is a critical factor in abandonment of applications. Users expect smooth navigation, intuitive design, and minimal frustration caused by difficult interfaces or intrusive advertising. Applications with poor design and confusing interactions often frustrate users, leading them to uninstall the application [26].

Another major factor leading to application abandonment is the lack of trust. The study [26] further shows that users are more likely to delete an application they perceive as misleading, insecure, or invasive. Issues such as too many permission requests, system instability, exposing too much personal data, or application tracking undermine trust. Ensuring security, stability, and transparency is essential to enable trust and foster user loyalty [26]. Trust is often defined as the "willingness to act under uncertainty and with

the risk of negative consequences" [15]. In the context of applications, trust consists of multiple dimensions such as honesty, competence, integrity, and benevolence [9, 15, 37]. The user must believe that the other party is willing and capable to fulfill their promises and that they are reliable, honest, and do not have self-interest [9].

Usability and user experience are connected to trust. Better usability improves the user's ability to understand and navigate the systems efficiently which reduces errors and leads to a greater perception of competence and reliability in the service provider [15]. In addition, a well-designed application might include data protection and transparency design patterns that lead the user to perceive the other party as honest and trustworthy [37]. This highlights the important role of usability and user experience in fostering and maintaining trust in digital systems.

2.1.2 The Rise of Decentralized Digital Identity Systems

Traditionally, identity verification has been carried out with physical credentials such as passports and birth or marriage certificates. These documents allow individuals to prove their identity or specific attributes of themselves. Individuals can decide where to store the documents and to some extent with whom they share them [38]. However, with the rise of computer technology, centralized identity systems emerged as a way to manage and verify identities more efficiently. These systems rely on large databases controlled by governments or corporations and often force users to give up control over their personal data. The centralized approach raises ethical questions, particularly in terms of privacy and security. Centralized databases store large amounts of (sensitive) personal information in a single location, making them a target for cyberattacks. In addition, these systems often require more information than is necessary for verification purposes [38]. Inspired by physical credential systems, a new type of digital credential, the verifiable credential (VC), has emerged. VCs are cryptographically secure credentials that can be stored in a digital wallet on a smartphone, a desktop, or in the cloud. They can be used for identification, authentication, and authorization in many different contexts, which is a more secure and privacy-oriented alternative to traditional centralized systems [3, 38].

Decentralized identity systems offer an alternative to centralized data management, shifting power from institutions to individuals. However, in such systems, certain centralization elements remain, such as the initial proofing process, where trusted authorities verify the user's identity before issuing credentials. The achievement of complete decentralization is difficult to implement in practice [49].

Self-sovereign identity is built on top of decentralized identity and introduces features such as selective disclosure, which allows individuals to share only the information that is necessary. For example, a person who wants to prove their age could share the confirmation that they are over 18 without announcing their birthday, or other information that can be found on their national ID card. This approach enhances privacy and efficiency [18].

A practical example where the DI / SSI architecture might has been suited was the COVID-19 pandemic. Centralized databases stored large amounts of vaccination data

that help facilitate the verification process. However, this raised ethical questions, since it poses risk of profiling and vulnerabilities to hacking. The European Union introduced digital COVID certificates that allowed Member States to issue credentials directly to citizens' digital wallets. This gave users greater control over their data, but the lack of interoperability and standardization across digital wallets limited their utility for broader identity management application [38].

2.1.3 Triangle of Trust

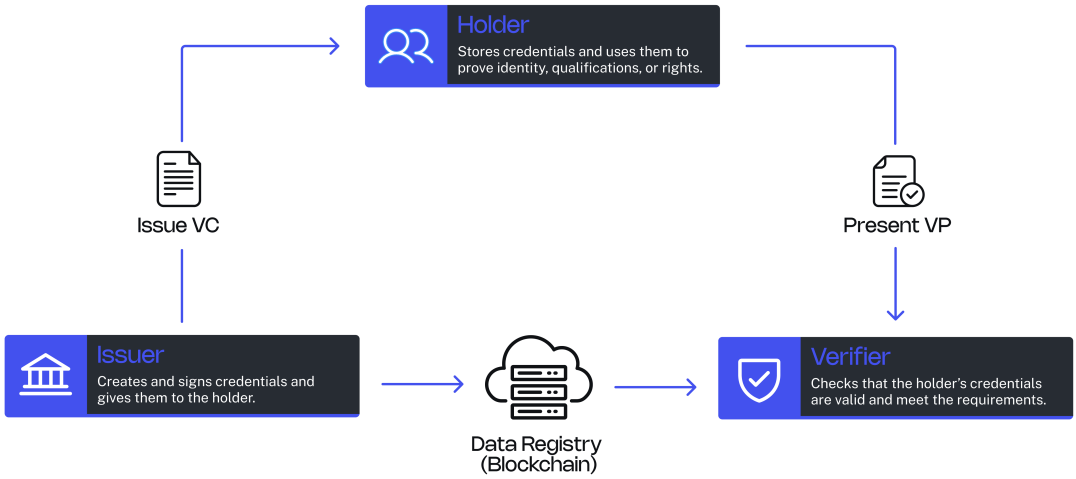


Figure 2.1: The Triangle of Trust, adapted from the illustration presented in the paper *Reimagining Digital ID* by the World Economic Forum [49].

A core mechanism in Decentralized Identity systems is the Triangle of Trust, which is illustrated in Figure 2.1. It involves three parties: The Issuer, the Holder, and the Verifier. This model is based on cryptography and allows for secure and privacy-preserving identity verification without reliance on a centralized database [49]. The process begins with a user (Holder) who wants to obtain a credential. To do so, the user must first create a decentralized identifier (DID). A DID works like a unique digital name that only the user controls. Unlike traditional usernames or IDs, it does not depend on a central authority like a government or a tech company. Instead, it is built using cryptographic technology so that it can be trusted and used securely on different apps and services, while protecting user's privacy [38]. The user's DID is stored in a public ledger (e.g., blockchain) along with a public key, allowing others to verify their identity through a cryptographic proof. To obtain the credential, the users make a request to an issuer, such as a university, a government agency, or an employer. The request contained the user's DID and is digitally signed with the users' private key, which proves their ownership of the DID. The Issuer then has to decide whether the user is eligible for the credential. Either the issuer already has a relationship with the user (e.g., the university knows that the student should receive a diploma), or the user needs to add additional documentation, such as government-issued

documents [18]. Once decided, the issuer digitally signed the credential using the private key and issues it to the users' digital wallet. This signature ensures that the authenticity of the credentials can be independently verified later without contacting the issuer. The credential includes information such as the issuer's DID, the holder's DID, and the specific claim (e.g., diploma, age, citizenship). Because it is digitally signed, the credential cannot be altered without breaking the signature, making it tamper-resistant and verifiable. When users want to access a service that requires proof of certain attributes, they must present a verifiable credential to the Verifier. Instead of sharing the entire credential, the Holder can use Selective Disclosure to reveal only the necessary information while keeping other details private [38]. The Holder creates a Verifiable Presentation (VP), which is a cryptographically signed package containing only the selected parts of the original credential. This VP proves that the Holder possesses a valid credential issued by a trusted party, without revealing unnecessary personal information. The VP is then sent to the Verifier, who evaluates its validity by checking several key aspects. First, it confirms the authenticity of the Issuer by verifying the Issuer's DID against records stored on a public ledger. This ensures that the credential originates from a trusted authority. Second, the Verifier checks the integrity of the credential by validating its cryptographic signature, ensuring that the data has not been altered. Finally, it queries the revocation registry to confirm that the credential is still valid and has not been revoked. This process allows for secure, privacy-preserving verification [49].

Only credential metadata and non-sensitive information, such as DIDs, public keys and revocation lists, are stored on the public ledger. The user's personal data remains securely stored on their device and is never published to the public ledger. This setup allows the Verifier to validate credentials independently, without needing to contact the Issuer, thereby preserving trust, security, and decentralization in the identity verification process [49].

2.2 Related Work

2.2.1 Usability and UX Guidelines for DI and SSI Systems

The design and user experience of DI and SSI systems are crucial to their adoption. These systems offer enhanced security and user control over personal data, but introduce new challenges due to their complexity and novel interaction paradigms. Many users are not familiar with the underlying technology. Research has shown that the steep learning curve and the need for users to manage things such as cryptographic keys can hinder adoption [20]. For example, research indicates that the technical complexities associated with blockchain technology create challenges for both users and developers, potentially slowing widespread adoption [20]. Furthermore, a study [36] that focused on the user experience of SSI wallets found that usability often plays a subordinate role in development efforts, although it is crucial for user satisfaction and adoption.

The first part of this section focuses on design patterns and usability issues specific to DI, SSI and digital identity applications. These insights are derived from dedicated research on these systems and are directly relevant to the goals and scope of this thesis.

However, the principles of good UX are not exclusive to Decentralized Identity. General UX research offers valuable insights for creating good design of DI/SSI systems, where examples are drawn from websites and applications in general. This includes patterns such as minimalist design, social proof, and cognitive load reduction, which is particularly relevant given the complexity of decentralized interactions. Many of these principles are broadly applicable across different domains. For the purpose of this thesis, they have been critically assessed and adapted to fit the specific context of DI and SSI systems.

2.2.2 Guidelines Derived from DI and SSI Application Research

The following paragraphs are based primarily on research on DI and SSI as well as the broader field of Digital Identity.

Simple Terminology One of the main obstacles to creating a good user experience is the extensive use of technical jargon that is unfamiliar to nontechnical users. Terms such as 'Decentralized Identifier', 'Verifiable Credentials' and 'Seed Phrase' are often unclear to users, leading to confusion and potential errors [40]. Simple terminology on buttons, labels, and system messages establishes accessibility and fosters trust in the application [24]. Furthermore, providing clear, nontechnical error messages and explanation on certain functions such as 'backup' or 'recovery' reduces frustration and cognitive load [28].

Aligning with the Users Mental Model A mental model is the internal understanding users form about how a system works, shaped by their past experiences and expectations. In application design, aligning interfaces with these mental models is crucial for usability. When there's a mismatch, users may become confused or make errors. Mental models are defined as "what the user believes about the system at hand," emphasizing that discrepancies between user expectations and actual system behavior can lead to frustration and abandonment of the application [19, 45, 48]. Users typically interact with DI and SSI systems based on existing mental models, formed from traditional authentication methods such as username-password logins. Many existing decentralized applications fail to align with those mental models and focus too much on representing the underlying technology. This results in usability issues such as difficulties in obtaining credentials and understanding how data is stored [24]. In a usability evaluation [24] the application SmartWallet, uPort, and Connect.Me performed poorly, with SmartWallet receiving a negative score for understandability. Users struggled to obtain credentials and backup the recovery phrase. Many users did not store the recovery phrase because they mistakenly assumed that their data was stored centrally and could be recovered without it. Furthermore, users were often confused about where their data were stored, leading them to believe that they could delete it from a central server, even though it was stored locally [24].

An approach to improving usability is to incorporate familiar metaphors that help users conceptualize SSI intuitively [30]. Research has shown that experts commonly describe SSI using metaphors, in particular, they make the analogy with the digital wallet and physical documents [30]. The digital wallet is already widely adopted in the context of

cryptocurrencies. It allows users to perceive credentials as items they carry and control, reinforcing the idea of self-sovereignty. Similarly, the physical document metaphor can highlight the importance and uniqueness of a specific credential, just like physical identity documents such as passports or driver's licenses. These metaphors help to convey self-responsibility because losing a physical documents requires re-issuance and losing a digital identity requires a recovery process [30].

The application should incorporate intuitive navigation, so that the user quickly finds the task and can complete it in a few steps [28]. The structure of the wallet should incorporate user-centered workflows rather than purely reflecting the decentralized architecture [40].

User Education and Onboarding Given that SSI systems are new to most users, educating them on key concepts is crucial. Users unfamiliar with credential authentication often struggle to complete the necessary interactions, such as scanning a QR code [40]. A brief onboarding tutorial in the form of a virtual tour or an introductory video in the beginning helps users understand the core features and possible interactions. They should be optional, but available at anytime [40]. Furthermore, contextual help, such as 'info' symbols or brief texts next to important features are beneficial to user education. In addition, FAQs and in-application assistance should be available at all times [40].

On the other hand, it is important not to overwhelm the users with instructions [28]. Instead of requiring users to learn complex concepts, SSI applications should hide technical complexity wherever possible. A study [30] suggests that users do not need to understand the underlying architecture of the system to use it effectively. Rather, a well-designed application should abstract these complexities and focus on a good and seamless user experience. SSI applications should be simple to use, just like Google or Facebook Single Sign-In (SSO) [30].

However, users need to get enough information to trust the system. For example, security symbols can be used to indicate that something is encrypted, without explaining in detail how encryption works [30].

With the right balance between education and simplicity, SSI applications can ensure that users feel confident in managing their identity without being overwhelmed by technical details. An essential part of user education should be on data storage and trust, which are key aspects of the minimal knowledge map for end users, proposed in the cited study [30]. Users must understand that they are responsible for storing their credentials securely in a wallet. In addition, SSI systems do not rely on a single centralized company but on open-source cryptographic protocols. Users should be aware that only the necessary information is shared with verifiers and not the full data. Educating users about these principles helps build confidence in SSI systems[30].

Trust through Transparency Transparency is a key factor in building trust in DI and SSI applications. The user needs clear information on where and how their data is stored, weather locally or cloud-based [28]. Otherwise, they would fear their data are still recoverable after deletion of the credentials [40]. The application should provide clear statements about data management [24].

Backup and Recovery Mechanism One of the main aspects of the literature on trust in SSI software is account recovery. Many wallets rely solely on mnemonic phrases (seed phrases) for recovery. This method is unfamiliar or even confusing to many users [30, 52]. More convenient methods include selecting the seed phrase from a pool of words or sharing the recovery key among multiple people or items [30]. Flexible storage options (e.g., cloud or local file) and clear explanations of their advantages and risks are essential for reducing user anxiety [24]. Solutions should also implement frequent backup reminders and provide intuitive recovery methods that align with user expectations [40].

2.2.3 Guidelines from DI/SSI Research and General UX Principles

The following paragraphs bring together insights from both general UX research and specific studies on DI and SSI systems. This includes principles from web and application design. These ideas have been carefully reviewed and adapted where relevant, focusing on topics such as visual clarity, ease of use, cognitive load, and trust, common problem areas in DI and SSI applications.

Social Proof and Trust indicators Social proof elements, such as logos of partner companies on the website or endorsements of reputable institutions, can foster trust in the application. In addition, trust symbols (e.g. locks and keys), third party security certifications reassure users about the credibility of the SSI wallets. This aligns with established best practices in web security and trust design [37].

Minimalist and Modern Design A well-designed and aesthetic interface contributes to usability and trust. [27]. While users usually do not acknowledge good design, they are quick to point out deficiencies such as cluttered content, overflowing containers, and inconsistent elements, which contribute to distrust [37]. The Aesthetic-Usability Effect describes how users tend to perceive more visually attractive interfaces as easier and more pleasant to use, even if the functionality is identical [34]. In addition, studies have shown that visual appeal is strongly correlated with the perceived trustworthiness of a system [27]. Research suggests that SSI wallets should have a minimalist and modern design, including readable font sizes, high contrast, and a harmonic color choice, to improve usability and build user trust [40]. A modern and aesthetically pleasing interface reassures users that they are interacting with a sophisticated and professional system [2]. Trust symbols, such as padlocks or shield icons, could be incorporated into the interface to indicate encrypted data, verified credentials, or secure transactions, similar to the lock symbol in the address bar of a web browser in TLS communication [30].

Reducing Cognitive Load Using a DI wallet is a higher cognitive burden than traditional authentication systems. A study on email encryption found that users struggled with key management due to unclear mental models and a high cognitive load [47]. Although you do not have to deal with keys directly in SSI software, similar usability challenges can be found, particularly in understanding the concepts of seed phrases, DID addresses, and

the principle of self-sovereign identity as a hole. Automated workflows, visual guides, and simplified interactions (e.g. one-tap approval for credential sharing instead of multiple pop-ups or steps) can help combat these challenges [28].

Error Handling and Feedback Providing clear feedback when users make mistakes is crucial for good usability. Wallets should not only block incorrect actions, but also explain why they are not permitted, and how the issue is resolved [40]. For example, if credentials transfer fails due to security policies, explanatory messages improve user confidence and limit frustration [37].

2.2.4 Extracted Features from Guidelines

The above guidelines and recommendations underpin the following set of concrete UX features that can be used not only to evaluate DI and SSI applications, but also to guide their design and development. These features are intended to address common usability challenges while fostering trust and transparency.

1. **Simple Terminology**
   (a) Use clear and non-technical labels on buttons and system messages [40].
2. **Mental Model and Intuitive Navigation**
   (a) Establish easy and consistent interaction patterns, such as obtaining credentials or backing up data [40].
   (b) Minimize the number of steps for key actions [17].
   (c) Provide a well-structured layout and clear navigation elements. Important functions should be found quickly [17].
   (d) Give users immediate and understandable feedback after each action to confirm that the system has processed their input.
   (e) Build on familiar mental models. Design interfaces that align with the existing experiences and expectations of users. Use familiar patterns, icons, or metaphors that users already know about from other tools or everyday contexts. For example, credentials should be visualized as cards or documents that can be stored in a physical wallet, reinforcing the concept of ownership and control [36].
3. **Minimalist and Modern Design**
   (a) Design a clean and modern user interface with an aesthetically appealing color scheme.
   (b) Incorporate intuitive icons and meaningful visual elements to enhance user experience. Trust and security symbols can be used. [36]

(c) Use a readable font with high contrast for accessibility. Appropriate font sizes should be maintained.

4. **Backup and Recovery Options**

   (a) Offer multiple backup solutions, including cloud and local options [40].

5. **Educate and Support Users**

   (a) Provide short and relevant instructions for the most important features.

   (b) Implement onboarding tutorials that can be skipped and revisited [40].

   (c) Include help and info buttons (e.g., '?' or 'i' -symbols) for contextual help [40].

   (d) Offer FAQs, chat bots, or help hotlines.

6. **Trust through Transparency**

   (a) Clearly communicate how the system works and how user data is stored and handled. Use visual cues, explanations, and real-time feedback to help users understand key processes and build confidence in the reliability of the application. [40].

7. **Minimize Cognitive Load**

   (a) Avoid large quantities of text and redundant information [40].

   (b) Present information in a structured and organized way to support quick scanning and understanding. Use grouping, sectioning, and layout techniques that reduce clutter and guide the user's attention. Support efficient navigation by including search and filter functionalities.

   (c) Use the progressive disclosure design pattern, which gradually reveals information to avoid overwhelming the user.

8. **Social Proof**

   (a) Display logos of trusted partners or certificates to enhance credibility.

   (b) Ensure a strong corporate identity by prominently displaying the logo on the start-up screen with a professional and trustworthy look.

9. **Error Handling and Feedback**

   (a) Provide clear error messages and warnings when actions fail [40].

   (b) Prevent user mistakes by explaining why an action is restricted [37].

10. **Security Measures**

    (a) Support biometrics and two-factor authentication [17].

2.3 Evaluation Setup and App Selection

The goal of this analysis is to evaluate the usability and user experience of three decentralized identity and self-sovereign identity application based on previously extracted usability and UX features. It aims to assess how well the application align with the established guidelines and recommendations in the field. The features are cited directly within the text using their corresponding number and letter in brackets. (e.g., 5.a)

The selection of SelfKey, PrivadoID and Truvera was influenced by their presence on numerous websites [1, 8, 11, 13, 16, 53] that discuss the topics of DI and SSI. (PrivadoID is the successor of PolygonID and Truvera is mostly referred to as its former name Dock.) Many other SSI services did not offer a free mobile or web application or were only available to businesses, making them unsuitable for evaluation. PrivadoID is explicitly advertised as an SSI application in the iOS App Store.

The testing was carried out using an iPhone 12 running iOS 18.1.1 and a MacBook Pro with macOS Sonoma 14.3, utilizing Google Chrome as the primary browser. The evaluations were performed on 11 February 2025.

2.4 SelfKey

SelfKey is a blockchain-based self-sovereign identity system and Ethereum wallet developed by KYC Chain, a Hong Kong-based solution provider specializing in Know Your Customer (KYC) processes since 2013 [10]. Through the wallet, users can securely store personal information, manage ERC-20 tokens, and access a marketplace that offers services such as business incorporation and bank account openings. They aim to enhance privacy and provide greater freedom in the management of personal data [39].

2.4.1 Application Setup and Onboarding

The corporate identity seen in the startup screen does not seem particularly trustworthy (8.b). Additionally, the lengthy Terms of Service (i.e., documents that cannot be quickly scanned or understood within a short interaction time) presented during setup may discourage users, as studies show that most users do not read these agreements in full due to their excessive length and complexity [32]. This creates unnecessary cognitive load at the very first point of interaction (7.a).

On the positive side, the application enforces a strong password (10.a) and the creation of a seed phrase for recovery purposes (4.a). As seen in Figure 2.3, the users are required to write the seed phrase in a playful and interactive way. This prevents the user from losing the seed phrase. Additionally, the user is informed about the importance of a seed phrase with brief and appropriate instructions (5.a) [Figure 2.2].

However, SelfKey lacks an onboarding tutorial, making it difficult for new users to understand how to use the application (5.b). Furthermore, enabling Face ID to unlock the

application does not provide any feedback, leaving users unsure whether their setup was successful (10.a, 9.a, 2.d).

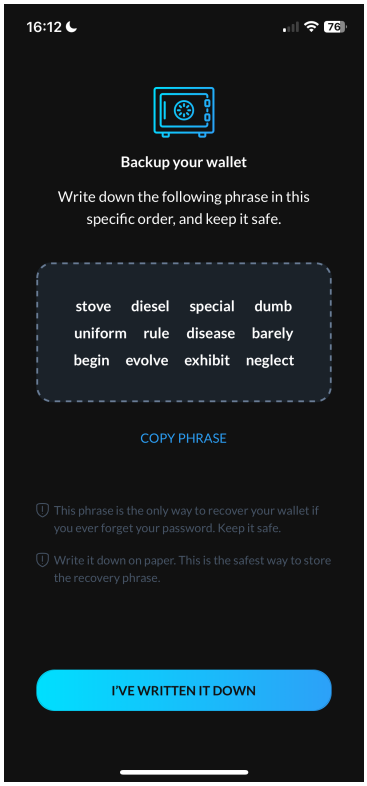


Figure 2.2: Seed Phrase Backup Process: Users get information about the seed phrase and are requested to write it down.

Figure 2.3: Seed Phrase Confirmation: Users must confirm the correct order of the seed phrase before proceeding.

2.4.2 Appearance and General Functionality

Figure 2.4 shows the application dashboard. The design relies heavily on shades of blue, which can have a calming and trustworthy effect. However, its overuse without contrast makes the interface feel cold and visually monotonous (3.a) [41]. There are also layout problems with certain elements falling out of the viewport, as can be seen in Figure 2.5

The application creates too much cognitive load, due to its strong focus on cryptocurrency features, which can overwhelm users who are not familiar with blockchain technology (7). However, readability is good and the application has a high-contrast font color (white on dark blue), ensuring accessibility (3.c).

A positive aspect of the application is that it includes information icons (represented by the letter 'i' in circle) that provide brief, contextual explanations for certain sections, as

seen in Figure 2.7 (5.c). These icons help users understand the purpose of a section or action without needing to leave the interface.

The application also provides a link to a Discord support channel, offering users an alternative way to receive assistance and ask questions (5.d).

The application lacks a dedicated credential section but instead, credentials are located under the 'Profile' tab, which may not be intuitive for users. It thereby misses the commonly used metaphor of representing credentials as cards in a digital wallet, which could have improved recognizability and ease of use (2.e). A significant technical issue is that the application crashes when switching blockchain networks, which questions the competence of the developers and therefore potentially undermines trust. Research indicates [42] that the stability of blockchain applications is crucial for user trust and adoption.

A positive aspect is that the application uses clear and non-technical labels for buttons, which enhances accessibility (1.a). It also avoids large quantities of text, which helps reduce cognitive load, except for the terms and services in the beginning (7.a).

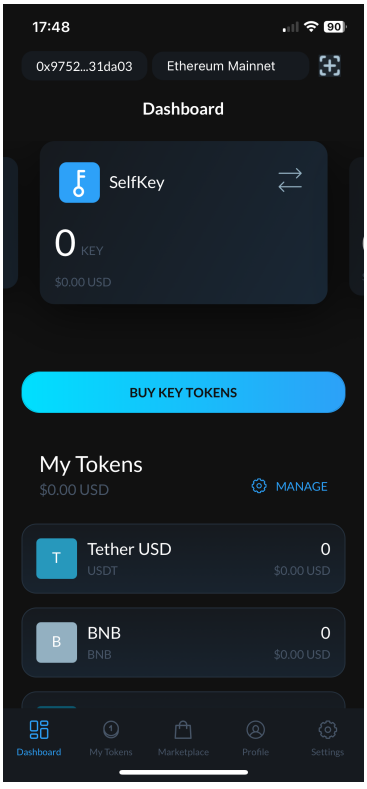


Figure 2.4: Dashboard Overview: The application's dashboard displays the user's wallet, token balances, and marketplace access.

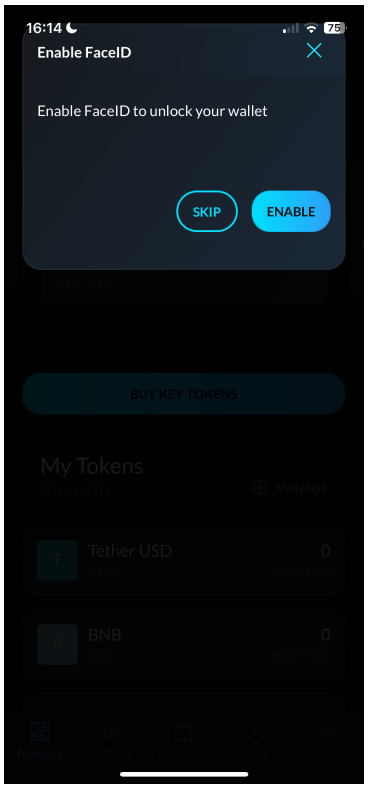


Figure 2.5: Layout Issues: Popup element is not centered and slightly extends the viewport.

2.4.3 Claiming Credentials

The credentials can be claimed in the profile section of the application. It is not entirely clear what the profile is in the context of this DI application (5.a). Figure 2.6 shows the screen when users navigate to the profile section for the first time. The application does not clearly explain how personal information such as the user's name, email address, and profile picture is stored or used, resulting in a lack of transparency regarding data management (6.a).

The process of uploading documents is simple, as it does not involve many steps (2.b). However, there is a notable usability issue with the date selection widget, when the user has to select the expiration date of the credential, for example. Its very poorly implemented making it difficult for the user to select the right date.

A lack of feedback after document submission presents a major usability and trust issue (2.d). As shown in Figure 2.7, the application provides no confirmation that the submission was successful, nor does it inform users who will review the document, how long it may take, or whether a response can be expected at all. This uncertainty leaves users feeling ignored and unsupported, which undermines their confidence in the system. Prior research highlights that transparency and timely feedback are key components of building user trust in digital systems, especially in privacy-sensitive contexts like identity verification [**selling2023research**, 26]. Users also do not receive feedback when they delete a credential, which creates uncertainty about whether the data have actually been removed properly (2.d).

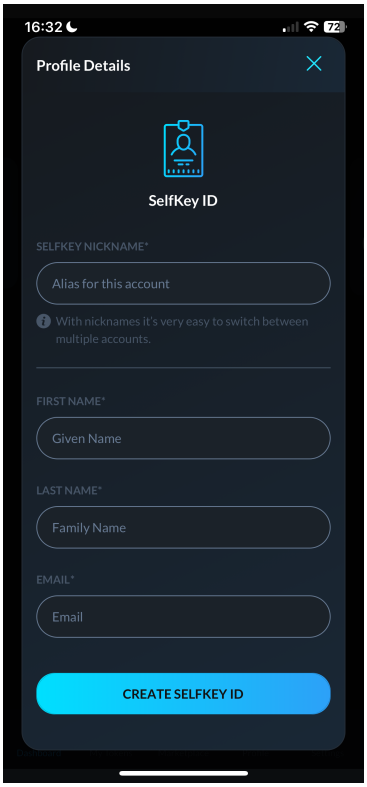


Figure 2.6: Profile Creation: Users are requested to set up their profile by entering a nickname, personal details, and email.

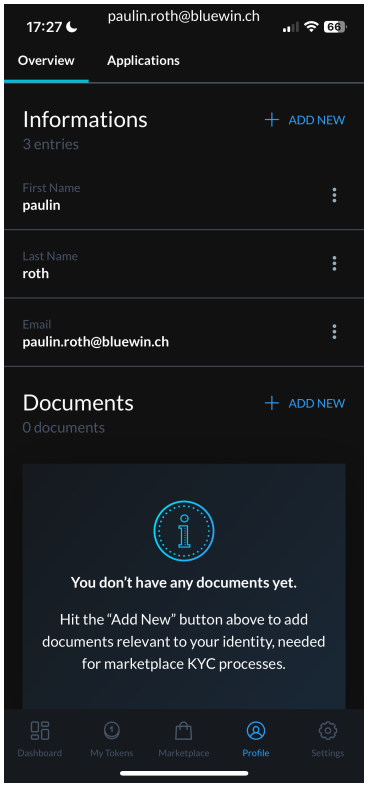


Figure 2.7: Credential Management: The profile section displays user credentials and allows adding new documents. Big ”i”-symbol for contextual help

2.4.4 Backup and Recovery

SelfKey allows users to review the seed phrase again in the settings, which is protected by password or Face ID, securing it against unrestricted access (10.a). However, no alternative backup solutions are offered, which is a significant drawback (4.a). Seed phrases are not intuitive for most users and can easily be lost. Relying solely on this method can lead to accessibility and security issues.

2.4.5 Website and Desktop Version

The macOS version of SelfKey cannot be installed because the operating system cannot verify that it is free from malware. This significantly reduces trust in the software and may deter users from installing it in the first place. The website is generally well designed, with appropriate visual elements and animations and a lot of information (3.b, 5.a). It also includes a FAQ section with some important questions (5.d).

2.4.6 Final Assessment

SelfKey offers some basic usability benefits, such as password enforcement (10.a) and a structured seed phrase recovery process (4.a, 5.a). It also incorporated some visual elements and security symbols (e.g. a key) and many information symbols (letter 'i' in circle). However, the overall user experience is limited by significant shortcomings in trust, transparency, and usability. The lack of onboarding (5.b) makes it difficult for new users, and navigation and layout aspects also need significant improvement. However, the most concerning aspect is the lack of transparency in data handling (6.a). The application does not clearly indicate who the issuer of the credential is, nor does it provide accessible explanations about how personal data such as name, email, or uploaded documents are stored, processed, or shared. This lack of clear communication can lead users to question whether their data is being collected or used without proper safeguards, ultimately undermining trust in the system.

To improve the application, SelfKey should implement the following:

- A onboarding tutorial (5.b)
- A more modern and user-friendly UI (3.a)
- Improved transparency on data handling and credential issuers (6.a)
- Feedback mechanism for important actions (2.d)
- Provide alternative backup and recovery solutions (4.a)

Addressing these issues may lead to better user experience and foster trust, and ultimately improve the adoption of the application.

2.5 PrivadoID

PrivadoID is advertised as a self-sovereign identity platform that empowers users to create and manage their decentralized identity. Originally developed by Polygon Labs, PrivadoID has become an independent entity to address the global demand for secure digital identity solutions. The platform offers features such as in-application identity verification and reusable credentials for age verification and Know-Your-Customer (KYC) processes. It uses zero-knowledge proofs to allow users to prove access rights and reputation without revealing personal information [23].

2.5.1 Application Setup and Onboarding

The welcome screen of the application is well designed with a strong brand presence, which contributes to trustworthiness (8.b). As can be seen in Figure 2.8 information

about data usage is clearly provided at the beginning, improving transparency (6.a). Security measures such as PIN and Face ID are available (10.a).

Despite these advantages, the setup process has some shortcomings in user guidance. When creating an account, users must choose between creating a local wallet or signing in with an external cryptocurrency wallet sucha s MetaMask. There is no explanation of which option is best suited for which use cases (5.b). There is no onboarding tutorial to become familiar with the core functionalities of the application (5.b).

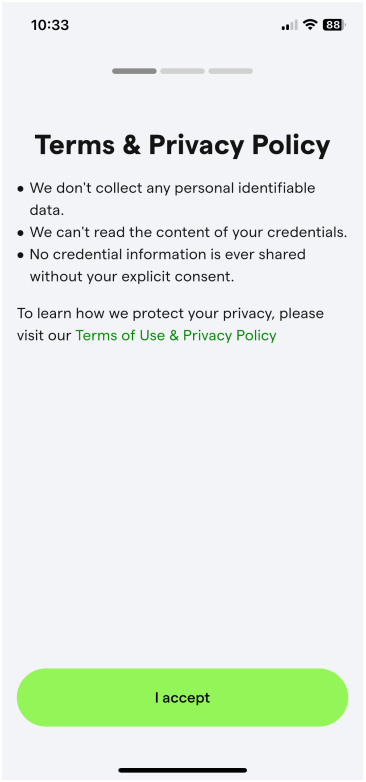


Figure 2.8: Transparency in Data Usage: The application informs users about its privacy policy, emphasizing that no personal data or credential contents are collected or shared without consent.

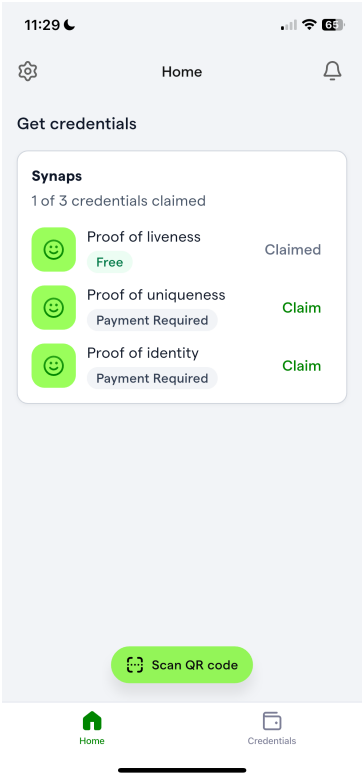


Figure 2.9: Claiming Credentials: Users can obtain different credentials, such as proof of liveness, uniqueness, and identity. Settings access in the top left corner.

2.5.2 Appearance and General Functionality

Figure 2.9 shows a screenshot of the home section. The design of the application is modern, clean, and visually appealing, making it seem professional and trustworthy (3.a). The well-structured and minimalist layout, with only two sections in the bottom navigation bar, allows the user to quickly find relevant functions such as QR code scanning and credential management (2.c). Technical terms like DID are hidden away in the settings menu, which

ensures that the application does not deter non-technical users (1.a). The labels and buttons are clear and easy to understand (1.a). The credentials are depicted with a wallet icon, which is a good choice because it corresponds to a traditional wallet where the credentials are stored (3.b). Users can quickly perform actions such as scanning QR codes or obtaining credentials, which contributes to a smooth experience (2.b). Additionally, font choice and text size, as well as high-contrast color scheme, ensure readability and an aesthetic look (3.c).

However, the application lacks security and trust symbols, which could further enhance trust (3.b).

2.5.3 Claiming Credentials

The application initially lets you claim three credentials: Proof of liveness, uniqueness and identity' (Figure 2.9). Although this is generally a good feature, there is no explanation on what the purpose of these credentials is (5.a). A short explanation text is provided once the credentials have been claimed. This is useful, but it might have been more helpful if it had been shown before obtaining the credential.

The process of claiming the credentials is mostly straightforward, however there are some areas that need improvement. It is generally very good that a chatbot is available during the process of claiming credentials as can be seen in Figure 2.10 (5.d). However, it is questionable that the chatbot is only available in this specific scenario and not elsewhere in the application. To obtain the credentials, PrivadoID uses a face scanning software provided by Synapse. As seen in Figure 2.11, a video and a link to the terms and services are provided. Although this is well designed, the video itself lacks meaningful information. It only shows how to scroll through the legal agreement and accept it. It would have been more helpful if the video provided the most important information on how facial data is being handled, contributing to greater transparency in data processing and storage (5.b, 6.a). The legal statements are long and not very effective in informing users (7.a) [32]. Another aspect that could be improved is that users are required to perform the same face scan for all three credentials, which feels redundant and could be simplified (2.b).

Two of the three credentials require a payment in cryptocurrency. The payment process is highly confusing with some bugs in the MetaMask integration. The application does not specifically inform users about which cryptocurrency or blockchain network should be used for payment, which causes frustration (2.a).

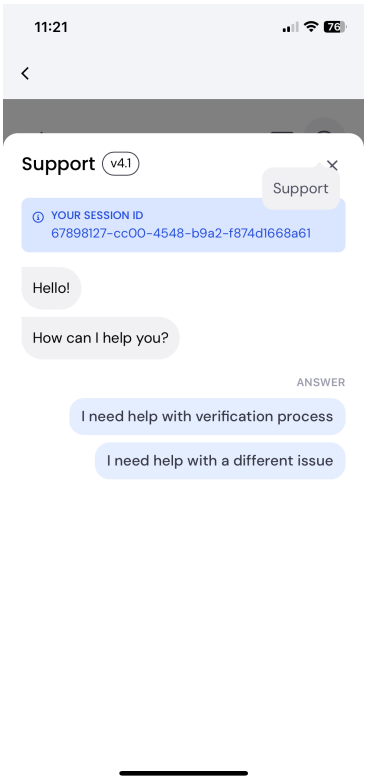


Figure 2.10: Chat bot Assistance: A chatbot is available during the credential claiming process, providing real-time support. However, its availability is limited to this step, reducing its usefulness in other areas of the application.

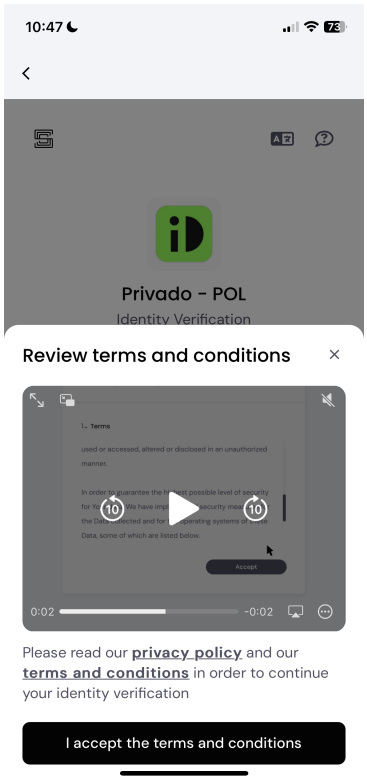


Figure 2.11: Terms and Conditions Confirmation: Users must review and accept the terms and conditions before proceeding. The application includes a video tutorial, but it lacks essential information about how facial data is processed.

2.5.4 Handling Credentials

Figure 2.12 displays the application's credential section, which is well designed and follows a familiar layout seen in other wallet applications, where credentials are displayed as credit cards (2.e). Users can filter credentials according to their status (active, expired, or revoked), which improves organization and usability and reduces cognitive load if there are many credentials (7.b). The interaction to add a new credentials can be found easily with the plus icons and 'add'-label in the top right corner (2.c). Another positive aspect is that the credentials section follows the progressive disclosure design pattern. In the credentials element itself, the most important information can be found, such as title, description, expiration date, and issuer. By clicking on the element, the user can access more detailed information, as seen in Figure 2.13. It makes sense not to present this information upfront, because it is rather technical and could overwhelm the users (7.c).

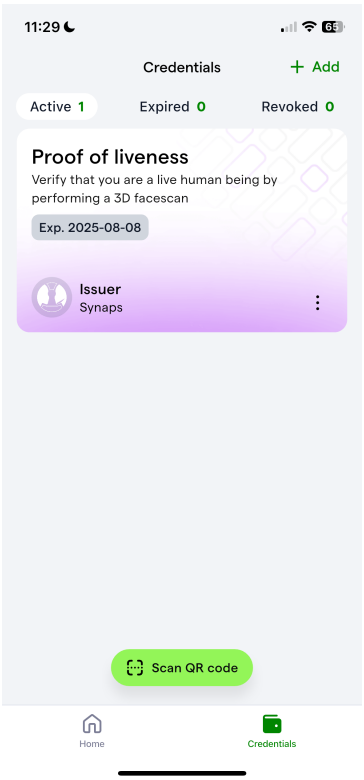


Figure 2.12: Credential Management: The credentials section is well-organized, using a familiar wallet-like interface. Users can filter credentials by status, improving usability.

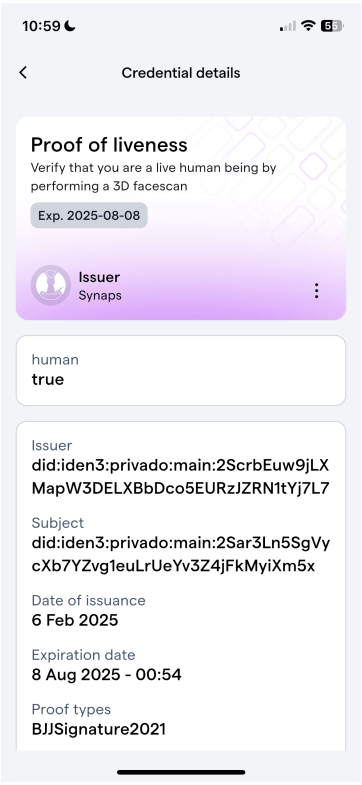


Figure 2.13: Credential Details: More detailed information is presented once users click on the credential element.

2.5.5 Backup and Recovery

Figure 2.14 shows that the application provides clear warnings about the importance of keeping private keys secure (6.a). To see the private key, the user must press the 'reveal private key' button and unlock with a pass code or face Id. This prevents unauthorized users from accessing sensitive data, improving security and trust (9.b). The application offers a recovery mechanism with the external wallet MetaMask (4.a). It provides a short information text along with a link to a documentation page (5.a, 7.a).

2.5.6 Website and Desktop Version

The PrivadoID website is well designed and aesthetic and appears very trustworthy with a competent team behind it (3.a). It prominently displays logos of its ecosystem and

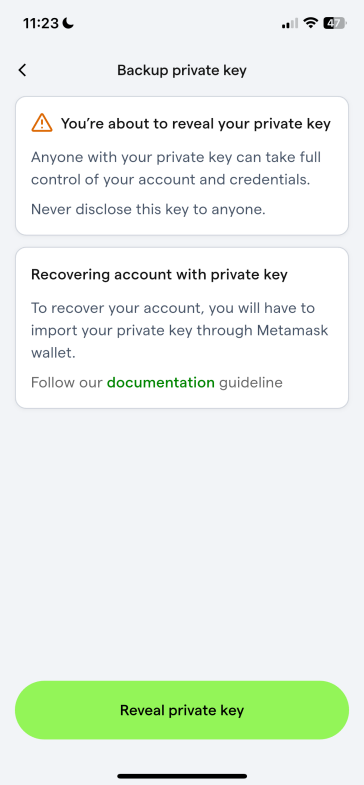


Figure 2.14: Backup and Recovery: Users are given clear warnings about the importance of securing their private key.

trusted partners, which reinforces credibility (8.a). In addition, the website includes an FAQ section that provides helpful information (5.d).

However, finding a link to the web wallet (desktop version) of the application is difficult (2.c). This may cause confusion for first-time users trying to access it.

In Figure 2.15 the desktop version of privadoID is shown. The Web application is visually clean and aesthetically consistent with the mobile application (3.a). However, there are some significant functional limitations.

Users are not informed from the start that signing in with an external wallet is required to use the desktop version synchronously with the mobile version, which means that without this integration, the credentials obtained on the phone do not appear on the desktop. This can lead to confusion (5.a). Additionally, the desktop version lacks some important features, including settings and the ability to obtain credentials. Instead, it promotes a marketplace that is still labeled 'coming soon'. Furthermore, there is no available help, support or chatbot section (5.d).

2.5.7 Demo Functionality

PrivadoID offers a demo on its website that allows users to issue credentials themselves and exercise the role of Issuer and Verifier. Its generally a very good idea and can help

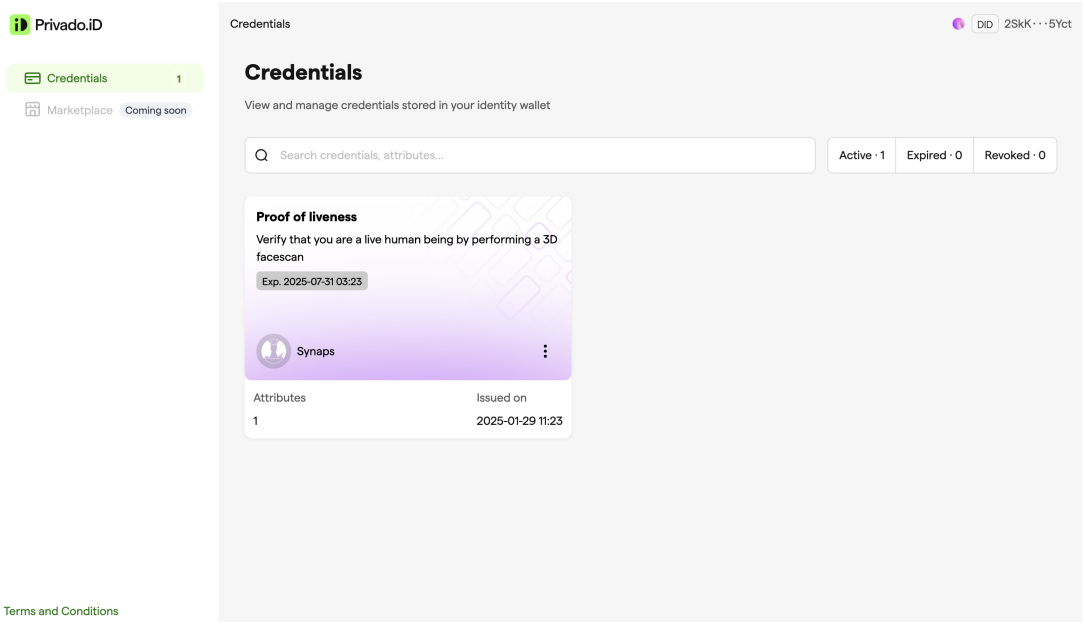


Figure 2.15: Desktop Version: The web version of PrivadoID maintains a clean design but lacks key functionalities such as the ability to claim credentials, making it feel incomplete.

improve the user understanding of the underlying technology, such as the triangle of trust. However, the demo does not work properly. When users generate and scan a QR code to obtain mock credentials, they encounter an error without explanation (9.a).

2.5.8 Final Assessment

PrivadoID stands out with a very usable and aesthetically pleasing application. The modern design (3.a), clear structure (2.c), fast performance (2.b), and easy credential management (7.c) contribute to a simple and enjoyable user experience.

However, the application has critical functional shortcomings. The lack of clear onboarding (5.b) makes it difficult for new users to understand the application. Key features such as payment processing (2.a) and the demo functionality (5.d) are unreliable, which undermines user confidence. Furthermore, transparency on data usage should be improved (6.a) and more security symbols could be incorporated to reinforce trust (3.b). The desktop version of PrivadoID is visually appealing but lacks essential functionality (2.a), making it feel like an incomplete version of the application.

To improve, PrivadoID should focus on:

- Providing better onboarding and guidance for users (5.b).
- Fixing bugs related to payment processing and MetaMask integration (2.a).
- Fixing the demo functionality so users can fully experience the issuer and verifier roles (5.d).

- Making legal agreements more digestible, such as using short videos to summarize key points (5.a).
- Including security and trust symbols to strengthen credibility (3.b).

By addressing these issues, PrivadoID can further improve its application and enhance usability, trust, and overall user satisfaction.

2.6 Truvera

Truvera is a decentralized identity platform developed by Dock Labs, an IT company registered in Switzerland [33]. It allows businesses and identity verification providers to issue and manage digital credentials. The platform helps organizations share and verify customer information securely, making onboarding easier and improving trust in digital transactions [12].

2.6.1 application Setup and onboarding

Truvera provides a moderate trustworthy brand presentation (8.b). The application enforces password creation and offers Face ID authentication that provides security and convenience (10.a). The application gives clear feedback on successful Face ID activation, which enhances UX (2.d).

However, an onboarding is missing, making it difficult for new users to see how they can best use the application (5.b).

2.6.2 Appearance and General Functionality

The user interface is clean and minimalist, with high-contrast colors and readable and appropriate sized font (3.a, 3.c). The layout is well-structured, allowing users to quickly find essential functions such as QR code scanning (2.c). The application also provides good feedback on user actions in general (2.d).

Despite these strengths, the overall design is not particularly modern or aesthetic (3.a) and generally lacks visual elements or trust symbols (3.b). For instance, there is no credential symbol (e.g. wallet or credit card). Furthermore, there are no 'i'-symbols or '?'-symbols for contextual help, which means that users do not receive any further explanation about important functions and features (5.c). The application also does not provide any information on data storage or offer a link to terms and conditions, which reduces transparency (6.a). The absence of a chatbot or help desk further reduces user support options. (5.d). In Figures 2.16 and Figure 2.17, the credentials and scan sections of the application are displayed. Although the application provides brief texts, it offers little actual guidance beyond stating the obvious. For example, the credentials section

simply informs users that the credentials will appear once accepted, and the scan section instructs users to place a QR code inside the box. This minimal level of information does not effectively support users in understanding the functionality or purpose of the application.

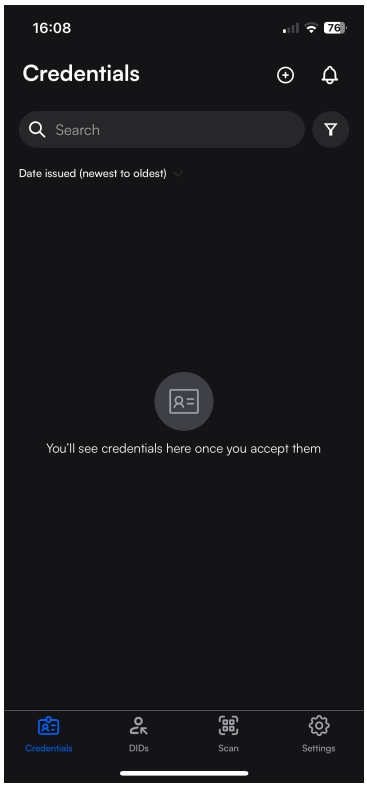


Figure 2.16: Credentials Overview: The application provides an empty credential wallet. Users can upload credentials via the plus button in the top right corner.

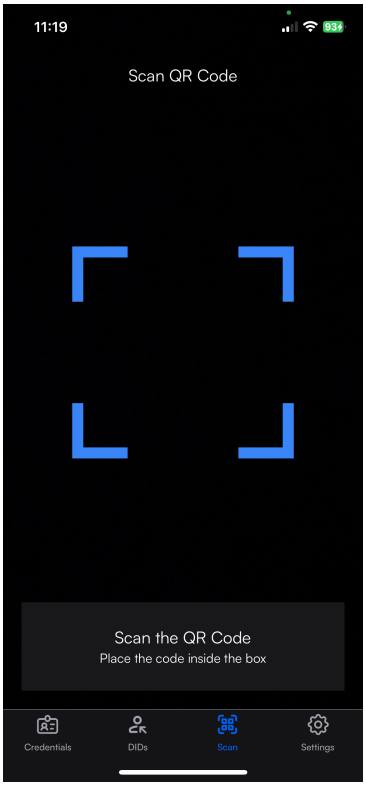


Figure 2.17: QR Code Scanning: The application uses QR code scanning for obtaining credentials.

2.6.3 Claiming Credentials

The process of adding credentials is straightforward and intuitive (3.b), with a plus button in the top right corner of the screen, as can be seen in Figure 2.16. The application also includes search and filter functionality, making it easy for users to organize and manage their credentials (7.b).

When clicking the plus button, the application allows users to upload a file. However, it does not explain what type of file is suitable, leaving users uncertain about the requirements (5.a). When users upload a file (e.g. pdf or png), the application crashes and there is no feedback on what went wrong. (2.d). Such failures not only disrupt the user experience but also violate basic expectations of system reliability [42].

2.6.4 DID section

Figure 2.18 shows the DID section where users get an overview of their DIDs, and Figure 2.19 shows how the application allows users to create one. As mentioned above, DID is a technical term that should not be prominently featured in the application without proper user guidance (1.a). The sections shown in Figure 2.18 and Figure 2.19 lack essential information, making it difficult for users to understand their purpose and functionality (5.a).

The application allows users to create a new DID and select between two types: did:key and did:dock. There is some information, as can be seen in Figure 2.19, but it is not adequate, as it does not sufficiently state what the difference between those two is and in which situation either one of them is needed (5.a). In addition, there is no guidance on with whom credentials should be shared.

Overall, the DID section feels too technical, making it inaccessible for users unfamiliar with DI concepts.

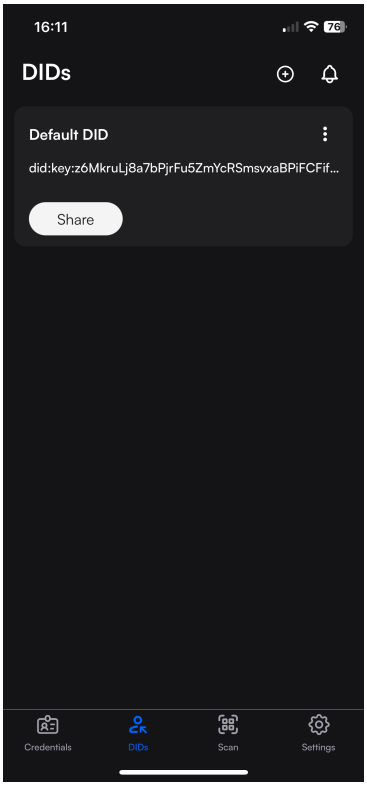


Figure 2.18: Decentralized Identifiers: Users can view and share their default DID, but the application does not provide an explanation of its role in the identity ecosystem or how it interacts with credentials.

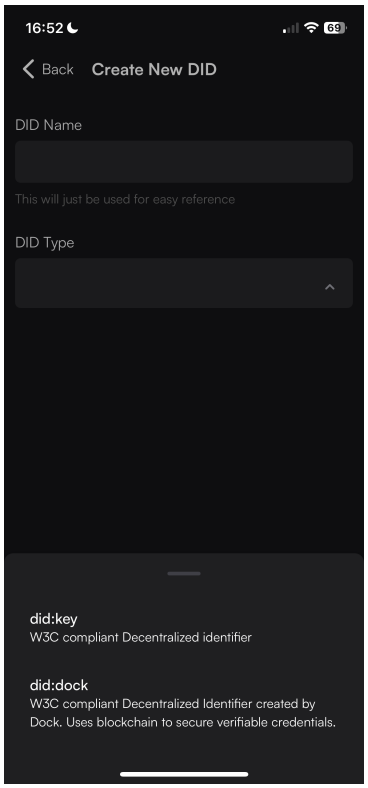


Figure 2.19: Creating a New DID: Users can generate new DIDs and choose between different DID methods. There is some information provided, but it might still be unclear.

2.6.5 Backup and Recovery

The backup process is not intuitive. When users make a backup, they are requested to download a JSON file, but there is no explanation of what this file is, how it should be stored, and how it is used for recovery (4.a, 5.a).

On the positive side, the application prevents users from deleting their wallet without first creating a backup, helping protect against accidental data loss (9.b).

2.6.6 Website and Desktop Version

The Truvera website is well designed, with clear information (5), visualizations (3.b) and trusted partners to reinforce credibility (8.a). These factors contribute to a trustworthy corporate identity and brand presentation (8.b).

However, singing into the web application (desktop version) is problematic. Users can sign in by scanning a QR code or with third parties like Google or Github. Scanning the QR code with the Truvera wallet results in an error without any explanation (9.a). Signing in with GitHub grants access permissions, but users are not actually able to access the Truvera desktop version afterward, making it feel like the application collects user data without providing any value in return. After reaching out to customer support, it was clarified that the desktop version is only available for businesses. This information could have been clarified in advance. These misleading authentication processes and unclear access restrictions undermine trust in the application and the company behind it.

2.6.7 Demo functionality

Truvera provides a demo feature to educate the user. It allows users to download a banking application called 'Quotient Wallet' which mimics Truvera and allows users to interact with a fictional bank and obtain mock credentials [44]. While this is an excellent idea to make users more familiar with the workflows, the demo does not work properly, as it results in an error message without any explanation (5.d, 9.a). This issue prevents users from fully completing the demo, which limits its effectiveness.

2.6.8 Final Assessment

Truvera has a clean UI with intuitive navigation. However, the application suffers from significant usability gaps, transparency issues, and some technical failures.

The DID section is too technical, making it inaccessible to average users (5.c). Additionally, many core functionalities lack explanation or do not work properly, such as file uploads causing crashes, authentication failure on the web application (9.a.) and the non-functional demo.

Lack of onboarding (5.d) makes it difficult for new users to understand how the application works and what the actual use case of the application is. The absence of clear data storage policies reduces trust (6.a). The backup and recovery process is also unclear (4.a), and there is little guidance on the sharing of credentials and the use of DID (5.a, 5.c).

The most concerning aspect is the failure of critical features such as file uploads, authentication, and demo functionality. If users cannot rely on the application's core functionalities, it is difficult to trust its security and identity verification mechanisms.

Recommendations for Improvement:

- Fix critical errors such as file upload, web application sign-in, and demo (9.a).
- Introduce an onboarding tutorial to guide users through DIDs, credential management, and data security (5.b). Explain what a DID is and how it can be used.
- Ensure transparency by including clear data handling policies (6.a).
- Incorporate more visual elements, such as credential symbols or trust symbols.
- Improve the backup and recovery process by explaining how the exported file should be stored and used (4.a).

By addressing these issues, Truvera can significantly improve usability, transparency, and trust, and therefore make it a more reliable and user-friendly self-sovereign identity solution.

2.7 Problem Statement

Decentralized Identity and Self-Sovereign Identity systems provide greater security and control over personal data for the user. However, despite these advantages, usability and UX challenges remain a major problem for mass adoption. Many users find these systems difficult to understand and use, due to complex underlying concepts, lack of guidance, and poor interface design. The following key issues represent the main usability and UX problems in current DI and SSI application.

Lack of Information on Functionality One of the primary issues is the lack of adequate information and contextual help within DI and SSI application. Many do not provide onboarding tutorials or in-application guidance, leaving users to figure out the system on their own. Most of the time, a lot of detailed information can be found on the website, but many users do not conduct thorough research before installing and using the application. This issue is especially problematic for new users who are unfamiliar with decentralized identifiers, verifiable credentials, and cryptographic keys, because those concepts do not align with traditional mental models of identity management.

Unclear Use Case and Purpose In addition to the lack of information about the underlying technology, users often also struggle to understand the application's purpose and when or how they should use it. In the applications analyzed, SelfKey, PrivadoID, and Truvera, it was often unclear how and where these applications could be beneficial. The lack of clear use case instructions can make the application feel pointless and disconnected from real-world needs. Without a clear use case, users are unlikely to engage with the technology.

Absence of Mental Models A key challenge in the adoption of DI and SSI is the absence of established mental models. Unlike traditional identity systems, where users understand the concepts of username and password, decentralized identity systems introduce unfamiliar elements such as DIDs and cryptographic keys and signatures. Without a clear mental model, users struggle to predict how the system behaves and are therefore hesitant and distrustful. Many DI and SSI applications fail to address this, leaving users to navigate complex processes without structured guidance. To improve adoption, applications must actively create and reinforce intuitive mental models. UX designers must develop creative solutions to simplify complex concepts, making them more accessible through intuitive onboarding, contextual assistance, and interactive learning experiences.

Lack of Transparency and Feedback Another significant issue is the lack of transparency in data usage. Users are often left uncertain about where there data is stored, who has access to it and what happens after they delete their credentials or account. In SelfKey, this problem was further reinforced by the lack of feedback after uploading personal documents. The users were wondering if the process was successful and what the next steps were. Similarly in privadoID, after submitting the facial data for the proof of liveness, the user was not informed who would analyze and confirm the data and if it would be stored on the servers of a private company. Although legal agreements are present, they are generally not useful, as most users do not read them [32].

Poor Visual Design Finally, visual design plays a crucial role in user experience, yet many DI and SSI applications suffer from outdated interfaces and cluttered layouts. Inconsistencies or poorly designed user interfaces make it difficult to complete tasks effectively. Furthermore, visual elements and security indicators, such as verified issuer badges or security icons, are often missing, reducing the user's confidence in the reliability of the system.

2.7.1 Positioning of the Thesis

While a number of studies have addressed individual challenges in the UX of DI and SSI systems, the literature remains fragmented in scope and depth. This thesis builds upon established research by consolidating key usability and trust-enhancing guidelines into one coherent catalog, explicitly focusing on transparency and user trust.

In contrast to prior work, this thesis contributes a prototype that supports all three roles of the Triangle of Trust: issuer, holder, and verifier. It covers the full credential lifecycle, including request, issuance, presentation, approval, and revocation, thereby enabling a more complete evaluation of usability and trust. This approach exposes frictions that are often overlooked in single-role demonstrators and provides a more realistic basis for usability analysis.

Moreover, while existing studies mention the importance of trust symbols or simplified terminology, they often fall short of proposing specific, reusable interface solutions. This thesis bridges that gap by contributing a UI pattern library, which includes visual indicators for verified DIDs, secure on-device storage, and clear issuer provenance. The impact of these patterns is validated through user testing, demonstrating measurable improvements in perceived transparency and trust. This pattern library is presented in the section "Extracted Features from Guidelines" as a structured set of extracted features derived from the guidelines.

In sum, this thesis does not aim to reinvent the foundational guidelines but instead identifies and builds upon their strongest aspects, filling in critical gaps through implementation and empirical validation. The result is both a practical toolkit for developers and a set of empirical insights for researchers interested in improving the usability of decentralized identity systems. The developed prototype is designed to directly address all key aspects outlined in the problem statement.

Chapter 3

Design

The prototype 'Mask Identity' was designed to address key usability, trust, and transparency issues found in current SSI applications such as SelfKey, PrivadoID, and Truvera. These applications often suffer from poor onboarding and instructions, insufficient feedback, unclear mental models, and outdated user interfaces. Mask Identity proposes an alternative approach that prioritizes clarity, simplicity, and trust through thoughtful UX and UI design.

3.1 Style and Visual Identity

To avoid overwhelming the user, a minimalist visual style was chosen. Pages include little to no explanatory text unless necessary. Sometimes small info messages with icons are used that provide information, but do not clutter the user interface. Most interactions are guided by the layout, icons, and short labels. This keeps things simple and makes the application easier to use, even for people who do not know much about decentralized identity. Where needed, the application provides visual feedback through color, icons, and short messages to help users understand what is happening.

The primary color chosen for the interface is a highly saturated blue. It is used in buttons, links, and illustrations. Blue is often associated with calmness, trust and security, making it especially suitable for creating a trustworthy application [41]. The application uses a dark mode theme to reduce eye strain and give the interface a modern look. The background is dark and has a very slight blue tint, so it goes well with the primary color, and the containers also have a subtle bluish tint to keep the overall appearance consistent. The logo, cards, and some containers use color gradients to give the application a modern and visually appealing look.

All icons are sourced from the open source Phosphor icon library. They all have a similar style, making the application look consistent and well-designed. The consistency helps users quickly recognize concepts like verified addresses, for example, which are always marked with the same green verification icon.

The visual identity is reinforced by a minimalist logo in the top left corner. It symbolizes a mask which conveys anonymity and therefore fosters the users perception of control over the data. When users click on it, they are taken back to the landing page, which serves as an onboarding screen that reminds them of the application's core values and key features.

The application uses the font 'Nohemi' for headings, giving the application a modern look that reinforces a professional visual identity. 'Public Sans' is used for body text because it is clean and highly readable. Combined with a high contrast color scheme (white on a dark background), it appears clear and reduces cognitive load (Feature 3.c). In general, the application was designed to create a smooth and trustworthy experience. What follows is a walkthrough of all the different screens in the prototype, as well as an explanation of key design choices with references to the identified features in brackets.

3.2 Screen Walkthrough

3.2.1 Landing Page and Navigation

At the top of the interface, there is a navigation bar where users can quickly jump to different sections of the application (2.c) (Figure 3.1). It contains clearly labeled sections such as 'My credentials', 'Sharing', 'Settings', and 'Info'. To have sections like settings align with familiar mental models users have from different applications (2.e). The navigation bar provides users with a stable mental model of the application structure. It is further underpinned by the use of icons (3.b), like the gear icon for the settings page or the i-icon for the info page, which are familiar to many users.

The landing page serves as an onboarding page and introduces the user to the core values of the application with a short slogan: 'Take control of your identity' (5.a) (Figure 3.1). Below this is a prominent call-to-action button labeled 'My credentials', which redirects the user to the credentials section (2.c). The interface avoids clutter and focuses the user's attention. On the left side of the bottom part of the page, there is an illustration that depicts a person with credentials on their phone. This aims to convey the idea that users carry their digital credentials with them at all times, like a physical wallet, which is a familiar metaphor (2.e). The use of visual elements supports understanding and strengthens the identity of the application (3.b). To the right of the illustration there is a visual breakdown of key features (e.g., request, receive, manage, and share credentials, access services). The elements are accompanied by short descriptions (5.a) and icons (3.b). Their purpose is to support mental models and clarify use cases, two of the major weaknesses identified in the problem statement.

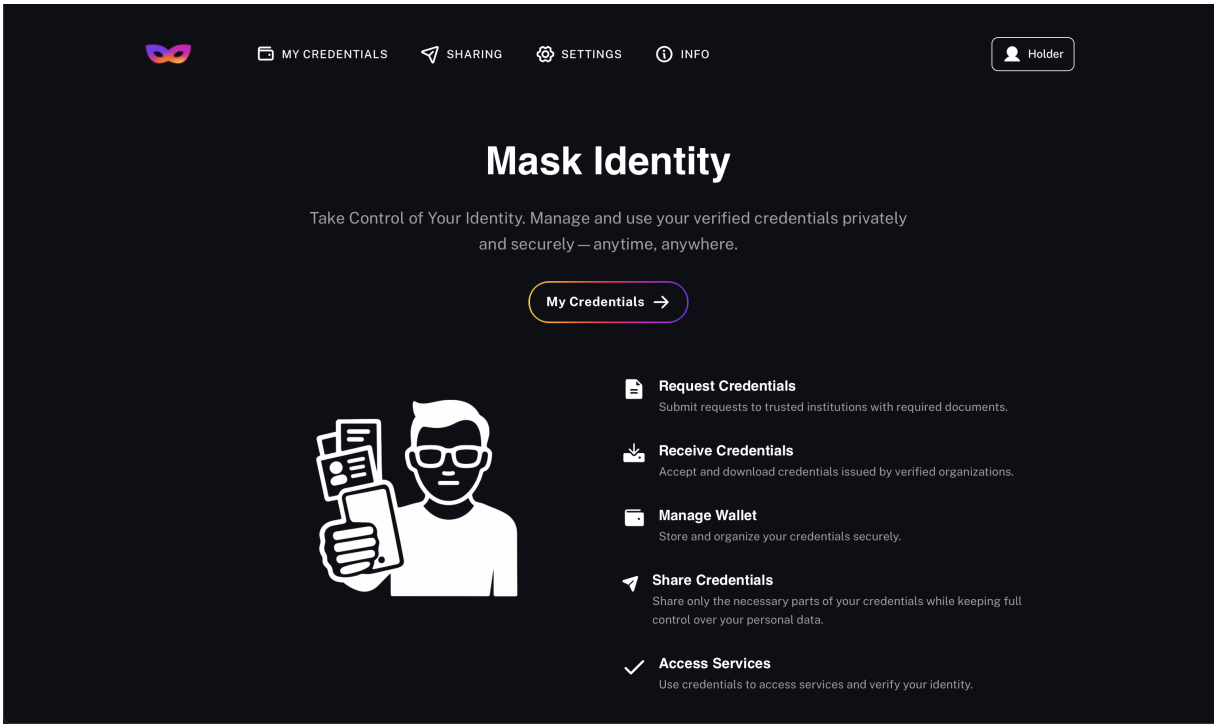


Figure 3.1: Landing page with a welcoming slogan, a central call-to-action button, and a visual overview of core application features.

3.2.2 Credential Overview Page

The second screen (Figure 3.2) provides an overview of the verified credentials in a card-based layout. This is an implementation of the best-practice approach of using the wallet as a metaphor (2.e). Each credential shows the most important information at a glance: title, status (e.g., Verified, Revoked), issuer, holder, and expiration date. This implementation reduces cognitive load, as users can quickly scan the cards and find the most important information (7.a). The cards show the logo of the issuer, improving transparency (6.a). The credentials states are clearly color-coded and labeled (e.g., 'Verified' in green, 'Revoked' in dark red). In the header, there is a prominent button '+ Claim New Credential' that takes users to a form where they can make a new request. It is purposefully big, such that users find this key action quickly (2.c). The footer of the credentials screen includes a message: 'All your credentials are securely stored on your device, not on a public server' (Figure 3.2). This addresses a core transparency issue mentioned in the problem statement. Users often do not know where their data is stored (6.a). The lock icon next to the message serves as trust indicators (3.b), strengthening the sense of security that is essential for establishing trust in decentralized identity systems.

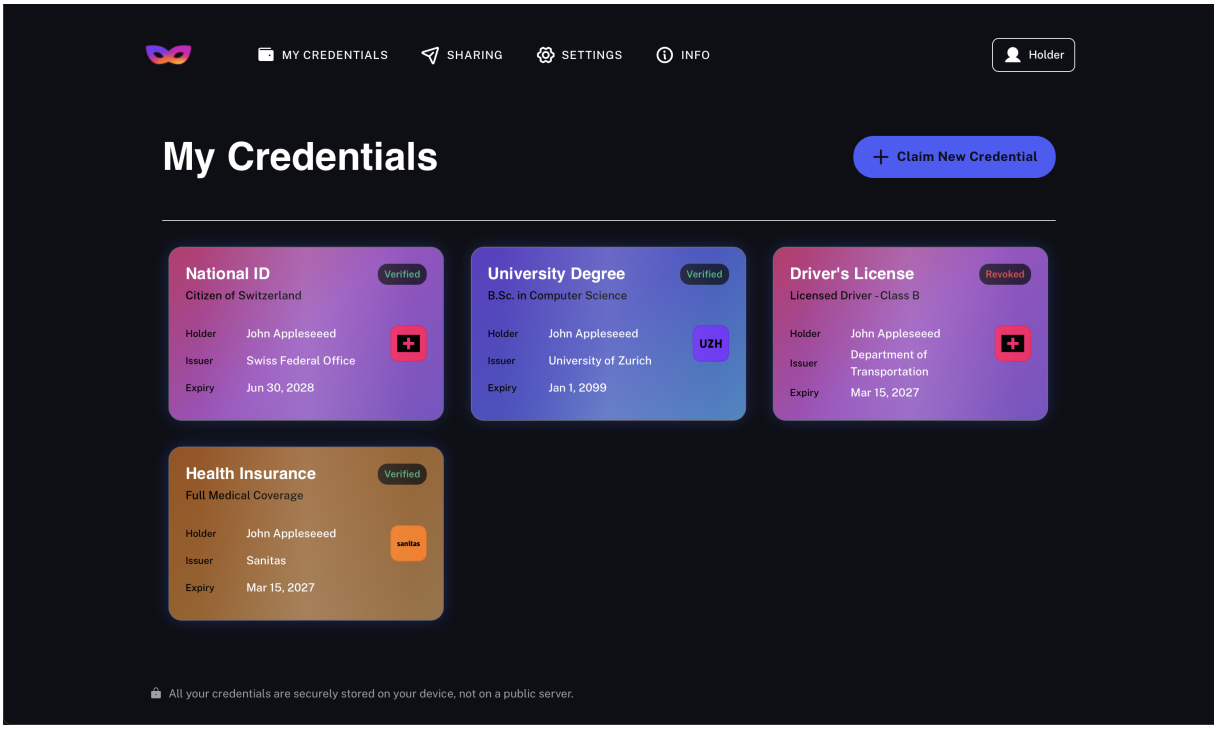


Figure 3.2: Overview of all user credentials with clear status indicators and a button to claim new credentials.

3.2.3 Credential Detail View

When a user clicks on a credential on the 'My Credentials' screen, they are taken to the credential detail view (Figure 3.3). This view is designed to reveal more information progressively, implementing the progressive disclosure pattern (7.c). It prevents the user from being overwhelmed. It is worth mentioning that, this is the first point where DID addresses are shown. Displaying them earlier, such as in the credential overview, would likely confuse users since it is an unfamiliar technical term (1.a). Placing them here respects the user's learning curve. The page is divided into three clearly defined sections: Credential Information, Credential Data, and Verification History. This segregation of content reduces cognitive load, and helps users mentally organize different types of information (2.c, 7.b). A verified icon is displayed next to each DID address throughout the hole application, contributing to transparency (6.a). This design pattern is familiar to users from other platforms like Twitter, where a checkmark signals a verified account. The same icon is also used in the status field to indicate that a credential has been successfully verified. These consistent icons help users quickly recognize trust signals across different sections of the application. Users can share or delete the credential with dedicated buttons next to the card. If 'delete' is clicked, a confirmation modal appears (Figure 3.4) to prevent errors (9.b). The design of the modal uses the strong color red and a clear text to communicate the risk.

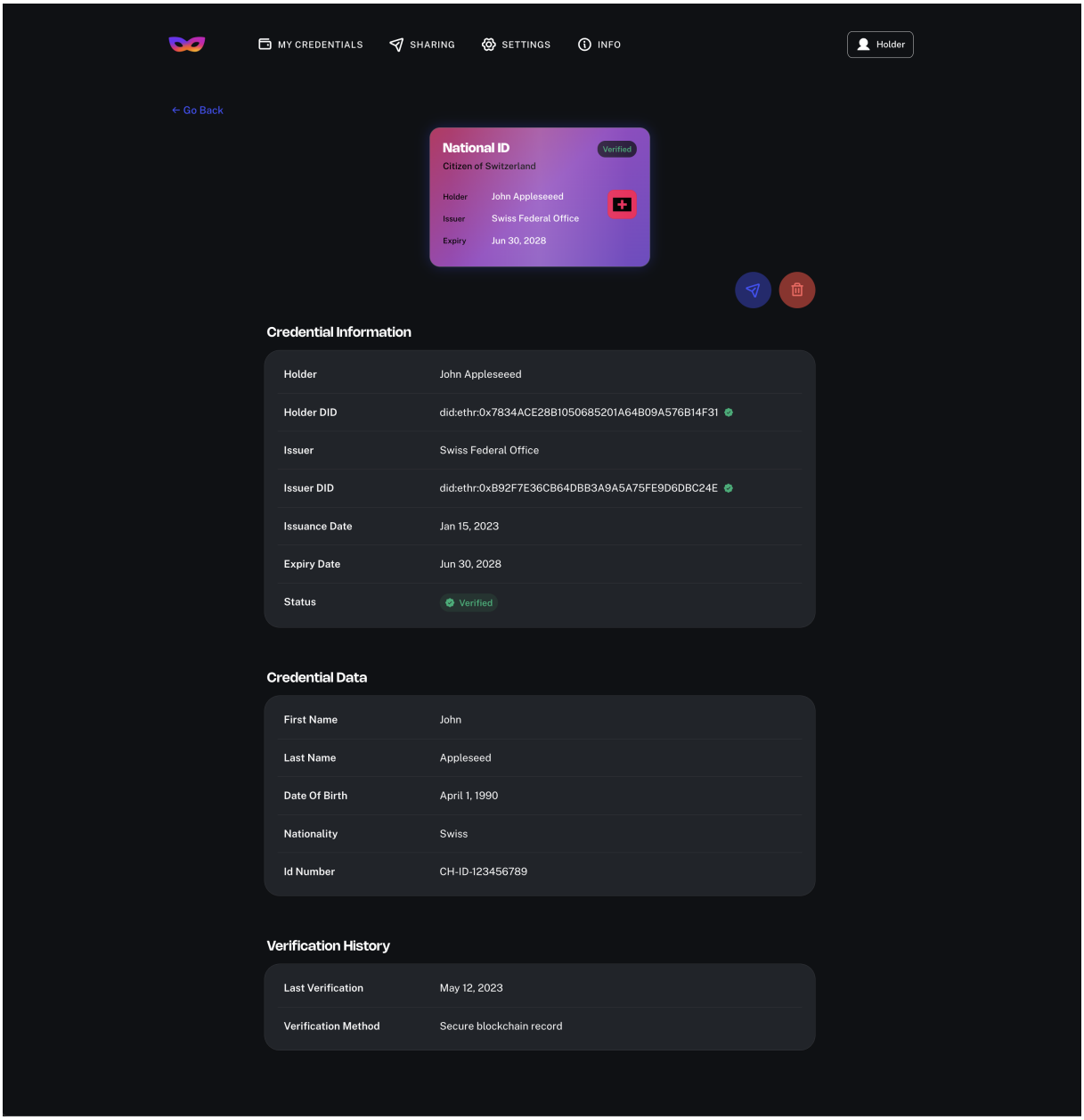


Figure 3.3: Detailed view of a credential, including metadata, credential data, and verification history.

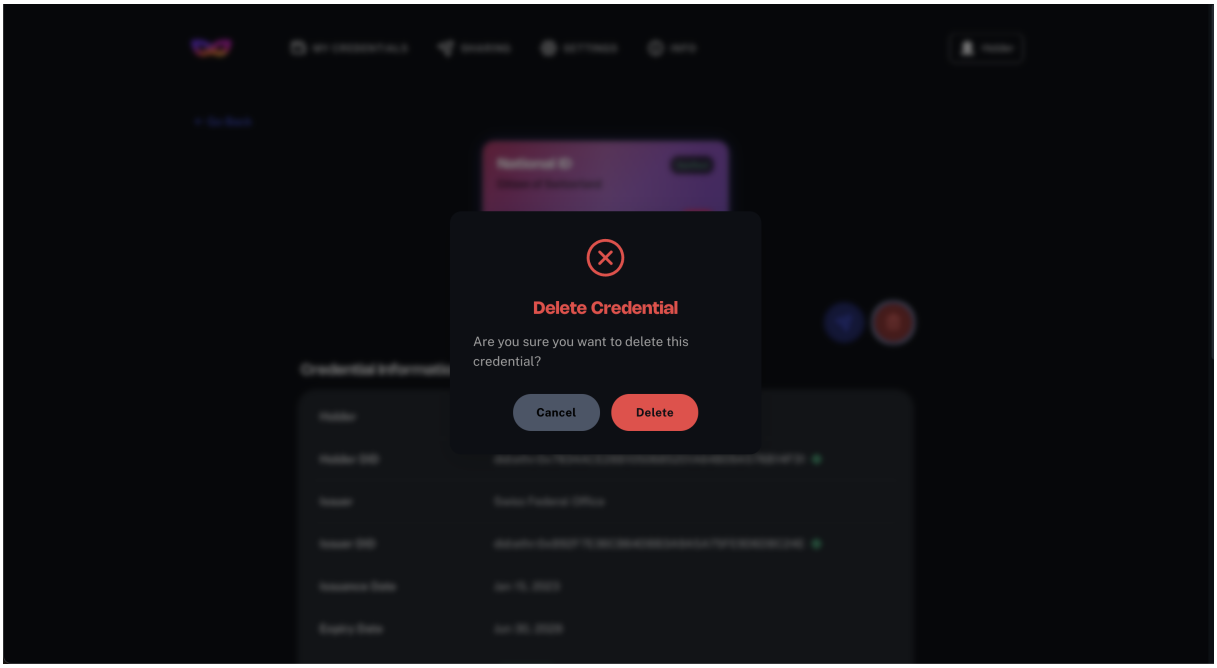


Figure 3.4: Confirmation modal that prevents accidental deletion of a credential.

3.2.4 Requesting a New Credential

When the user clicks the '+ Claim New Credential' button (Figure 3.2), they are taken to the Request New Credential screen (Figure 3.5). This page guides users through the process of requesting a credential from an issuer. There is a short instruction placed directly below the page title that reads: 'Send a request to a trusted issuer to claim your credential'(5.a). The entire screen is designed with visual hierarchy. The actions 'Send Request' and 'Cancel' are placed at the bottom of the screen, encouraging the user to work through the form from top to bottom first. The buttons are initially disabled, guiding the user to complete the necessary tasks first. Signing is only possible once the required information has been entered and sending is possible once the user has signed the request. This structure aims to create a natural interaction flow (2.a) and prevent errors (9.b).

The layout once again separates the form into logical blocks, reducing the cognitive load (7.b). The signing block is placed at the bottom, as in a legal contract that the user signs (2.e). A gray info line at the top of the container contains a brief instruction: 'Enter the issuer's DID (their unique digital identifier). If you don't have it, contact the issuer directly'. The Issuer DID field includes an inline hint, 'DIDs are verified automatically for your security', which prepares the user for an automated validation. As can be seen in Figure 3.6, once the user enters a DID, the system automatically verifies the DID address and provides feedback. There are three possible outcomes, each clearly communicated through color and icon: Green with a checkmark icon: Indicates the DID belongs to a trusted issuer. There is the success message: 'This DID belongs to a trusted issuer.' as can be seen in Figure 3.6 Orange with a warning icon: Signals the DID is unknown, using the message 'This DID is not recognized.' As can be seen in the next Figure 3.8 Red with an X icon: Signals the DID is suspicious, with a warning such as 'This DID is

flagged as suspicious.' This feedback helps users quickly assess the validity of the issuer, improving transparency (6.a) and trust. Another key interaction in this page is the digital signature process. There is a simple and clear button labeled 'Sign Document' along with a sign icon that aims to not overwhelm the user with technical terms (1.a), but still strengthen the users' mental model. Once signed, the signature is displayed with full metadata (name, time, and hash) for transparency (6.a) (Figure 3.6). After signing and sending the request, users receive a positive feedback modal (Figure 3.7) with a success message: 'Your credential request has been sent to the issuer. You will be notified once it's processed.' This sets expectations for what happens next, an essential part of good UX (2.d).

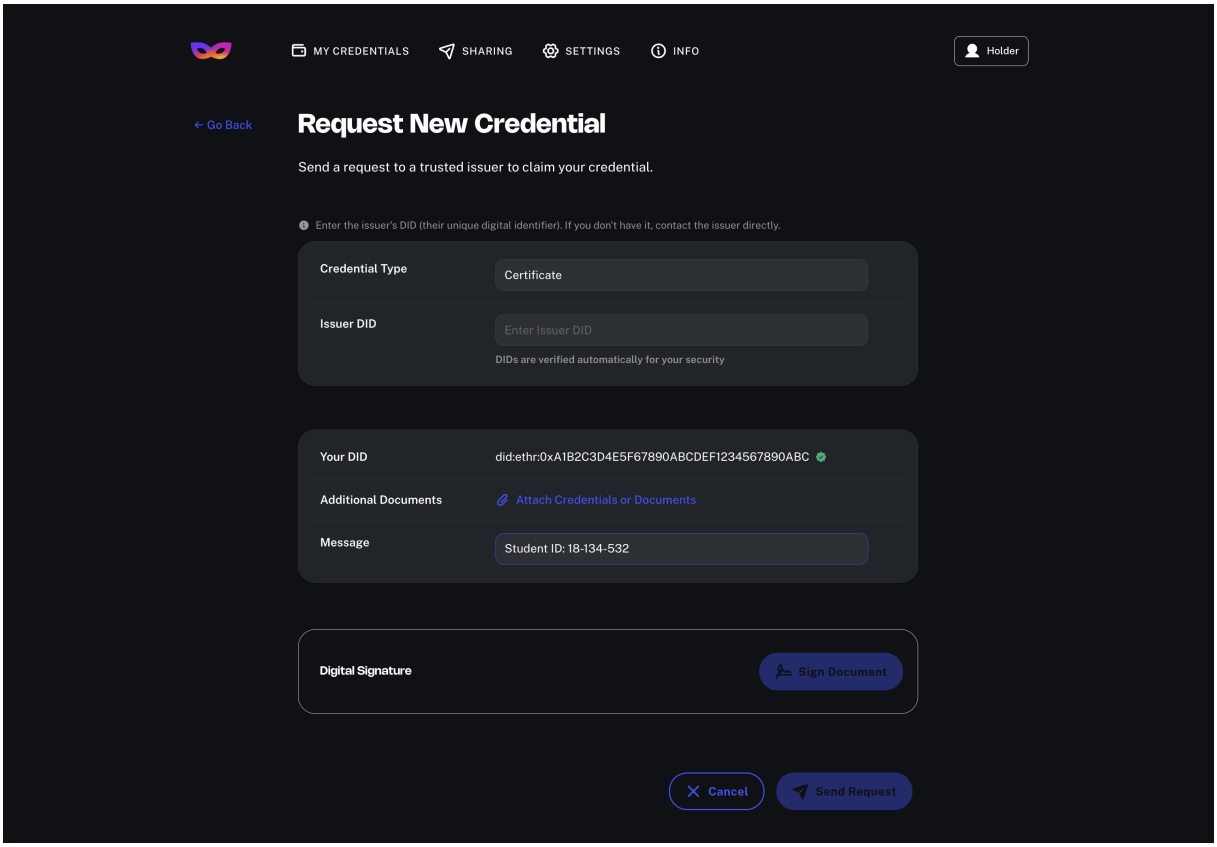


Figure 3.5: Initial credential request. The DID address has not been entered and verified and the request has not been signed yet.

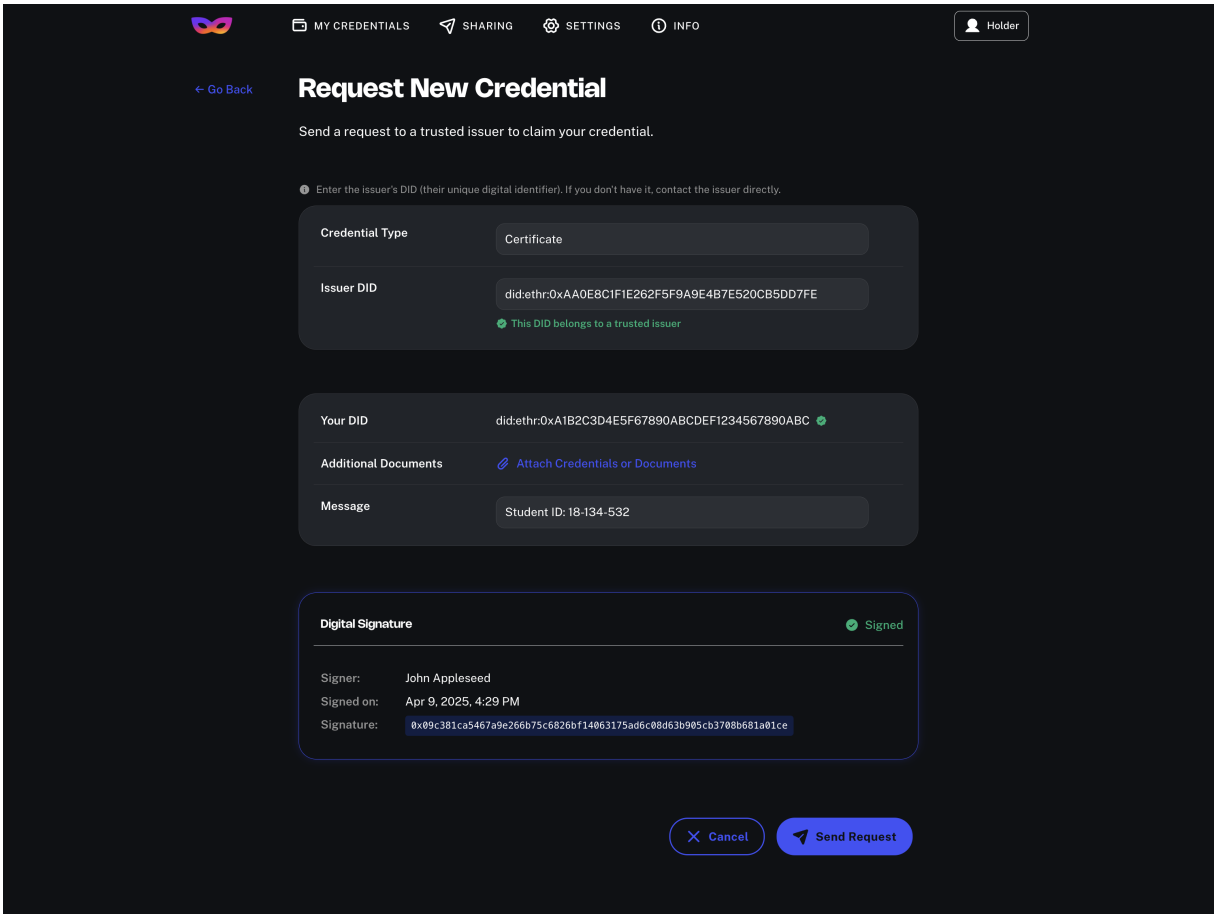


Figure 3.6: A fully completed credential request, including a digital signature and all required fields. The green system message shows that the DID has been successfully verified.

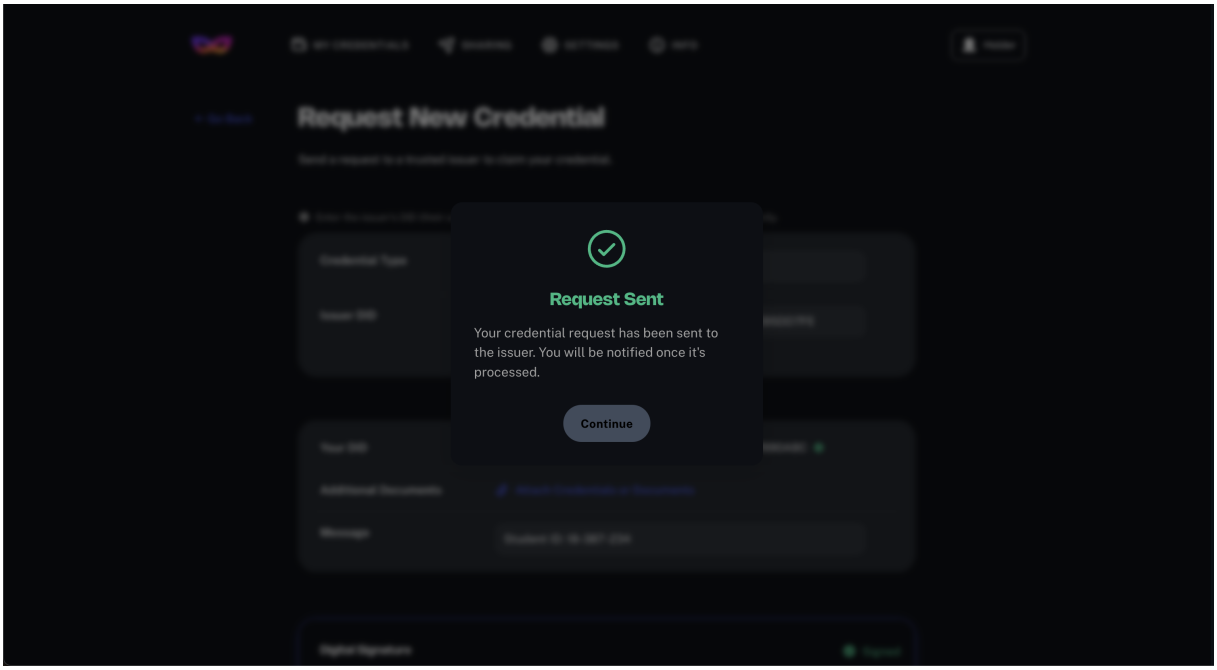


Figure 3.7: Feedback modal confirming that the credential request has been sent successfully.

3.2.5 Sharing Credentials

The sharing page is accessible through the navigation bar or the credential detail page. The goal is that the user presents a credential to a verifier. On the sharing screen (Figure 3.8), the selected credential card remains visible at the top of the page. This ensures that the user always knows which credential they are about to send, helping to prevent mistakenly sending the wrong credential (9.b). The credential card alongside an arrow and verifier illustration visually reinforces that the credential will be transmitted to another party. This visual cue helps to make the abstract sharing process more understandable (3.b, 2.e). At the top of each container, short info texts provide guidance. For example, the verifier information section includes a hint to contact the verifier if their DID is not known (5.a). In the credential section, a message explains that the listed fields are required and will always be shared. This clarification supports transparency (6.a). The data to be shared are divided into two categories: required and optional. The required fields include credential metadata such as the credential type, issuer name, DID addresses, and issuance dates. These are always included in the share and are marked as required. This clear distinction prevents confusion and supports mental clarity about the elements the user has control over. In the section labeled 'Share Credential Data' (Figure 3.9), users can select exactly which optional attributes they want to share by clicking on them. Selected items are highlighted with a check icon and color change, while unselected ones remain colorless. This is a common and intuitive interaction pattern (2.a).

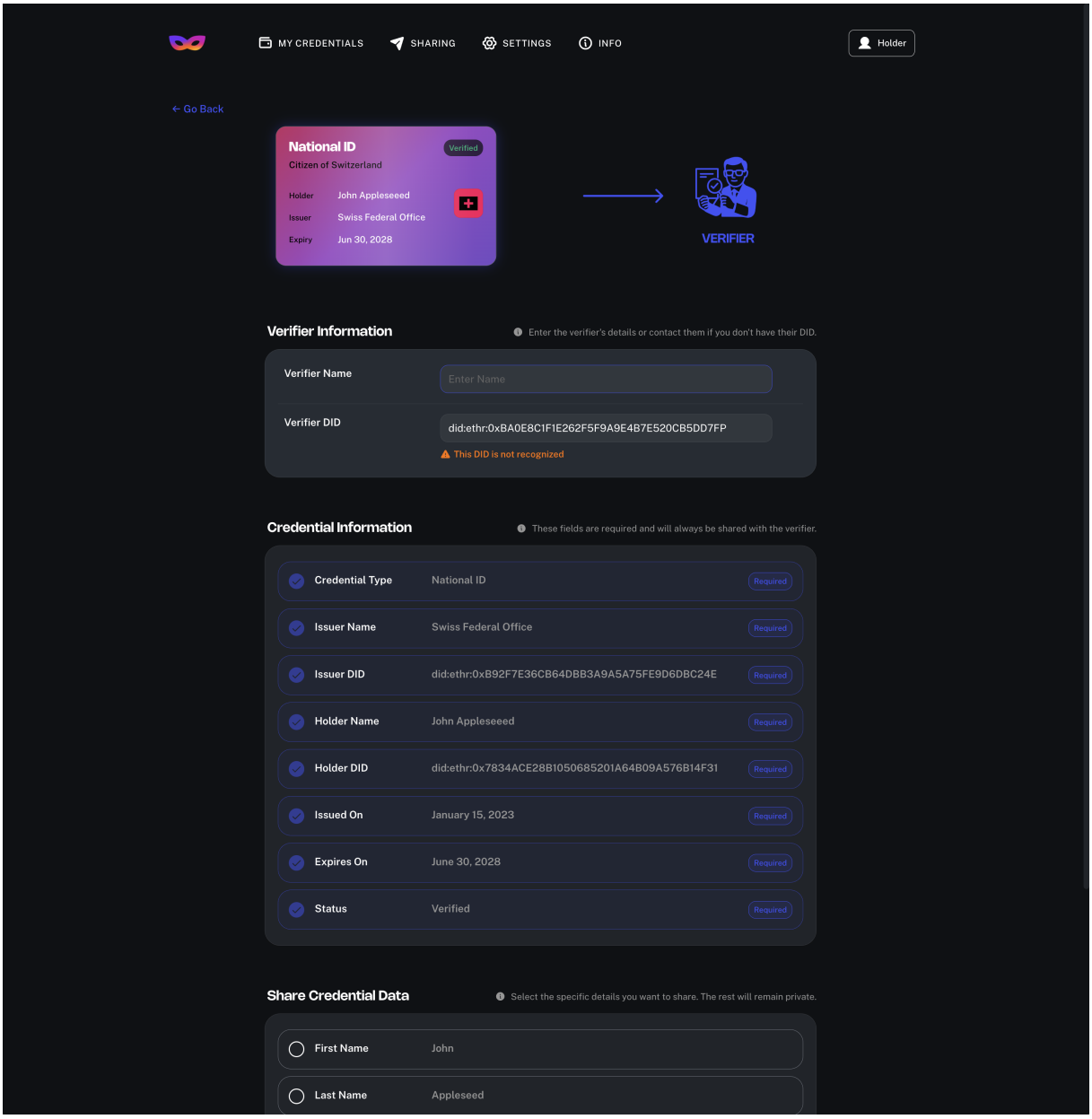


Figure 3.8: Interface for sharing a credential with a verifier, including required fields and a warning of a not recognized DID address. There are instructive info message on top of each container.

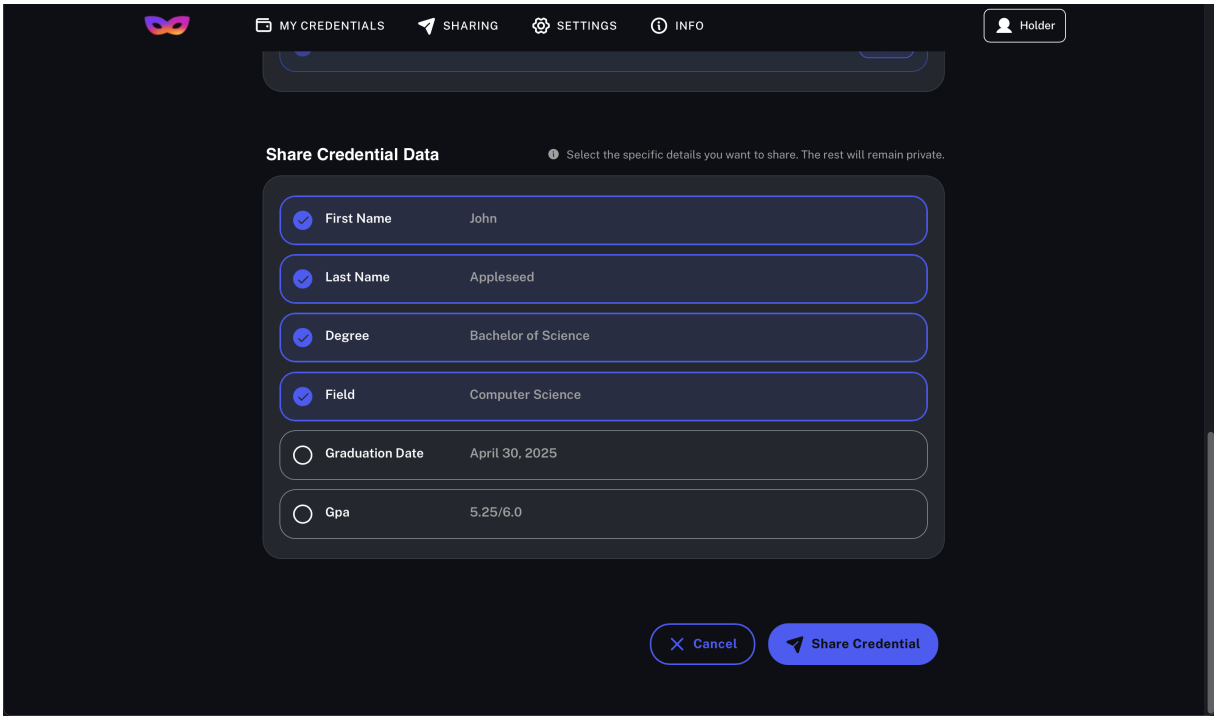


Figure 3.9: Selective disclosure interface allowing users to choose specific data fields for sharing.

3.2.6 Settings

The settings screen (Figure 3.10) gives users control over their identity, security, preferences, and access to useful resources. At the top, the user's decentralized ID and role are shown clearly. Security features include biometric login, PIN protection, backup, and recovery. There is the option to reveal the private key, which remains hidden by default for safety and simplicity reasons (1.a). The section also shows the most recent backup date, adding transparency (6.a).

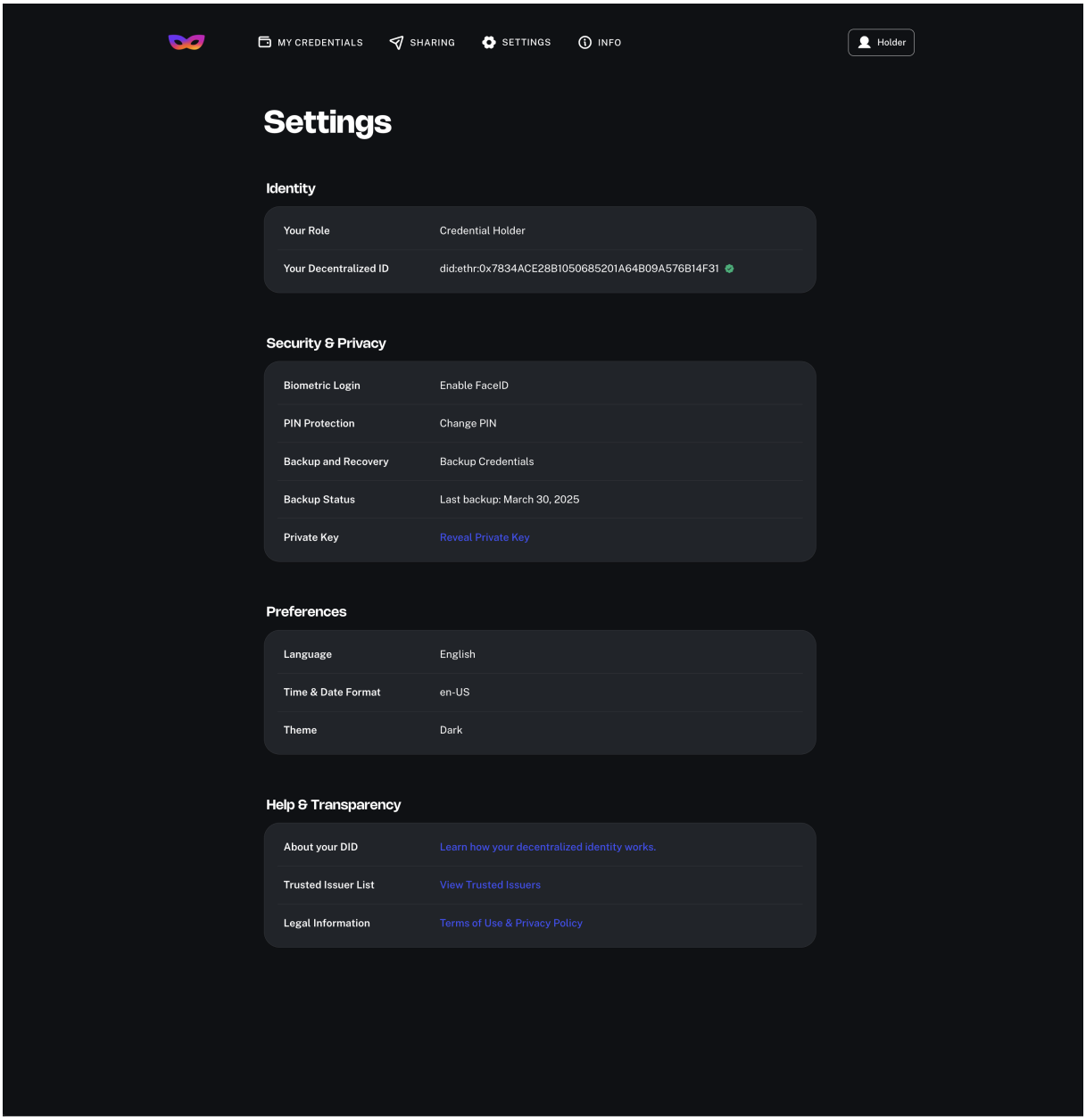


Figure 3.10: Settings page with identity info, security and privacy options, user preferences, and help links.

3.2.7 Info Page

The Info page (Figures 3.11 3.12) is designed to help users understand the core concepts of decentralized identity with the help of simple and accessible language. It introduces the Triangle of Trust, with the three key roles: issuer, holder, and verifier. At the top of the page, a diagram illustrates how credentials are issued, stored, and verified with the help of a blockchain-based registry. Arrows are used to represent the direction of interaction, making the process easier to follow. The visual structure is based on a concept from the a report on DI [49], and helps users to grasp the relationships between roles

without needing to understand the underlying technology. Below the diagram, each role is explained in more detail, making the concept of DI easier to understand. Technical terms are intentionally avoided to keep the explanations simple and accessible to all users (1.a). It also highlights key benefits such as selective disclosure, trust without direct relationships, and protection against data breaches.

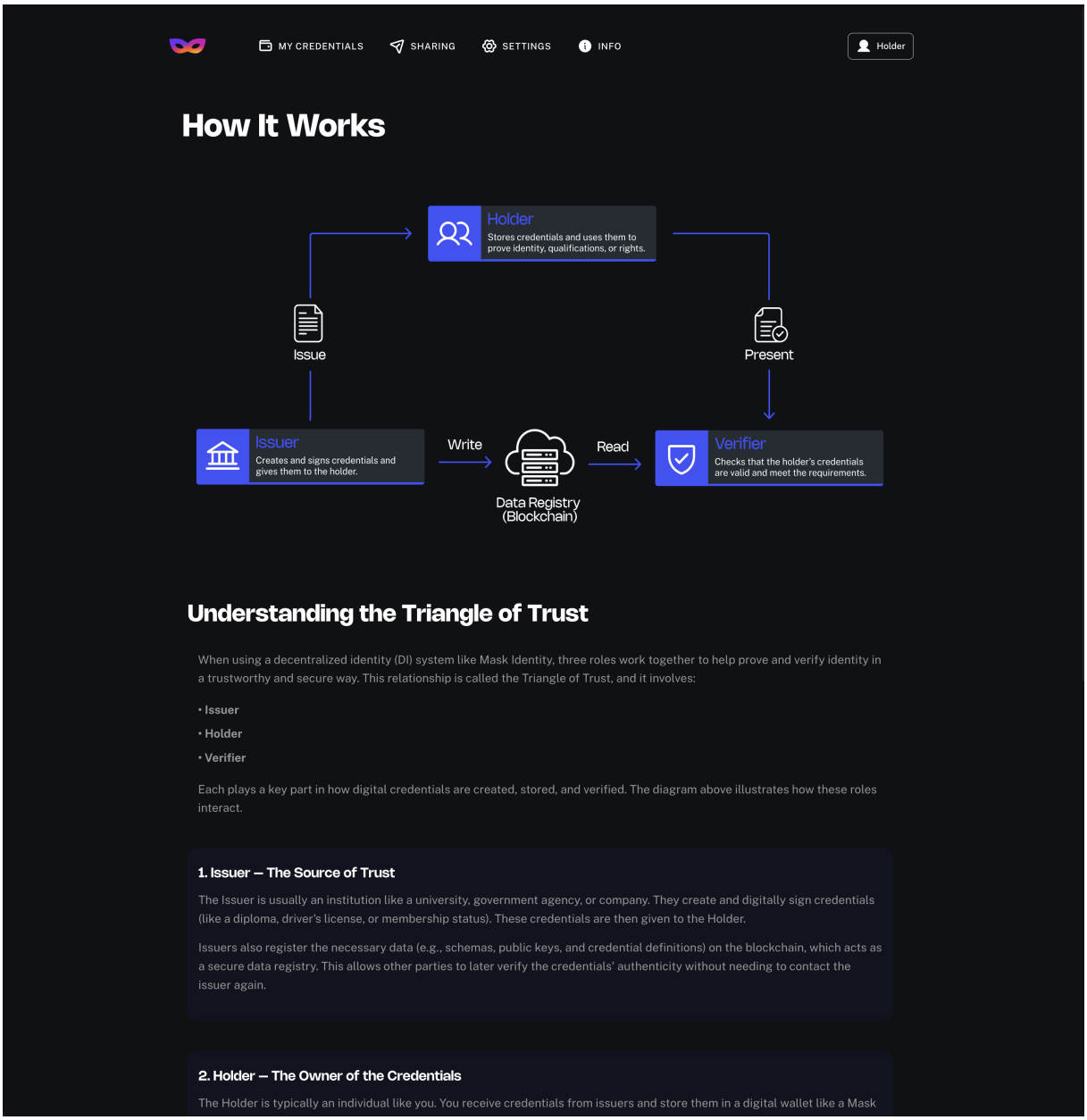


Figure 3.11: Diagram of the Triangle of Trust that illustrates the relationship between issuer, holder, and verifier.

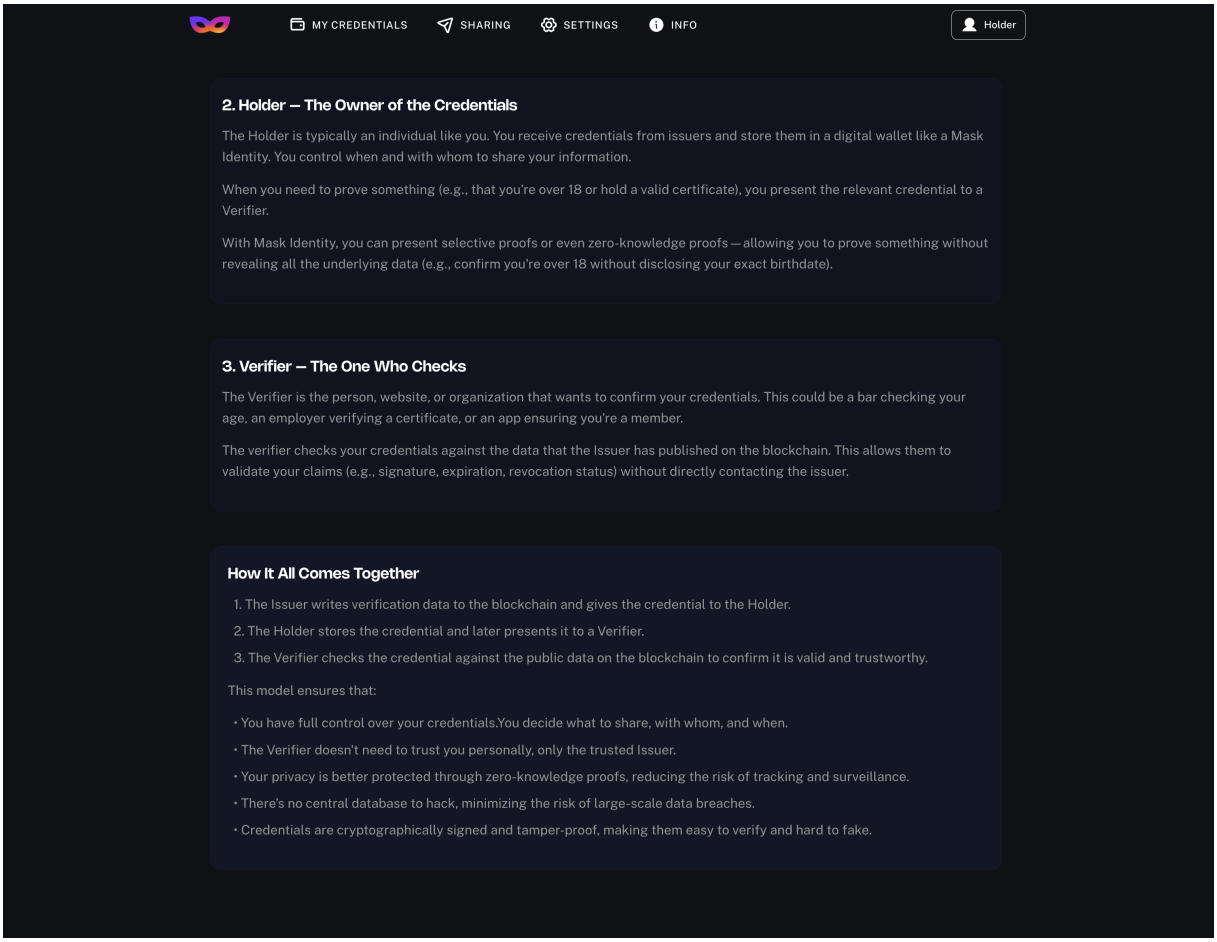


Figure 3.12: Explanation of the three user roles and a summary of the benefits provided by decentralized identity.

3.3 Issuer Landing Page

The issuer landing page (Figure 3.13) is the starting point for users in the issuer role. In the upper right corner, a role-switching button allows a seamless transition between the holder, verifier, and issuer. The screen introduces the role with a short welcome message and an illustration of an institutional building, and a hand that hands over a credential document. This visual representation reinforces the idea that the issuer has the authority to issue credentials. To the right of the illustration, the key functions are listed with matching icons, such as issuing and managing credentials, using the same layout and style as on the holder's welcome screen.

A call-to-action button leads the user to the issuer dashboard.

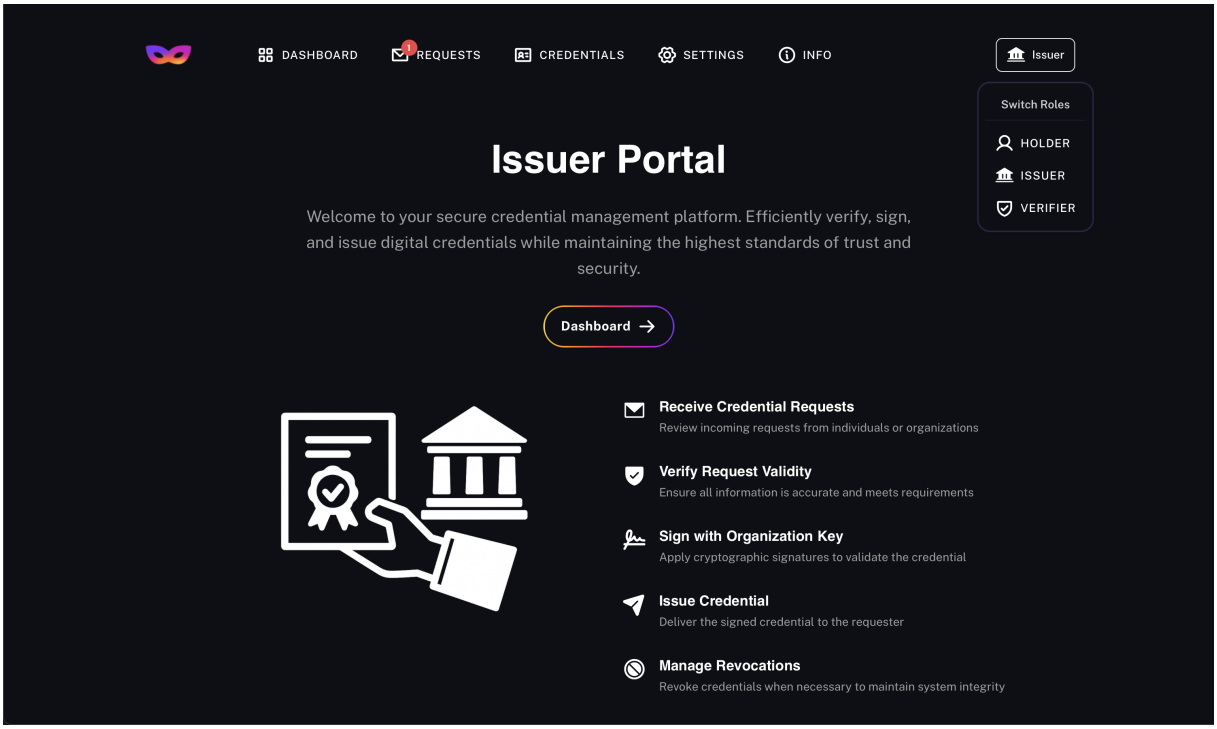


Figure 3.13: Issuer landing page outlining core functions with a call-to-action and and illustrations. In the top right the drop down for switching roles is visible.

3.4 Issuer Dashboard

The issuer dashboard (Figure 3.14) functions as an overview of the key tasks that an issuer needs to perform. It includes three interactive cards, new requests, issued credentials, and statistics. The new request card has a call-to-action button that gives users a clear entry point (2.c). This layout helps reduce cognitive load by showing only the most relevant actions, without overwhelming the user with too many choices (2.c). Each card has an icon that makes its purpose easy to understand immediately, supporting the user's mental model (3.b). The icons also match the rest of the application's design, since they are all from the Phosphor icon library.

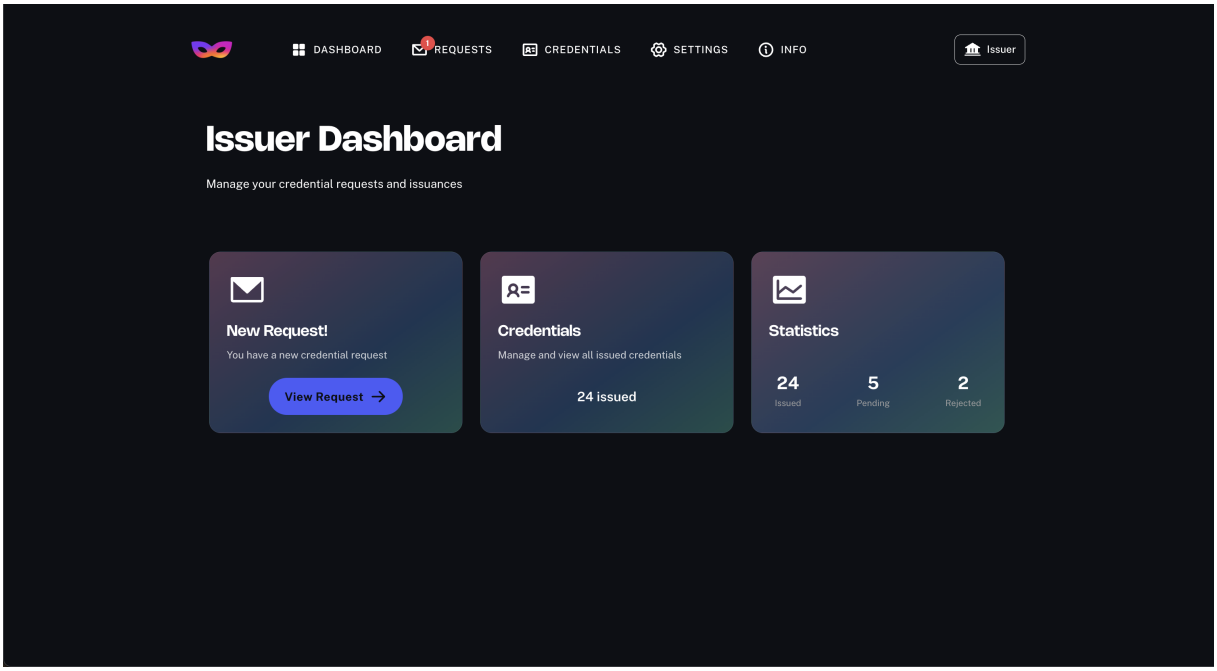


Figure 3.14: Issuer dashboard showing access to new requests, credential management, and statistics.

3.4.1 Reviewing and Verifying a Credential Request

When an issuer receives a new credential request from a holder (someone who wants to obtain a verifiable credential), they are guided to the New Credential Request screen (Figure 3.15). The screen can be accessed either through the dashboard or the dedicated Requests section in the navigation bar. The request section on the navigation bar follows a familiar mental model, as it looks similar to a message inbox with a letter icon and a red badge that signals new activity (2.e). This helps users to immediately understand where to look for incoming requests. At the top of the page, there is a clear heading and a short introduction text that helps instruct the issuer (5.a). Below this, the credential information and credential data are presented in a structured format, just like the for the holder (7.b). The Credential Verification section (Figure 3.16) offers a prominent 'Verify Now' button with a shield icon that conveys the concept of security and trust (3.b). Each element that has to be verified includes a clear title, such as 'Requester Identity' or 'Request Signature' and a short explanatory question (e.g., 'Is the request properly signed by the requester?'). This helps users understand what is being checked, using simple wording without technical language (1.a). On the right side of each item, a small label says 'Blockchain Verification', which informs the users what process is about to take place, contributing to transparency (6.a). After clicking the button, the system checks the relevant data and updates the interface (Figure 3.17). Verified elements are highlighted with green coloring, check marks, and status tags like 'Verified,' providing clear visual feedback that reinforces trust (3.b, 2.d, 6.a,). In addition, a digital signature box confirms that the request was properly signed and shows details such as the signer, timestamp, and signature hash. Together, these elements support a transparent and

structured credential issuance process. Visual cues, intuitive language, and real-time feedback help issuers review and approve requests with confidence.

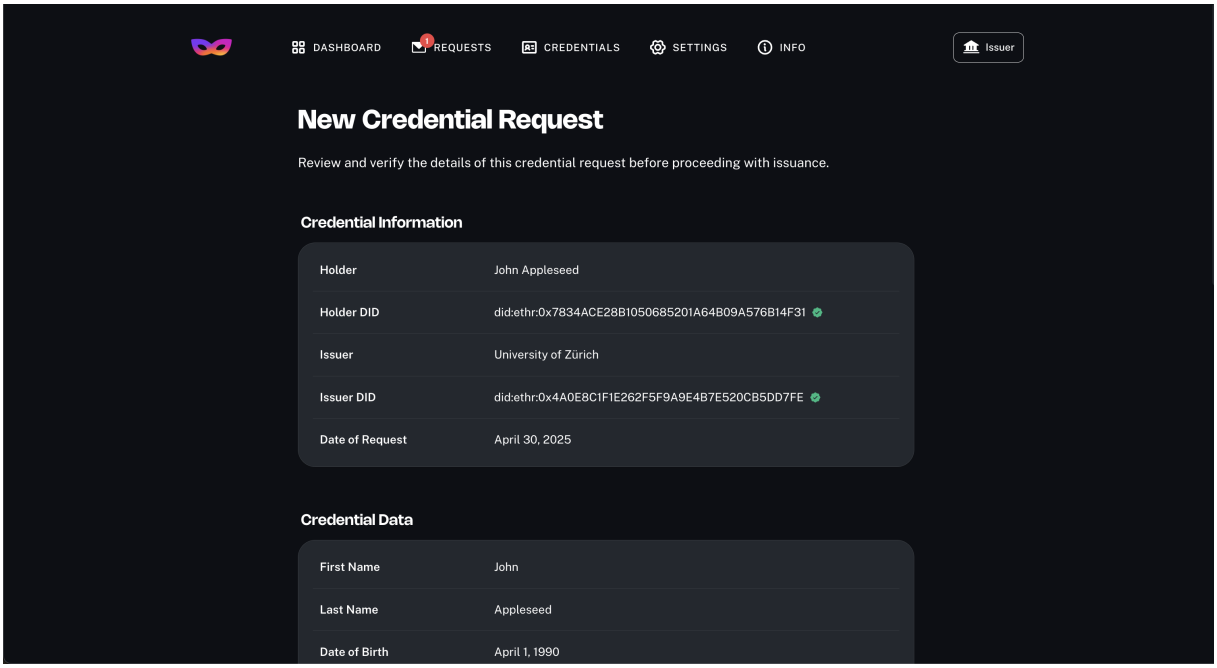


Figure 3.15: Detailed overview of a new credential request submitted by a holder.

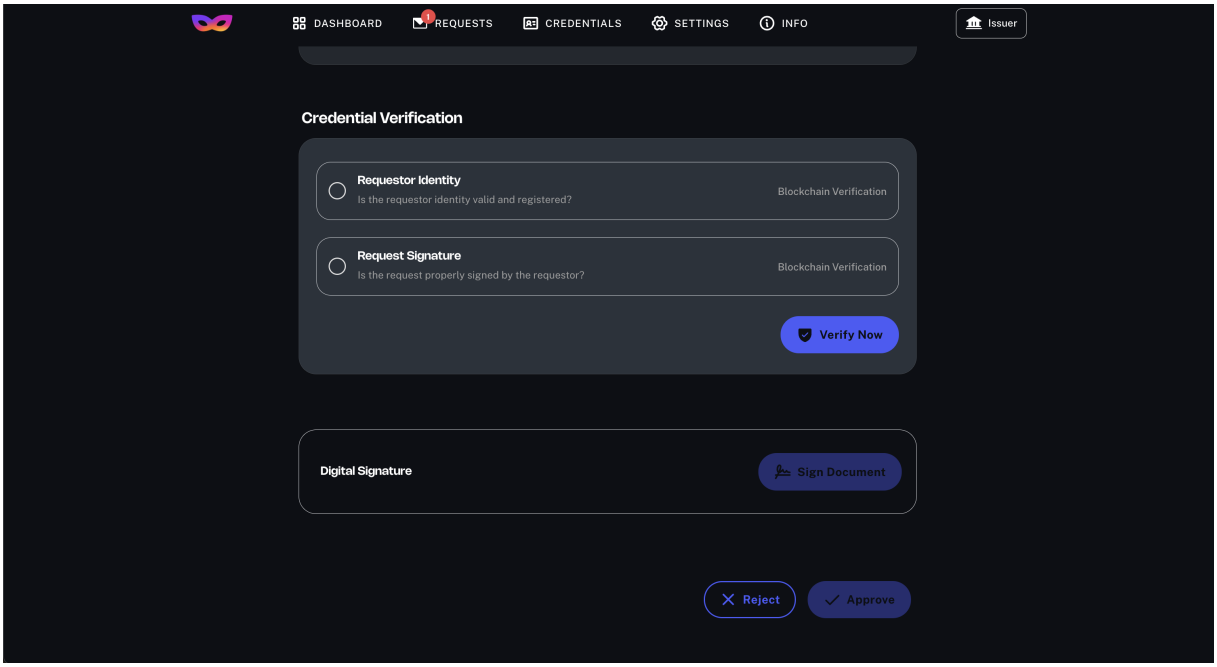


Figure 3.16: Credential request before signing, showing unverified checks.

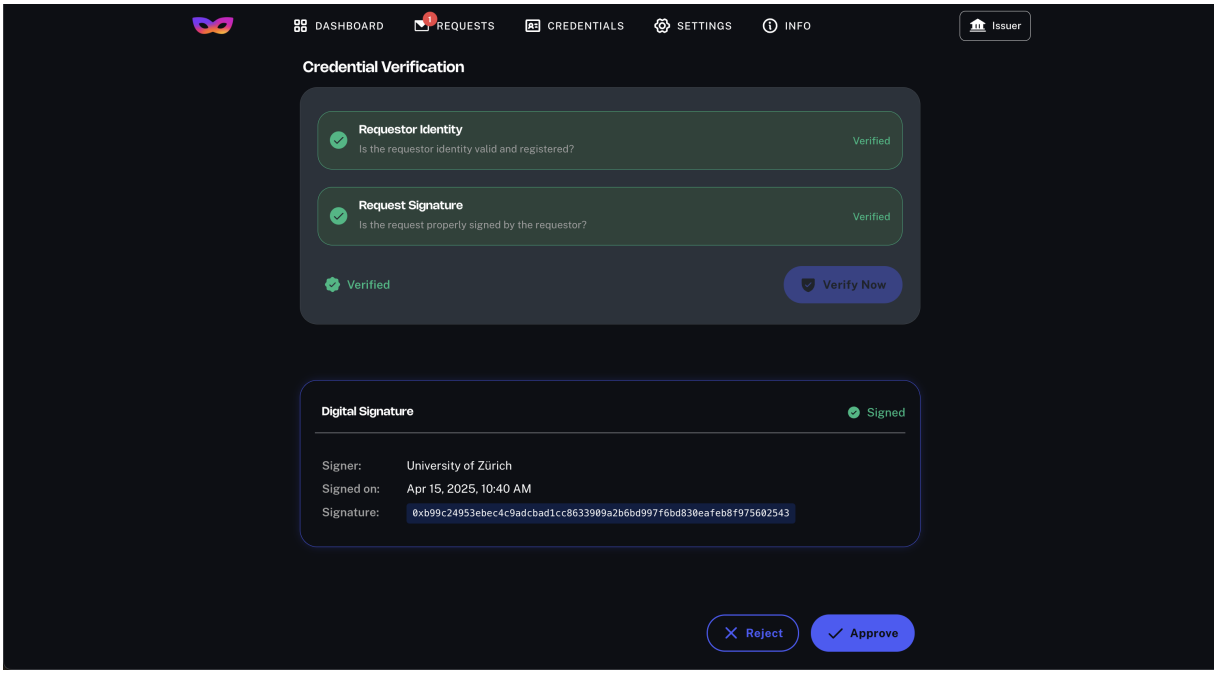


Figure 3.17: Verification screen with all checks passed and a digital signature included.

3.4.2 Issued Credentials and Revocation

The issuer can view all previously issued credentials by clicking on the credentials section in the navigation bar (Figure 3.18). This screen shows an overview of each credential, including who received it and its current verification status. The consistent layout and use of visual status tags like 'Verified' or 'Revoked' make it easy to scan and understand the state of each item. When the issuer clicks on a credential card, they are taken to a detail view (Figure 3.19). Here, they can see all the information about the credential. A red revoke button is clearly visible next to the credential card. If the issuer chooses to revoke, a confirmation modal appears (Figure 3.20) asking if they're sure about the action. This follows the prevent user mistakes pattern (9.b), ensuring that credentials are not accidentally revoked.

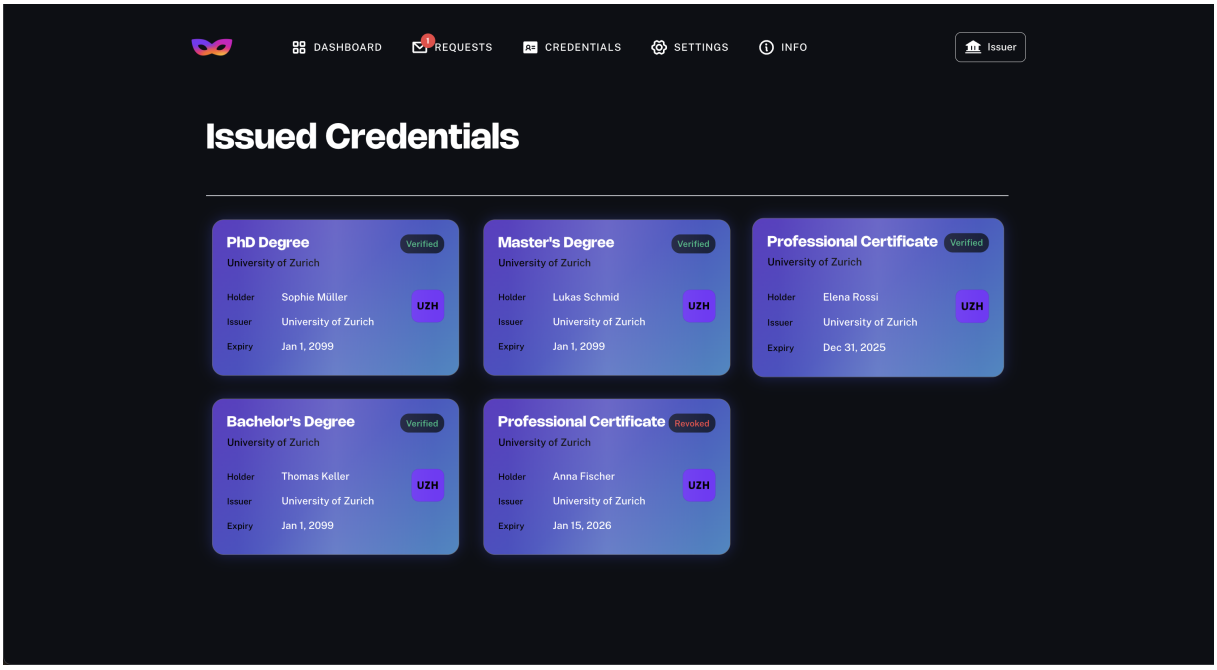


Figure 3.18: Overview of all issued credentials displayed in a card-based layout.

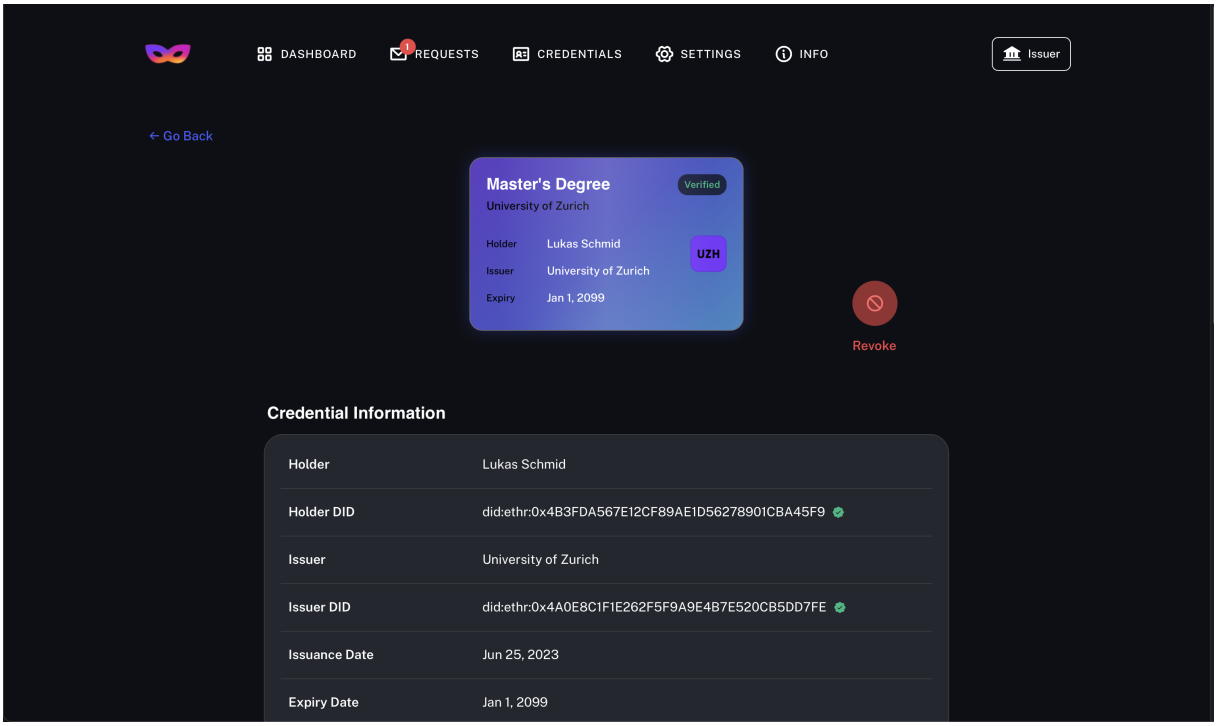


Figure 3.19: Credential detail view showing all metadata and a revoke option.

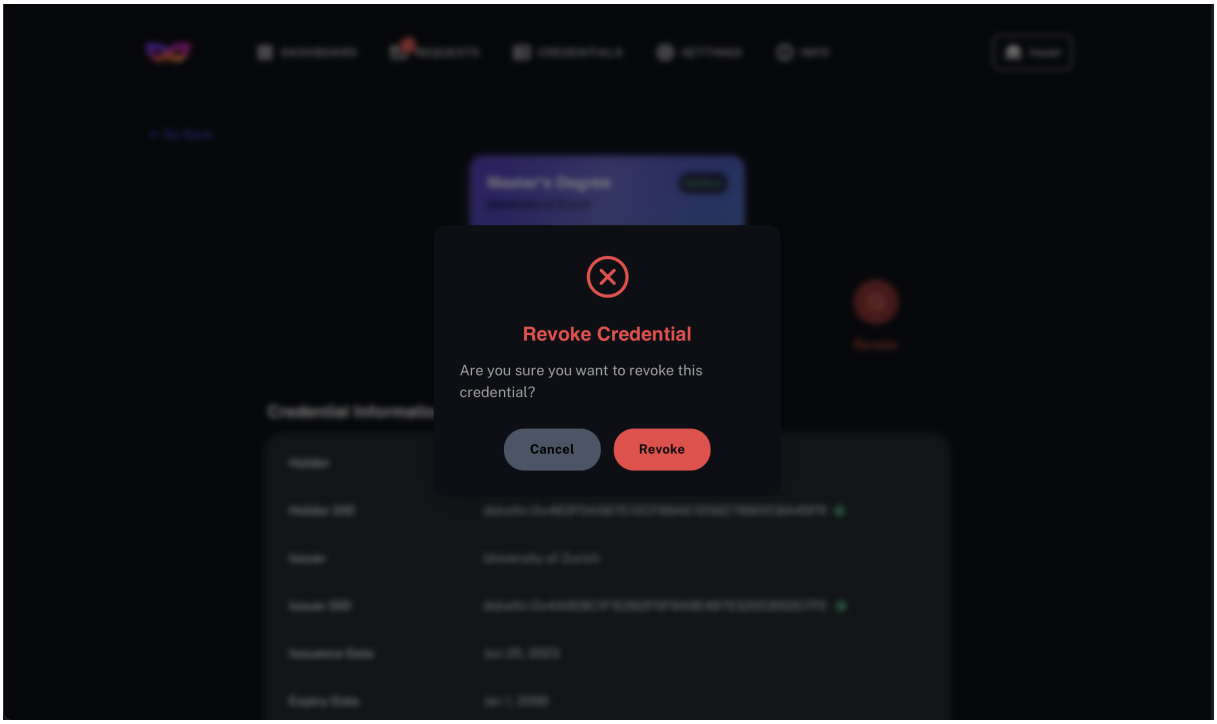


Figure 3.20: Confirmation modal asking the issuer to confirm revocation of a credential.

3.4.3 Verifier Portal

The verifier landing page (Figure 3.21) introduces the verifier role just like for the holder and issuer. The verifier dashboard (Figure 3.21) gives a quick overview of the most important tasks, such as reviewing incoming requests. The two cards follow the same design pattern as the issuer dashboard, maintaining consistency across the application. The 'View Request' button and envelope icon clearly indicate new activity, helping users stay oriented and reducing the chance of missing important tasks.

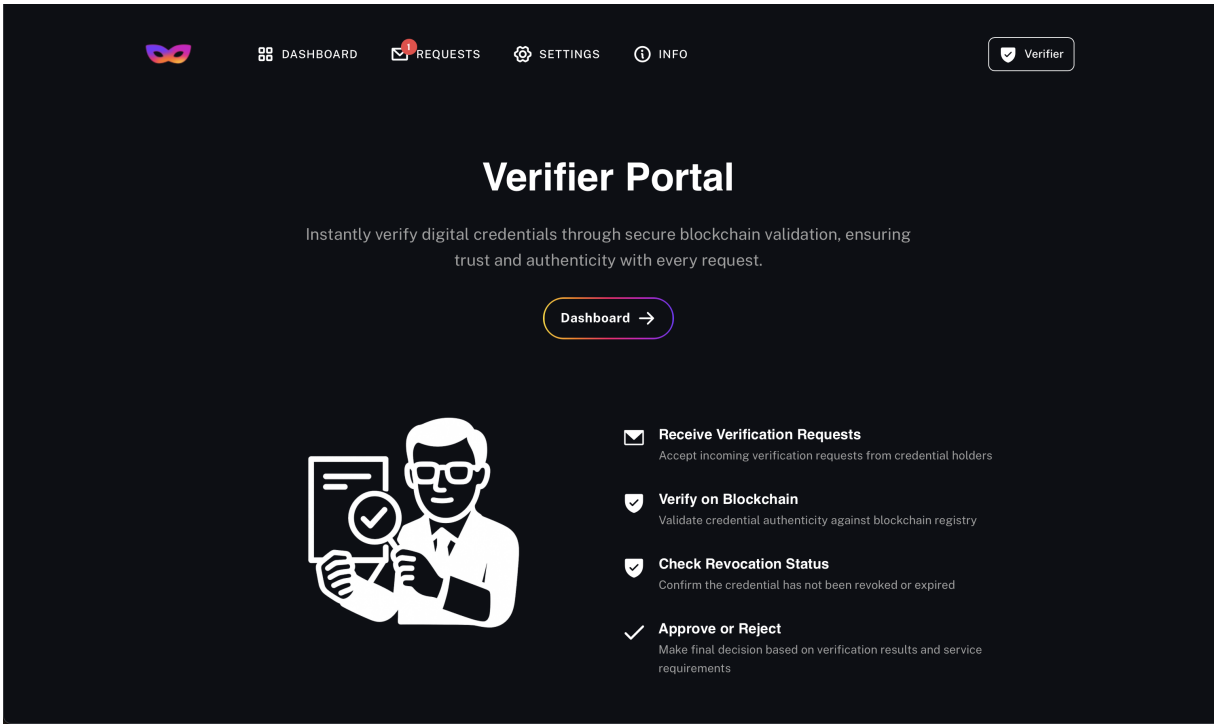


Figure 3.21: Verifier landing page introducing the main functions of the portal, including request handling and blockchain-based verification.

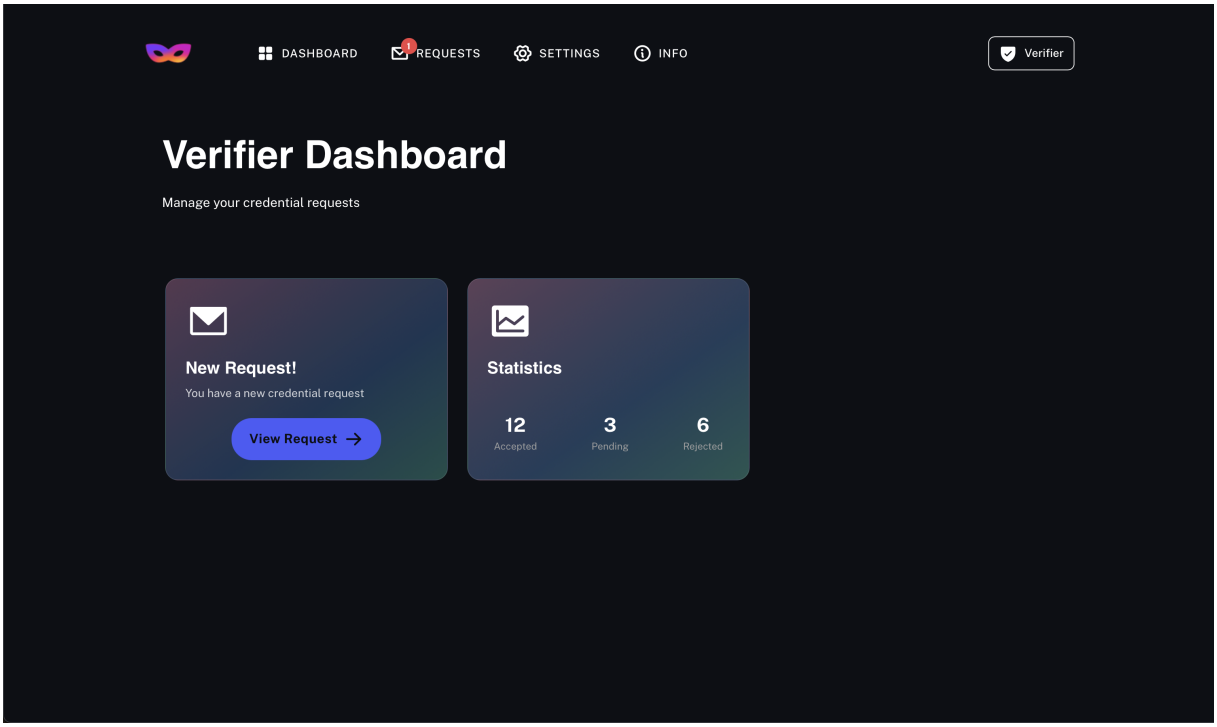


Figure 3.22: Verifier dashboard with quick access to new requests.

3.4.4 Verification of Incoming Requests

The interface visible in Figure 3.23 belongs to the request section, where incoming verification requests are handled. The credential data itself are shown at the top of the page, but is not visible in the current screenshots. Figure 3.23 shows the ongoing verification process, where several checks have failed. These failed checks are clearly marked with red color and icons, making the outcome easy to understand (3.b). If the verification fails, the user can choose to reject the request. Then a feedback modal appears, informing the user that the credential was rejected and that the holder will be notified (2.d) (Figure 3.24).

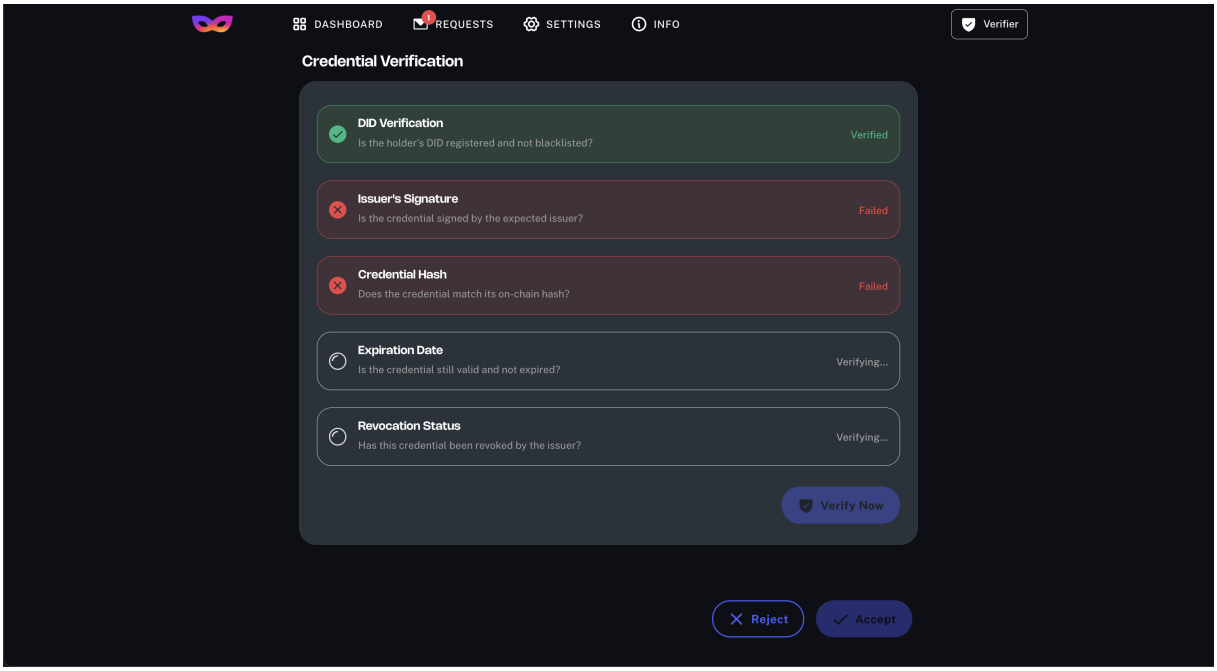


Figure 3.23: Credential verification interface showing the ongoing process of blockchain validation.

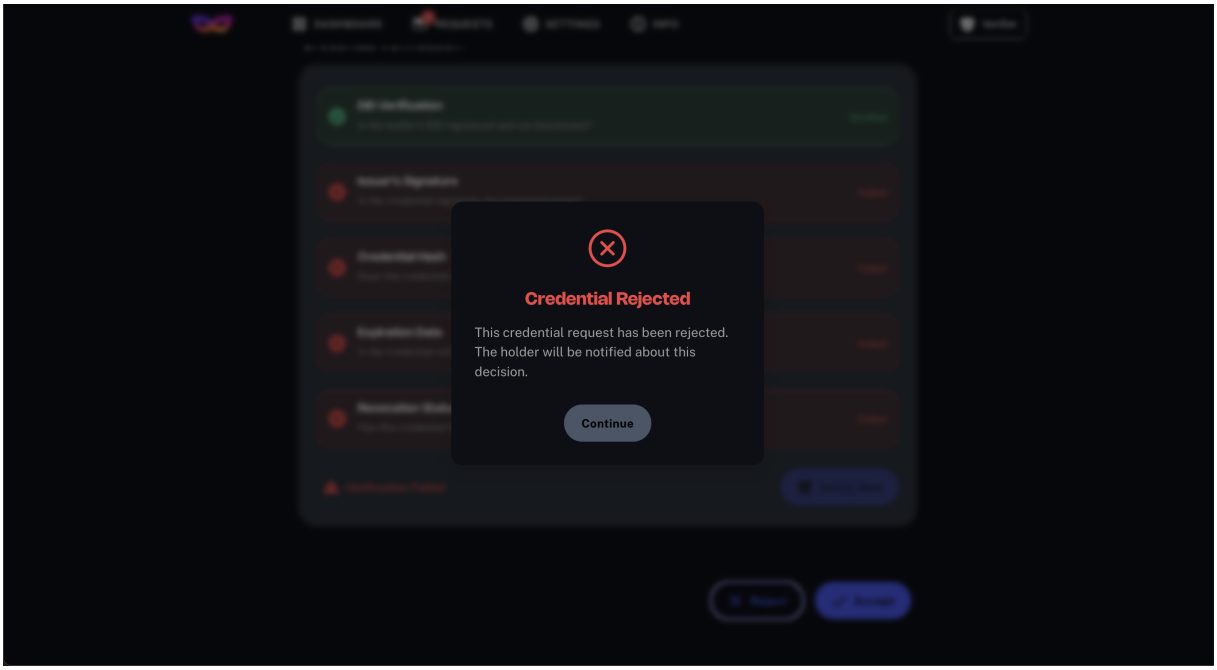


Figure 3.24: Feedback modal displayed after a credential has been rejected by the verifier.

Chapter 4

Implementation

4.1 Technology Stack

The development of the application prototype relies on the following technologies and frameworks:

- **Vue 3.5.13:** Core framework used for building the user interface. [51]
- **Vue Router 4.5.0:** Handles navigation between different views in the application. [51]
- **Tailwind CSS 4.0.14:** Utility-first CSS framework used for styling and layout. [46]
- **Phosphor Icons:** Icon library used for consistent and modern visual elements. [43]
- **Vite 6.2.0:** Build tool and development server for fast and efficient project setup. [50]

The main technology used was the JavaScript framework Vue.js 3, which is known for its simplicity, flexibility, and strong performance. A key feature of Vue 3 is the Composition API, which allows related logic to be grouped together more clearly within components. This approach improves code readability and maintainability, especially in larger projects. Styling is handled using Tailwind CSS, which provides utility classes for basic layout and visual design choices. It is not heavily used throughout the project, but it is still useful for quickly applying inline style and keeping the styles consistent.

4.2 Application Structure

The application follows a organized, component-based architecture with great reusability, maintainability, and separation of concerns. Each part of the interface is broken down into smaller reusable parts that appear throughout the application. This makes it easier to manage and extend.

4.2.1 Entry Point

The application is initialized by creating a Vue App instance with App.vue as the root component. Global styles are imported, and Vue Router is configured for client-side navigation. The App is then mounted to the HTML element with the ID 'app'.

4.2.2 Routes

The application uses Vue Router to manage navigation between the three main user roles: holder, issuer, and verifier. Each role has its own base route (/holder, /issuer, /verifier) and a few child pages (Figure 4.1). To keep the code organized, these child routes are defined separately and imported into the main route configuration file (index.js). The roots path (/) is redirected to the holder, since its the core part of the application. The routes for each role have a shared layout component (DefaultLayout.vue), which will be explained in more detail in the following section.

```
1  const routes = [
2    { path: "/", redirect: "/holder" },
3   {
4      path: "/holder",
5      component: () => import("../layouts/DefaultLayout.vue"),
6      children: holderRoutes,
7    },
8    {
9      path: "/issuer",
10     component: () => import("../layouts/DefaultLayout.vue"),
11     children: issuerRoutes,
12   },
13   {
14     path: "/verifier",
15     component: () => import("../layouts/DefaultLayout.vue"),
16     children: verifierRoutes,
17   },
18 ];
```

Figure 4.1: Route Configuration with Child Routs (index.js)

The next part of the code (Figure 4.2) defines which pages belong to which route. For example, the holderRoutes file specifies which components should be shown when a user visits pages like /holder, /holder/credentials, or /holder/settings. These pages are then loaded inside the shared layout defined for that route.

```
export default [
  { path: "", component: () => import("../pages/holder/index.vue") },
  { path: "credentials", component: () => import("../pages/holder/
      credentials.vue") },
  { path: "settings", component: () => import("../pages/holder/settings.
      vue") },
];
```

Figure 4.2: Holder Routes (holder.routs.js)

With this structure, navigating to '/holder/credentials' will display the credentials page for the holder role inside the shared application layout.

4.2.3 Layout

To provide a consistent interface for all user roles, the application uses a shared layout component called DefaultLayout.vue (Figure 4.3). This ensures that the layout is consistent across the hole application. The navigation bar, for instance, is always at the same position, and the spaces between the main content and the left and right edges are always the same. The key element in this setup is the <router-view />, which acts as a placeholder for a specific page component. This allows role-specific pages to be injected into the layout, while keeping the outer structure the same.

The layout also dynamically adapts to the current route by determining the active role from the URL path and passing it to the navigation bar component. This way, the navigation bar can have different sections in it, depending on the current user role.

```
<template>
  <NavBar :current-role="currentRole" />
  <main class="main-content">
    <router-view />
  </main>
</template>
```

Figure 4.3: Default Application Layout (DefaultLayout.vue)

4.2.4 Styling and Design System

The application uses Tailwind CSS together with custom styles to create a clean and consistent design. Most of the styling variables are defined in the styles.css file. It includes two fonts, Nohemi and Public Sans, and sets styles for headings and text, such as font size, weight, spacing, and color. The overall design uses a dark background with light text and includes specific colors for things like buttons, warnings, and status messages.

4.2.5 Component Structure

The application is built around a modular, component-based architecture that focuses on reusability and flexibility. Each part of the interface is broken down into specific components, including small elements such as buttons, labels, and data fields. This makes the codebase easier to maintain and extend as the application grows.

Some components follow the software design pattern of inheritance. This means that a parent component provides the basic structure and logic, and a child component builds on it and is more specific. For example, BaseButton is the parent component that defines the core styles and functionality for all buttons. A more specific child component, like IconButton, inherits from BaseButton and extends it by adding an icon. This approach allows for reuse of the shared logic. Communication between parent and child components is handled through Vue's props and emits. Props are especially important for creating different variations of a component.

For example, the credential information is displayed using a nested composition of specialized components (Figure 4.4). A DataContainer component defines the overall section with a title, while individual details are shown using DataField. In this case, the holder's name is passed as a simple text value, and the holder's DID is rendered using a nested DIDAddress component inside the DataField. The DIDAddress includes a small verified icon to indicate that the DID is verified. This setup keeps the structure clean and allows flexible combinations of text, icons, and other visual elements. If the appearance of verified DID addresses was required to change, it can be done directly within the DIDAddress component, and the changes will automatically apply throughout the entire application wherever the component is used.

```
<DataContainer title="Credential Data">
  <DataField label="Holder Name" :value="credential.holder" />
  <DataField label="Holder DID" is-last="true">
    <DIDAddress
      address="did:ethr:0xAA0E8C1F1E262F5F9A9E4B7E520CB5DD7FE"
      icon="verified"
    />
  </DataField>
</DataContainer>
```

Figure 4.4: Displaying Credential Information Using Nested Components

4.2.6 Mock Credential Store

In the file credentialStore.js there is mock data defined that is used across the entire application to simulate digital credentials for both holders and issuers. It contains two separate JSON objects: one for the user's personal credentials (such as a national ID, university degree, driver's license, and health insurance) and another for the credentials issued to others by an organization, in this case the University of Zurich. This mock data

is the main source of content for displaying data and was generated with the assistance of the generative AI model Claude 3.7 Sonnet by Anthropic. Each credential includes metadata such as the type, holder name, issuer, and DID, as well as more specific data (additionalData) that vary from different credential types. (Figure 4.5)

The DataField component can loop through the entries in the credential JSON. This means that if information is changed or new information is added, it will automatically be displayed wherever that credential is shown in the application. This makes the system very flexible and easy to update, which is especially useful when testing different scenarios, for example during a user study.

```
"national-id-1": {
    id: "national-id-1",
    type: "National ID",
    subheading: "Citizen of Switzerland\n",
    verified: true,
    holder: "John Appleseeed",
    holderDid: HOLDER_DID,
    issuer: "Swiss Federal Office",
    issuerDid: "did:ethr:0xB92F7E36CB64DBB3A9A5A75FE9D6DBC24E",
    expiryDate: "June 30, 2028",
    logoUrl: switzerlandLogo,
    colorTheme: "pink",
    issuanceDate: "January 15, 2023",
    additionalData: {
      firstName: "John",
      lastName: "Appleseed",
      dateOfBirth: "April 1, 1990",
      nationality: "Swiss",
      idNumber: "CH-ID-123456789",
    },
    verification: {
      lastVerified: "May 12, 2023",
      method: VERIFICATION_METHOD,
    },
  },
```

Figure 4.5: Example Credential Entry from the Mock Store. The data was generated using Claude 3.7 Sonnet. (credentialStore.js)

4.2.7 Icons and Visual Elements

The application uses the Phosphor Icons library for all icons in the application. This library offers a flexible set of icons that come in multiple visual styles, including thin, light, regular, bold, fill, and duotone. These styles can be used dynamically, depending on the context. For example, in the navigation bar, icons switch to the fill style when their corresponding section is active. This is a standard design convention that helps indicate which view is currently selected. Figure 4.6 shows a navigation section (NavItem) that has an icon that has a weight property that changes to ”fill” if the state isActive is true.

The icons are also customizable in terms of size and color, allowing them to seamlessly integrate into different visual contexts.

```
<component
  v-if="icon"
  :is="icon"
  :size="iconSize"
  :weight="isActive ? 'fill' : iconWeight"
/>
```

Figure 4.6: Dynamic Icon Styling in NavItem Component (NavItem.vue)

In addition to the icons, the application uses static images, such as PNG and SVG files, for issuer logos and illustrations, which are stored in the assets folder. They are used to represent institutions like the University of Zurich or the Swiss government on the credential cards or for illustrations on the landing pages. The illustrations on the landing pages were generated using OpenAI's ChatGPT-4o model to ensure a visually consistent and professional look throughout the prototype 4.7.

Figure 4.7: Illustrations of the three user roles in the prototype: Holder, Issuer, and Verifier. The images were generated with ChatGPT-4o.

4.3 Summary

The implementation demonstrates how modern front-end technologies can be used to build a role-based modular application for decentralized identity. By using Vue.js 3 with reusable components, scoped and global styling, and mock data, the application remains maintainable and scalable while offering a consistent user experience across different user roles. Through the use of a shared layout and dynamic routing, the application architecture supports a clear separation of concerns while maintaining flexibility. Additionally, the structured use of mock data allows for thorough testing and demonstration of application functionality without the need for a live back-end. This setup can be seen as the foundation for further development and real-world integration, such as connecting to actual blockchain networks or incorporating actual user accounts.

Chapter 5

Evaluation

5.1 Introduction

The prototype, referred to as Mask Identity, was designed to improve the major usability flaws in current self-sovereign identity applications with a strong emphasis on trust, transparency, and user experience. The prototype was evaluated in a comparative user study against PrivadoID, an existing decentralized identity application that is in use today. The study's goal was to investigate how well the prototype performed in a real-world scenario and whether its design and functionality could solve issues that have been identified in the other apps. Such issues include: lack of information on functionalities and use cases, inadequate strengthening of mental models, overuse of technical jargon, and low trust due to unclear data storage as well as poor visual design in general.

PrivadoID was selected for comparison out of the three previously analyzed apps for many reasons. First, it is explicitly advertised as a self-sovereign identity application, whereas SelfKey and Truvera are more general-purpose apps, and SelfKeys cryptocurrency features would likely distract users. Second, PrivadoID has a noticeably cleaner and more modern visual design compared to SelfKey and Truvera. Its layout is simpler, with good use of icons and generally more visually appealing typography and color choice. This design makes it easier to navigate and also gives the impression of a more professional and trustworthy application. Third, PrivadoID was the only application among the three that provided a working mechanism of claiming a new credential. SelfKey and Truvera did not provide any mock or real credentials that could be claim, leaving the users with not many actions to perform. This made it impossible to compare these apps against the prototype, as there were no comparable task flows. To ensure that participants could engage with this feature, the mobile version of PrivadoID was chosen for the study. The desktop version did not support credential claiming and therefore lacked essential functionality. Testing the mobile version ensured that both applications offered comparable interactions, including credential claiming and storage, or DID display and onboarding elements.

However, PrivadoID still lacks important transparency features in data handling, does not clearly communicate the identity of credential issuers, and offers minimal onboarding

or guidance to users. All of these shortcomings are documented in the problem statement for existing SSI applications.

5.2 Study Design and Methodology

5.2.1 Participants

The study involved seven participants from different demographic backgrounds. Ages ranged from 26 to 70, and the participants included students, working professionals, and individuals with various levels of technical abilities. This aligns with Nielsen's recommendation that testing with five or more users reveals most major usability issues [35]. The participant group included mostly individuals with nontechnical backgrounds. One participant had a technical background, while another was 70 years old and had no prior exposure to digital identity systems.

Before the test was conducted, all participants filled out a pre-questionnaire that captured basic demographic information, previous experience with digital wallets or decentralized identity apps, and how trustworthy they perceived digital identity applications. These data helped contextualize post-test results and interpret observed patterns. The complete pre-questionnaire can be found in the Appendix B.

5.2.2 Environment

The evaluation was conducted as an unmoderated usability study, which means that the participants completed tasks on their own in a separate room, without the supervision of a supervisor. According to usability testing guidelines [31], unmoderated testing allows users to interact more naturally with systems without pressure. It better reflects real-life conditions. Participants were asked to complete the test in a quiet environment without distractions and to test both applications in separate sessions. To avoid a learning bias, some participants tested PrivadoID first, others Mask Identity.

5.2.3 Mixed Method Approach

The evaluation followed a mixed-method approach that combines both quantitative and qualitative methods. The quantitative part included Likert scale responses to standard usability questions as well as other questions that focused on trust and transparency in particular. Qualitative insights were collected through open questions that provided a deeper look at user experiences and personal opinions. This combination of those two methods aligns with the recommendations from the Nielsen Norman Group and human-computer interaction (HCI) researchers, who emphasize that usability is best understood through both statistical indicators and observed user reactions [29, 35].

5.2.4 Task Structure

Participants were asked to complete a set of tasks in Mask Identity and PrivadoID. The tasks aimed at evaluating core interactions found in self-sovereign identity systems **??**.

The tasks for the two apps were designed as similarly as possible to ensure consistency but had some substantial differences due to the different functionality and supported features. This aligns with the view that usability studies should be tailored to the technical constraints and specific context of the application being evaluated [29].

In Mask Identity, participants performed these tasks in a simulated role: John Appleseed, who is a university student and wants to request a course certificate (credential) from the University of Zürich. In addition, he wants to present his university degree to a company (verifier) without reviewing his grade-point average. This fictional scenario aimed to create a meaningful context and show the participants what the use cases for the application might be. The task set included the following:

- Take a look at the credentials and their detailed information.
- Request a new credential from the University of Zürich.
- Share the university degree to a company (verifier) without revealing your GPA.
- Check if your drivers license is still valid.
- Get a general understanding of the application. Find your DID address.

In PrivadoID, participants acted as themselves and were asked to set up the application and claim the 'proof of liveness' credentials. To do so, participants had to scan their face, which they were free to skip if they were not comfortable sharing their facial data. This approach reflects real-world differences in user behavior and highlights how applications need to handle different levels of user trust. The tasks included:

- Application setup
- Claim new credentials by performing a facial scan. (Optional)
- Take a look at the credentials and their detailed information.
- Get a general understanding of the application. Find your DID address.

5.2.5 Evaluation Categories

The evaluation was based on four main categories, grounded in existing literature. Each category included a mix of positive and negative statements to minimize agreement bias. The participants responded using a 5-point Likert scale ranging from 'strongly disagree' to 'strongly agree'. The full questionnaire can be found in the Appendix. B

System Usability Scale (SUS)

The SUS is a 10-item questionnaire that serves as a standardized tool to evaluate usability. It was developed by Brooke [7] and it captures the overall ease of use of a system across subjective dimensions like complexity, ease of learning, and consistency. SUS has been widely used in industrial and academic settings for decades and is particularly suited for quick assessments across different systems.

Trust

Trust was evaluated based on indicators such as user perception of data security, professional appearance, and issuer credibility. These dimensions were partially drawn from the literature that emphasizes that trust in SSI is often undermined when users feel unsure about data storage or the legitimacy of credentials [29, 40]. The participants responded to the following statements:

1. I felt secure while using the application.
2. I trusted the information shown in the application.
3. I was unsure whether the credential issuers were legitimate.
4. I felt confident the application would handle my data responsibly.
5. The application appeared professional and competent.

Transparency The transparency statements targeted data sharing and storage, clarity of issuer identity, and relevant feedback.

As highlighted in several SSI evaluations, most applications fail to communicate what happens to data after it is entered, who has access, and whether the information is stored locally or in the cloud. These are the core factors that affect user trust and adoption [40].

The participants responded to the following statements:

1. I knew who would receive my data.
2. I understood what data would be shared.
3. The issuer of the credential was not clearly identifiable.
4. I understood where my data is stored (locally or in the cloud).
5. The application gave clear feedback when data was shared or received.

UX and Emotional Response

This category is an extension of the SUS, in the sense that it also covers usability aspects. However, it goes into more detail on User Experience and other key concepts like the quality of visual design, cognitive load, familiar concepts and metaphors, navigation and error handling. UX research shows that aesthetic appeal and perceived control directly affect how users judge system quality and trustworthiness, especially in mobile apps [29, 40].

Open-ended follow-up questions in the UX section allowed participants to describe their emotional reactions and aspects of the application they particularly liked or disliked:

1. What did you like about the design or the application in general?
2. What did you not like or find confusing about the design or the application in general?
3. Did any part of the application make you feel frustrated, anxious, or unsure?
4. How would you describe your overall experience in a few words?
5. Any suggestions for improvement?

5.2.6 Scoring Methodology

The scoring mythology of all for categories is essentially the same with adaptions in the formula to account for the number of positive and negative statements. In a positive statement, the answer 'strongly disagree' results in +1 points and 'strongly agree' in +5 points. In a negative statement, it is -1 and -5, respectively. The four categories were converted into a normalized scale of 0 to 100, which allows a direct comparison between the metrics. It also allows the data to be clearly presented in bar charts, which can be seen later in the chapter. Normalizing scores in this way follows established best practices in usability research and helps to summarize and analyze the results in a consistent and meaningful way [7].

The following formulas were used:

- **System Usability Scale (SUS):**

$$\text{SUS Score} = (20 + \sum \text{responses}) \times 2.5$$

  (Like in Brooke's original design)

- **Trust:**

$$\text{Trust Score} = (1 + \sum \text{responses}) \times 5$$

- **Transparency:**

$$\text{Transparency Score} = (1 + \sum \text{responses}) \times 5$$

- **UX & Emotional Response:**

$$\text{UX Score} = (4 + \sum \text{responses}) \times 3.125$$

5.3 Results and Discussion

5.3.1 Quantitative Results Overview

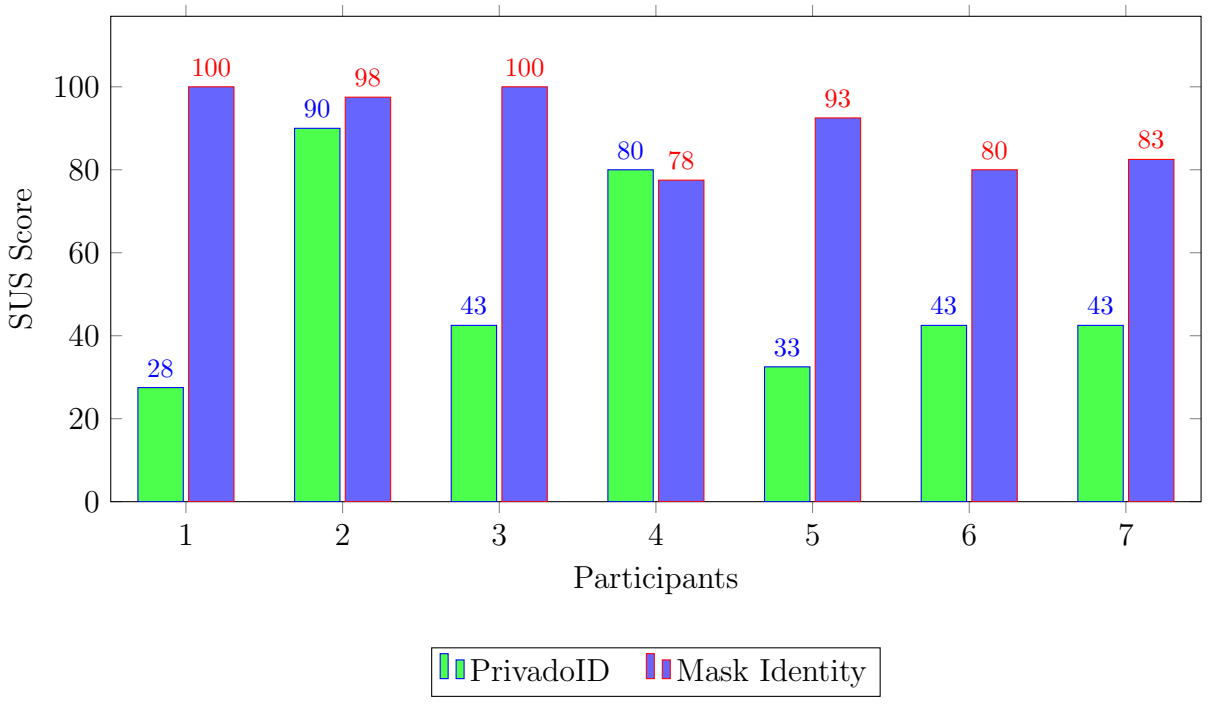


Figure 5.1: SUS Scores for PrivadoID and Mask Identity

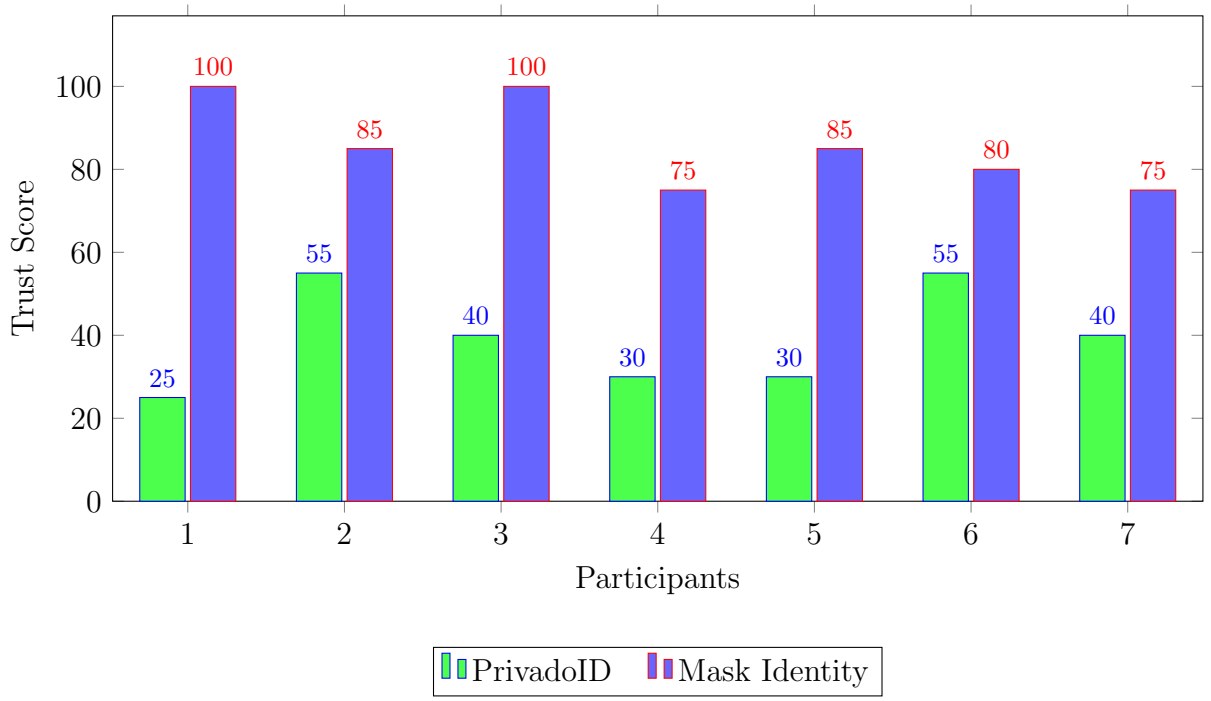


Figure 5.2: Trust Scores for PrivadoID and Mask Identity

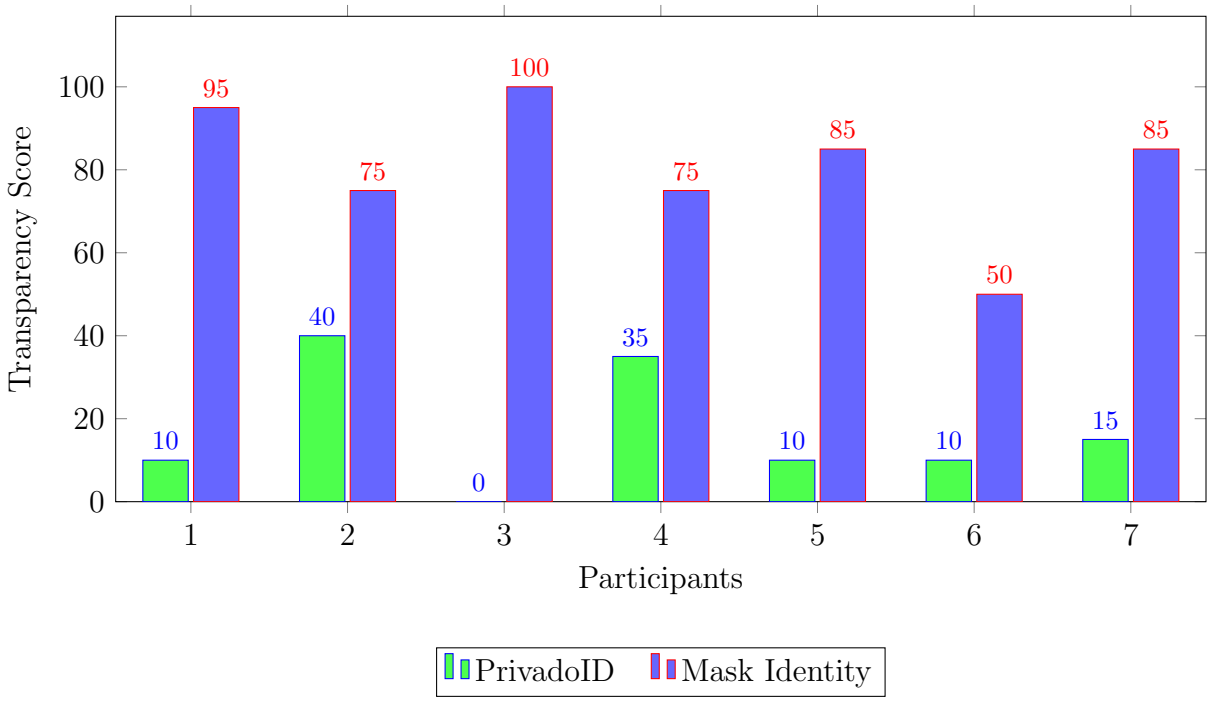


Figure 5.3: Transparency Scores for PrivadoID and Mask Identity

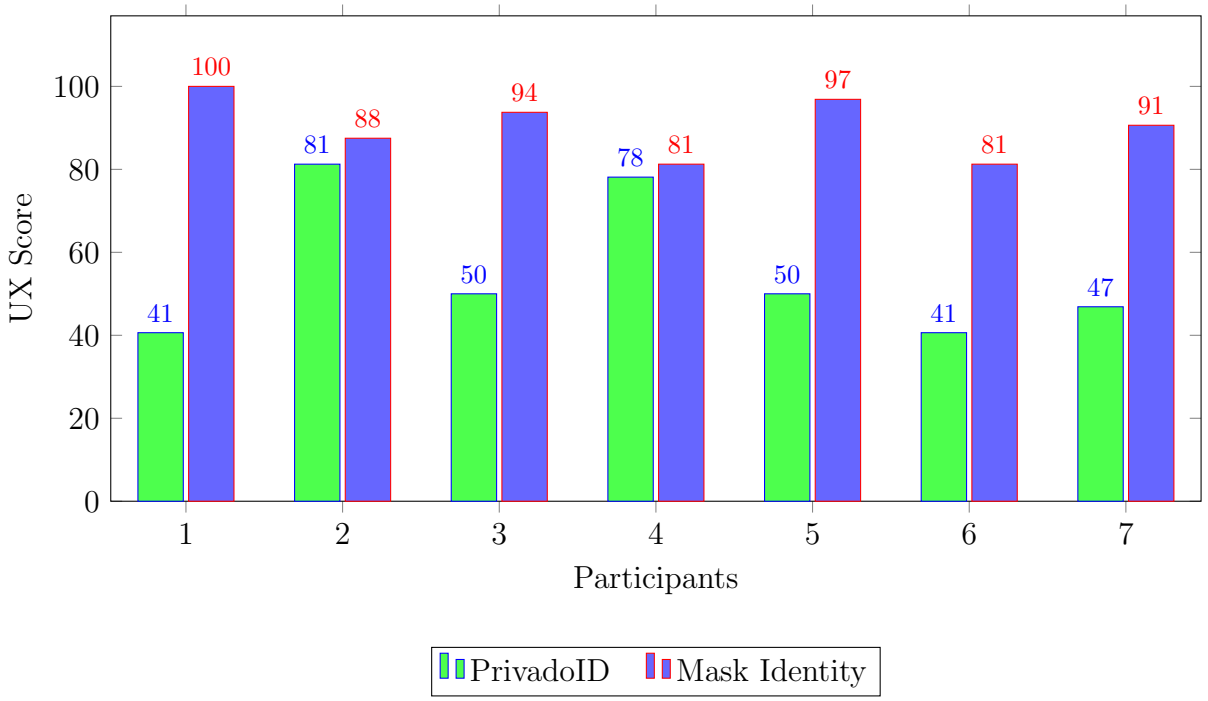


Figure 5.4: UX Scores for PrivadoID and Mask Identity

5.4 Results and Discussion

This section presents the analysis and critical discussion of the results of the user study, focusing on how participants experienced Mask Identity and PrivadoID. The goal is to determine whether the usability, trust, transparency, and UX features proposed and implemented in the prototype led to measurable improvements. These results are important not only for evaluating the prototype, but also for validating the design guidelines, features, and problem statements discussed in Chapters 2.

5.4.1 Quantitative Results

The study used a normalized scoring system (0–100) to evaluate four categories: System Usability Scale (SUS), Trust, Transparency, and UX and Emotional Response. As can be seen in Figures **??**, Mask Identity significantly outperformed PrivadoID in all four categories for nearly every participant.

SUS Scores

Participants consistently rated Mask Identity high on usability, with an average SUS score of 90.0, compared to 50.1 for PrivadoID. This big difference suggests that the design principles embedded in the prototype, such as clear navigation, meaningful feedback, and intuitive task flows, made the application easier and more pleasant to use. Some questions

in the SUS concern the learnability of the application. The users indicated that Mask Identity was more learnable and felt more confident that they could use the system without much technical knowledge. This suggests that elements like the info message on top of the container, the landing page, or the dedicated info section in the application may have helped users better understand the system and feel more in control.

Trust Scores

The trust scores also showed a strong improvement: Mask Identity received an average of 85.7, while PrivadoID received 39.3. Participants did not feel particularly secure when using PrivadoID, and they were not sure if the credential issuer (Synaps) was legitimate. In Mask Identity, they had a greater sense of security and trusted the information provided by the user interface more. This increased trust could be attributed to the consistent use of verified icons, clear status indicators, and other visual cues that reinforced trust.

Transparency Scores

Transparency was where the largest gap, with a 63.6 point difference. Mask Identity achieved an average score of 80.7, while PrivadoID reached only 17.1. This supports the hypothesis that lack of clarity around data handling is one of the most severe trust blockers in SSI applications. Although PrivadoID presented legal agreements and privacy policies, most of the participants did not trust the face-scan mechanism. Several users felt uncomfortable submitting biometric data, especially because the process was carried out by a third-party provider (Synapse), without a clear explanation of what happens to their data afterward. At least one participant chose not to claim the credential at all, indicating a high level of distrust. This uncertainty around who processes the data, how long it is stored, and where it goes was a major concern. In contrast, Mask Identity used plain language, visual cues, inline messages to explain data flows.

UX and Emotional Response

Finally, the UX score averaged 91.2 for Mask Identity compared to 55.4 for PrivadoID. This reflects a strong performance in visual design, positive emotional response, and clarity. Users praised Mask Identity for its modern look, pleasant color scheme (like the gradients), and minimalist layout. They also appreciated trust-building UI elements such as icons, issuer badges, and immediate feedback after every major action.

5.4.2 Qualitative Feedback and Observations

The open responses (Appendix E) provide rich insight into why the prototype performed better and where there are still some issues.

Mask Identity

Participants praised Mask Identity for its modern, minimal and visually appealing design, describing the application as 'intuitive,' 'clean and modern,' and 'stress-free.' Users also highlighted how the interface helped them feel in control and secure. A recurring positive opinion in the feedback was the clear structure and logical navigation:

> "It was very easy to use, intuitive navigation, great instructions, good overview of credentials." – Participant 3

> "The credentials were displayed as cards. This made it very intuitive to use." – Participant 1

> "It was nice and simple. I felt like I knew what I was doing and why." – Participant 5

Participants also appreciated the provided feedback. For example, the inline DID verification mechanism, which contributed to user confidence:

> "Once I entered an invalid DID, it notified me instantly." – Participant 5

The visual design was also received positively. The modern color schemes with color gradients and the clear structure and the logo contributed to a professional and legitimate look:

> "It felt secure but also appealing. I liked the gradient and use of colors." – Participant 6

> "Not overloaded. Clear structure. The logo is simple and unique." – Participant 7

However, there were still areas for improvement, particularly in onboarding and use of technical terminology Several participants wanted a short introduction or tutorial when they first launched the application:

> "I would have liked to have the introduction of the application in the beginning." – Participant 1

> "Maybe some more guidance for first time users." – Participant 6

Terms like "DID" were confusing to some:

> "Confusion between holder DID and certificate DID." – Participant 4

This feedback directly supports and validates two of the key design guidelines developed earlier in this thesis: user education and onboarding, and simple terminology.

There were also interface suggestions:

> "Input validation on the share screen." – Participant 2

> "Perhaps a rollover explanation might help people with little IT experience." – Participant 7

These suggestions confirm the importance of building better mental models, aligning interface concepts with familiar metaphors, and embedding more contextual help, especially for users with limited technical backgrounds [40].

PrivadoID

PrivadoID received mixed feedback. On the positive side, users commented that the interface was clean, consistent, and not overly cluttered:

> "Modern UI." – Participant 3

> "Very few buttons, which is good. No unnecessary clutter." – Participant 5

> "Simple structure." – Participant 7

Some appreciated the onboarding flow during setup:

> "Onboarding process." – Participant 2

Despite its clean appearance, the application was criticized for a lack of guidance and confusing structure:

> "No explanation or tutorial." – Participant 3

> "Didn't understand what the different sections were all about." – Participant 1

> "I felt a bit confused as to how it works." – Participant 3

> "It lacked some kind of introduction." – Participant 1

> "I felt a little bit lost with the functions of the application." – Participant 6

Transparency was a particularly serious issue. Many users expressed concerns with the face scanning feature, especially since the process involved a third-party (Synapse) and did not have a clear explanation:

> "The face scanning. Where my data goes. What it is used for." – Participant 3

> "Understanding how the data I gave the application would be integrated or transmitted." – Participant 3

This discomfort led to a breakdown in trust:

> "Very easy but little reassurance of legitimacy." – Participant 4

> "The whole process is very barebones and opaque." – Participant 5

The participants also criticized how the credential data was presented. They mentioned that it felt too technical and poorly organized:

> "The data displayed wasn't named or labeled in a way I would know what it is." – Participant 5

> "Credential details page could be slightly more appealing... better organization instead of just one list." – Participant 2

As with Mask Identity, participants emphasized the need for better onboarding, guidance, and contextual information:

> "More information in the beginning and maybe an FAQ section." – Participant 1

> "Guide the user more through the application. Show more about privacy and security." – Participant 6

5.4.3 Critical Reflections and Limitations

The results clearly show that Mask Identity achieved its goal of creating a more usable, transparent, and trustworthy SSI application. However, it is important to acknowledge a few limitations:

Sample Size: With seven participants, the study provides valuable insights, but lacks statistical meaningfulness with a high number of participants. However, according to Nielsen, five users are enough to uncover most major usability problems [35].

Prototype vs. Production application: Mask Identity was tested in a controlled mock setting. PrivadoID, while functional, introduced real-world issues such as a failure in face scanning and unclear wallet integration. These may have influenced perception, but also reflect challenges real users would face.

Platform Differences: A further limitation is that the platforms of the two applications were different: Mask Identity was tested as a desktop web application, while PrivadoID was tested as a native mobile application. User expectations and behaviors may vary significantly between desktop and mobile environments. For example, mobile users may expect more simplified flows, while desktop users may tolerate more information density and larger interface elements. This difference could have influenced the way users interpreted navigation structure, readability, and general usability in each application.

Different Features: PrivadoID does not support selective disclosure for credential sharing or in-application credential requests, which may have skewed the comparisons. However, the evaluation focused on shared themes (navigation, transparency, trust), which remain valid points of comparison.

Bias Toward Novelty: Since participants interacted with Mask Identity as a new prototype, there may have been a novelty bias. Nevertheless, the detailed qualitative feedback suggests that participants were not only focusing on the look and feel of the application, but also considered the way the application helped them understand and trust what they were doing.

Potential Positive Bias Toward the Prototype: Another limitation is that the participants knew that Mask Identity was developed by the organizer of the study. This could have unconsciously influenced them to provide more favorable feedback, either out of politeness or encouragement. Although every effort was made to present the applications neutrally, the possibility of an implicit positive bias cannot be fully ruled out.

5.4.4 Summary of Key Findings

The results of the user study show that Mask Identity clearly outperformed PrivadoID in all evaluated areas: usability, trust, transparency, and overall user experience.

Participants found Mask Identity to be much more transparent. The app clearly showed who issued each credential, what data was being shared, and where it was stored. Visual elements like verified icons and status badges helped users feel more confident and secure. The app also gave immediate feedback, for example when entering an invalid DID, which built trust and helped users feel in control. The modern and clean design was praised for being calming and easy to use. Users liked the layout, color scheme, and simple structure. The use of cards and logical navigation made the app feel intuitive. Many participants described their experience as stress-free and trustworthy.

However, some areas still need improvement. Several users mentioned that the app should offer a short introduction or tutorial, especially for first-time users. Technical terms like 'DID' were confusing to some, and a few participants suggested adding small explanations or tooltips to help them understand key functions.

In contrast, PrivadoID received mixed feedback. Some participants liked the minimal interface and clean layout, saying that the app felt simple and uncluttered. A few users also appreciated the onboarding flow at the beginning.

But many users said they felt confused and unsure what to do. The app lacked explanations or guidance, and the navigation was unclear. A major issue was the face scanning feature, which used a third-party service without clearly explaining how the data would be used or stored. This caused concern and reduced trust. One participant even refused to complete the task because of it. Users also found the credential data hard to understand, with poor labels and no clear organization.

Chapter 6

Final Considerations

6.1 Summary

The primary goal of this thesis was to improve usability, trust, and transparency in DI and SSI applications. To achieve this, a clear methodology was followed that consisted of three main phases: first, the extraction of usability and trust-related guidelines from existing research as well as a concrete feature catalog; second, the design and development of a prototype called Mask Identity based on these guidelines and features; and third, the evaluation of this prototype through a comparative user study against an existing SSI application, PrivadoID. Several key achievements were reached. A practical set of UX features was created, focusing on enhancing trust and transparency and the unique challenges of decentralized identity systems. Building on these principles, a functional and user-friendly prototype was implemented. Finally, the prototype's effectiveness was validated in a structured user study, which showed that Mask Identity performed significantly better than PrivadoID in terms of usability, trust, transparency, and overall user experience.

6.2 Conclusions

6.2.1 Achievement of Objectives

The main goal of this thesis, to develop a structured set of usability and trust-enhancing guidelines and features for SSI applications and validate them through a prototype, was successfully achieved. Mask Identity demonstrated that it is possible to design an SSI system that feels accessible, understandable, and credible to users. Its clear advantage over an existing SSI application in terms of usability, trust, and transparency was confirmed through empirical user testing.

The specific objectives were also fully met. A thorough literature review on the relationship between usability, UX, and trust in decentralized identity systems laid the foundation

for the work. Existing SSI applications were analyzed to identify recurring UX problems. Based on this research, the guidelines were applied in the design and implementation of a functioning prototype. Finally, a comparative user study was conducted to evaluate the effectiveness of the proposed solutions and to gather practical feedback from users.

6.2.2 Factors Contributing to Success

Several factors played a critical role in the success of the project. One important aspect was the solid literature foundation, which allowed for an extraction of relevant principles and best practices. Another factor was that the project maintained a strong focus on visual design, user education, and transparency throughout all development stages. Especially the visual design was refined repeatedly during the course of the project. Initially, wireframes were used in a divergent idea-finding phase to explore different design choices. In a more convergent phase, Figma [14] was used to develop and structure these ideas into interfaces. During the implementation phase, certain design elements were adjusted once more. This was not originally planned, but ultimately contributed to the success. These later refinements, particularly regarding aspects like background color, card layout, hover effects, and brand identity, helped to polish the overall user interface. As a result, the visual and functional quality of the application was significantly enhanced through continuous iteration.

Another aspect that contributed to the achievement of the goals was a structured evaluation framework that combined both qualitative and quantitative methods. The quantitative data allowed for a clear and efficient comparison of two prototypes across four dimensions, usability, trust, transparency, and overall user experience. Each of the seven participants rated both applications, resulting in a rich dataset that revealed consistent trends. Even with a relatively small sample size, the structured scoring approach made it possible to identify significant differences between the two systems quickly and accurately. In addition to the quantitative analysis, the qualitative part of the study, (e.g. the open-ended questions) offered valuable insights into the specific strengths and weaknesses of each application. These responses helped to explain the reasons behind the scores, highlighting design elements that users particularly appreciated, as well as areas that caused confusion or frustration.

6.2.3 Limitations

Despite its successes, the thesis also had some notable limitations. The prototype did not address backup, recovery, or alternatives to seed phrases, even though these aspects represent significant usability issues in current SSI systems. The evaluation was also limited to a web-based prototype and did not involve integration with a real blockchain backend. Moreover, the user study had certain limitations, as described in Section 5.4.3, particularly regarding participant number and platform differences, which may have influenced the results. Lastly, although the application provided onboarding and contextual help, even more instruction and user education elements could further improve the experience, especially for users with limited technical background.

6.2.4 Main Considerations and Lessons Learned

Several key insights emerged from the work. First, transparency and trust are essential for the adoption of SSI systems. Users must understand exactly how their data is handled and who has access to it. If users are asked to upload documents or perform a facial scan without knowing precisely who the other party is, trust is easily lost. Second, informative onboarding, user guidance, and clear instructions are crucial for building user confidence. Users need to understand what they are doing to feel in control and to trust the system. Visual design also plays a major role, as a professional, modern appearance strongly influences credibility perception. Another important lesson is that users expect familiar mental models when interacting with new systems. Concepts like wallets and cards help bridge the gap between complex decentralized identity management and users' existing mental models. All of these aspects were carefully considered during the design of Mask Identity and were central to achieving the project's main goals.

6.2.5 Main Difficulties Encountered

One major challenge during the project was the selection of applications for evaluation. Many decentralized identity apps were not usable for this study because they were restricted to businesses, were not freely available, or offered too few features to allow meaningful evaluation. In many cases, the documentation was poor, and apps had to be tested manually before it became clear that they were unsuitable. This process was unexpectedly time-consuming.

6.2.6 Modifications During Execution

During the project, some minor adjustments to the original plan were undertaken. Initially, backup, recovery, and seed phrase usability were included in the problem statement. However, these aspects were later removed because they could not be addressed within the scope of the thesis. Their technical complexity would have required extensive implementation and security considerations, which would have exceeded the available resources and time. As a result, the focus was narrowed to core usability and trust aspects.

It was also initially planned to use an existing backend and integrate certain blockchain functionalities into the prototype. However, this idea was discarded during development. While technically feasible, the integration was not essential to the thesis goals, as the core objective was to explore how interface design could support usability and trust. Existing research suggested that highlighting technical details too strongly, such as blockchain interactions, could actually overwhelm users. Therefore, the prototype was limited to front-end features, without direct blockchain integration. This allowed for a more focused implementation and evaluation of the design principles that contribute to user trust and transparency.

6.2.7 Relationship to Initial Timeline

The project timeline deviated slightly from the original plan. The research phase took longer than expected, mainly because a thorough understanding of the field required more time than anticipated. This delay pushed back the start of the design and implementation phases. However, the design and implementation proceeded smoothly and within the planned duration. Overall, despite early delays, the project was completed within a reasonable timeframe.

One unplanned development was that the design continued to evolve during the implementation phase. Initially, the design was expected to be finalized before development began. However, several visual and functional aspects were refined or changed while the application was being built.

6.3 Future Work

Several directions for future research and development emerged from this work. One important step would be to design and implement more robust backup and recovery solutions for SSI systems. For example, social recovery methods or more intuitive alternatives to seed phrases could help overcome significant usability challenges. Another important improvement would be the integration of a real decentralized backend. Full testing, including blockchain-based credential issuance and verification, would make the evaluation even more realistic and could further strengthen user trust if sufficient feedback on these blockchain interactions is provided. Better onboarding tutorials could also be developed, potentially including interactive walkthroughs that users can revisit at any time. In general, finding ways to inform users effectively, without overwhelming them, is an important goal. For example, information could be delivered through popups or tooltips. Improving the visualization of issuer legitimacy and credential provenance is another area worth exploring, as it could make the system even more transparent and trustworthy. Future studies should also include a broader and more diverse participant pool to capture a wider range of users. Finally, conducting a study over a longer time frame would allow researchers to observe how trust, usability, and user behavior develop over time when interacting with decentralized identity applications.

Bibliography

[1] Alchemy. *List of 57 Decentralized Identity Tools (2025)*. 2025. URL: https://www.alchemy.com/dapps/best/decentralized-identity-tools.

[2] Ana A Andrade, Vitor V Lopes, and Augusto Q Novais. "Quantifying the impact on distrust of e-commerce trust factors: A non-parametric study". In: *2012 International Conference for Internet Technology and Secured Transactions*. IEEE. 2012.

[3] Oscar Avellaneda et al. "Decentralized identity: Where did it come from and where is it going?" In: *IEEE Communications Standards Magazine* 3.4 (2019).

[4] Nigel Bevan. "Classifying and selecting UX and usability measures". In: *International Workshop on Meaningful Measures: Valid Useful User Experience Measurement*. Vol. 11. Institute of Research in Informatics of Toulouse (IRIT) Toulouse, France. 2008.

[5] Nigel Bevan. "Quality in use: Meeting user needs for quality". In: *Journal of systems and software* 49.1 (1999).

[6] Nigel Bevana, Jurek Kirakowskib, and Jonathan Maissela. "What is usability". In: *Proceedings of the 4th International Conference on HCI*. Citeseer. 1991.

[7] John Brooke et al. "SUS-A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194 (1996).

[8] Bybit Learn. *Best Decentralized Identity Projects in 2024*. 2024. URL: https://learn.bybit.com/defi/best-decentralized-identity-projects/.

[9] Sandy C Chen and Gurpreet S Dhillon. "Interpreting dimensions of consumer trust in e-commerce". In: *Information technology and management* 4 (2003).

[10] Coin Bureau. *SelfKey Review: Your Digital Identity on the Blockchain*. 2023.

[11] Coin Bureau. *Top Digital Identity Networks: Blockchain Identity Solutions*. 2024. URL: https://coinbureau.com/analysis/top-digital-identity-networks/.

[12] Dock. *Dock: Decentralized Identity Solutions*. 2023.

[13] EMURGO Africa. *Examples of Decentralized Identity Solutions*. 2024. URL: https://www.emurgo.africa/blog/posts/example-of-did-solutions.

[14] Figma Inc. *Figma - The Collaborative Interface Design Tool*. https://www.figma.com/. 2024.

[15] Carlos Flavián, Miguel Guinalíu, and Raquel Gurrea. "The influence of familiarity and usability on loyalty to online journalistic services: The role of user experience". In: *Journal of Retailing and Consumer Services* 13.5 (2006).

[16] Gartner, Inc. *Decentralized Identity Reviews and Ratings*. 2025. URL: https://www.gartner.com/reviews/market/decentralized-identity.

[17] Gataca. *User Experience (UX) in Self-Sovereign Identity*. 2023.

[18] Fariba Ghaffari et al. “Identity and access management using distributed ledger technology: A survey”. In: *International Journal of Network Management* 32.2 (2022).
[19] Devam Ghoghari. *What is a Mental Model? Definition and examples*. 2025. URL: https://octet.design/journal/mental-model/.
[20] BlockApps Inc. *Challenges and Solutions in Implementing SSI*. 2024. URL: https://blockapps.net/blog/challenges-and-solutions-in-implementing-ssi/.
[21] International Organization for Standardization. *ISO 9241-11:2018 Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts*. 2018. URL: https://www.iso.org/standard/63500.html.
[22] ISO. *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*. 2019.
[23] Kaleido. *Polygon ID on Kaleido: Privacy-Preserving Identity Infrastructure*. 2023.
[24] Alina Khayretdinova et al. “Conducting a usability evaluation of decentralized identity management solutions”. In: *Selbstbestimmung, Privatheit und Datenschutz: Gestaltungsoptionen für einen europäischen Weg*. 2022.
[25] Gabriella Laatikainen, Mekhail Mustak, and Nicky Hickman. “Self-Sovereign Identity Adoption: Antecedents and Potential Outcomes”. In: *Technology in Society* (2025).
[26] Shaobo Liang, Ziyi Wei, and Lan Zang. “An exploratory study of factors influencing user app abandonment on smartphones”. In: *Library Hi Tech* (2024).
[27] Gitte Lindgaard et al. “An exploration of relations between visual appeal, trustworthiness and perceived usability of homepages”. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 18.1 (2011).
[28] Mick Lockwood. “An accessible interface layer for self-sovereign identity”. In: *Frontiers in Blockchain* 3 (2021).
[29] Joanna Lumsden. *Handbook of research on user interface design and evaluation for mobile technology*. Vol. 1. 2008.
[30] Alexandra Mai. “Expert Mental Models of SSI Systems and Implications for End-User Understanding”. In: *Cryptology ePrint Archive* (2022).
[31] Maze. *Usability Testing Methods: A Complete Guide*. 2023.
[32] Aleecia M McDonald and Lorrie Faith Cranor. “The cost of reading privacy policies”. In: *Isjlp* 4 (2008).
[33] Moneyhouse. *Dock Labs AG Company Profile*. 2023.
[34] Kate Moran. *The Aesthetic-Usability Effect*. https://www.nngroup.com/articles/aesthetic-usability-effect/. 2024.
[35] Jakob Nielsen. *Why You Only Need to Test with 5 Users*. Nielsen Norman Group. 2000.
[36] Sebastian Sartor et al. “Love at First Sight? A User Experience Study of Self-Sovereign Identity Wallets.” In: *ECIS*. 2022.
[37] Mirjam Seckler et al. “Trust and distrust on the web: User experiences and website characteristics”. In: *Computers in human behavior* 45 (2015).
[38] Johannes Sedlmeir et al. “Digital identities and verifiable credentials”. In: *Business & Information Systems Engineering* 63.5 (2021).
[39] SelfKey Foundation. *SelfKey: Your Identity, Your Terms*. 2023.
[40] Rachelle Sellung and Michael Kubach. “Research on User experience for digital identitywallets: state-of-the-art and recommendations”. In: *Open Identity Summit 2023*. Gesellschaft für Informatik eV. 2023.

[41] Davide Strozzieri. *UI Design: The Allure of Blue*. 2020. URL: https://medium.com/@dstrozzieri/ui-design-the-allure-of-blue-1f82c1aea42a.

[42] Teck Ming Tan and Saila Saraniemi. "Trust in blockchain-enabled exchanges: Future directions in blockchain marketing". In: *Journal of the Academy of marketing Science* 51.4 (2023).

[43] Phosphor Icons Team. *Phosphor Icons*. https://phosphoricons.com/. 2025.

[44] Truvera. *Truvera Bank Demo*. 2023.

[45] Cynthia Vinney. *Mental Models in UX Design: The Ultimate Guide*. 2021. URL: https://careerfoundry.com/en/blog/ux-design/mental-models-ux-design/.

[46] Adam Wathan and the Tailwind Labs Team. *Tailwind CSS*. https://tailwindcss.com/. 2025.

[47] Alma Whitten and JD Tygar. "Why Johnny can't encrypt". In: *USENIX Security (Aug. 1999)* (2005).

[48] Wikipedia contributors. *Mental model*. https://en.wikipedia.org/wiki/Mental_model. 2025.

[49] World Economic Forum. *Reimagining Digital ID*. https://www.weforum.org/publications/reimagining-digital-id/. 2023.

[50] Evan You and the Vite Team. *Vite*. https://vite.dev/. 2025.

[51] Evan You and the Vue.js Team. *Vue.js*. https://vuejs.org/. 2025.

[52] Razieh Nokhbeh Zaeem et al. "On the Usability of Self Sovereign Identity Solutions "". In: *University of Texas at Austin Center for Identity, UTCID* (2021).

[53] Alexander Zavodovski, Alexander Rieger, and Gilbert Fridgen. "Are We There Yet? A Study of Decentralized Identity Applications". In: *arXiv preprint arXiv:2503.15964* (2024).

Abbreviations

DI	Decentralized Identity
DID	Decentralized Identifier
FAQ	Frequently Asked Questions
HCI	Human-Computer Interaction
KYC	Know Your Customer
SSI	Self-Sovereign Identity
SUS	System Usability Scale
TLS	Transport Layer Security
UI	User Interface
UX	User Experience
VC	Verifiable Credential
VP	Verifiable Presentation

List of Figures

Appendix A

System Documentation for Mask Identity

A.1 Systems Viewpoint

Mask Identity is a frontend prototype designed to improve trust, transparency, and usability in decentralized identity (DI) and self-sovereign identity (SSI) systems.It acts as a digital identity wallet that allows users to manage their verifiable credentials.It can be used in three different roles. The Credential Holder is the main user who owns and manages their credentials. The Credential Issuer is responsible for creating new credentials, managing them, and revoking them if needed. And the Credential Verifier checks if the credentials shown by the holder are real and valid.

The prototype is built to create a user-friendly experience.It supports important actions like storing, requesting, sharing, and verifying credentials, while keeping the focus on usability, trust and transparency.

A.2 Use and Installation Manual

A.2.1 Installation

Clone the repository:

```
git clone git@github.com:Paulin3000/mask-identity.git
cd mask-identity
```

Install dependencies:

```
npm install
```

Run the application locally:

```
npm run dev
```

The app will be available at `http://localhost:5173/`.

A.2.2 Basic Operation

- **Start Screen:** When the app opens, users land on the homepage where they can easily navigate between the "Credentials", "Sharing", "Settings", and "Info" sections.
- **Credential Storage:** Users can view all their credentials, check their status, and see more details on a separate page.
- **Request New Credential:** Users can request a new credential by entering the issuer's DID and signing the request.
- **Selective Sharing:** Users can choose which parts of a credential to share with a verifier using selective disclosure.
- **Settings Page:** In the settings, users can view their personal information and preferences.
- **Info Page:** In the info section, users can learn more about decentralized identity.
- **Role Switching:** Users can switch between different roles (Holder, Issuer, Verifier) at any time using the role selector at the top right of the navigation bar.
- **Issuer Role:**
  - Receive credential requests, verify the provided data, and decide to issue or reject a credential.
  - View all previously issued credentials and open each one for more details.
  - Revoke issued credentials if necessary.
- **Verifier Role:**
  - See incoming verification requests.
  - Check the validity of received credentials through blockchain-simulated verification.

A.3 Implementation Description

Technology Stack

- **Vue 3.5.13:** Core framework used for building the user interface. [51]
- **Vue Router 4.5.0:** Handles navigation between different views in the application. [51]

- **Tailwind CSS 4.0.14:** Utility-first CSS framework used for styling and layout. [46]
- **Phosphor Icons:** Icon library used for consistent and modern visual elements. [43]
- **Vite 6.2.0:** Build tool and development server for fast and efficient project setup. [50]

A.3.1 Contents of the Repository

Vue Components:

1. **App.vue**:
   - Root component that serves as the application's entry point
   - Contains the main layout structure and navigation routing
2. **dashboard.vue**:
   - Main dashboard interface showing overview of user's identity data
   - Entry point for accessing credentials and other identity functions
3. **credentials.vue**:
   - Component that displays a list of user credentials
   - Integrates with the credential store for data management
   - Allows users to browse and select their credentials
4. **credential-details.vue**:
   - Component that renders detailed information for a specific credential
   - Displays all properties and metadata of a selected credential
   - Provides options for managing individual credentials
5. **request-credential.vue**:
   - Interface component for users to request new credentials
   - Contains forms for inputting credential request information
   - Handles submission of credential requests to issuers
6. **CredentialCard.vue**:
   - Reusable component that displays credential information in a card format

- Properties include:
  `id`: Unique identifier for the credential
  `type`: Type of credential being displayed
  `subheading`: Descriptive text shown below the type
  `verified`: Boolean indicating verification status
  `holder`: Entity that holds the credential
  `issuer`: Entity that issued the credential
  `expiryDate`: When the credential expires
  `logoUrl`: URL for credential logo image
  `colorTheme`: Visual theme (yellow, orange, pink, purple, blue, grey)
  `glowy`: Boolean for enhanced visual effect
- Emits `credential-click` events when clicked

7. **CredentialVerifier.vue**:

- Component that handles the verification of credentials
- Features two modes:
  `verifier`: For third-party credential verification
  `issuer`: For credential issuance validation
- Verification checks include:
  DID verification
  Issuer's signature validation
  Credential hash verification
  Expiration date check
  Revocation status check
- Uses `VerificationItem` components to display verification steps
- Emits `verified` events when verification completes

8. **DataContainer.vue**:

- Generic container component for structured data display
- Properties include:
  `title`: Header text for the container
  `variant`: Visual style (primary or secondary)
  `subtitle`: Secondary descriptive text
  `subtitleIcon`: Icon component to display with subtitle
  `iconSize`: Size of the subtitle icon
  `iconWeight`: Weight of the subtitle icon
- Used as a wrapper for content sections throughout the application

9. **BaseButton.vue**:

- Foundational button component used throughout the application

- Provides consistent button styling and behavior
- Likely supports different variants and states

10. **IconButton.vue** (referenced in CredentialVerifier):
    - Specialized button component that includes icon support
    - Used for actions in the credential verification process
    - Extends BaseButton with icon positioning capabilities
11. **VerificationItem.vue** (referenced in CredentialVerifier):
    - Component for displaying individual verification steps
    - Shows title, description, and status of verification checks
    - Supports different status states (initial, verifying, verified, failed)

Store:

1. **credentialStore.js**:
    - Centralized store for managing credential data
    - Handles state management for credential-related operations
    - Provides methods for adding, retrieving, and verifying credentials

Assets and Styling:

1. **style.css**:
    - Global stylesheet for the application
    - Implements consistent styling across all components
    - Defines color variables, typography, and shared UI elements
2. **Assets**:
    - Contains visual resources such as MaskLogo.svg
    - Used for branding and UI elements throughout the application

Appendix B

Pre-Questionnaire

The following pages contain the pre-questionnaire completed by all participants prior to the user study. It gathered demographic data, previous experience, and initial perceptions of trust in digital identity systems.

Please write your name:

Pre-Test Questionnaire

Section 1: Demographics

1. Age:

   ☐ Under 18 ☐ 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-54 ☐ 55+

2. Gender:

   ☐ Female ☐ Male ☐ Non-binary ☐ Prefer not to say

3. Occupation / Field of Study:

   _____

4. Highest level of education completed:

   ☐ Secondary school ☐ Gymnasium ☐ Bachelor's degree
   ☐ Master's or higher

Section 2: Experience with Wallets and Digital Identity

5. Have you used a digital wallet before (e.g., Apple Wallet, Google Wallet, crypto wallets)?

   ☐ Yes ☐ No

6. Have you heard of Self-Sovereign Identity (SSI)?

   ☐ Yes ☐ No

7. Have you used any decentralized identity app (e.g., PrivadoID, PolygonID, Truvera, SelfKey)?

   ☐ Yes ☐ No

Section 3: Initial Trust Perception

Please rate your agreement with the following statements (1 = Strongly Disagree, 5 = Strongly Agree):

8. I generally trust digital identity systems.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

9. I am concerned about how my personal data is handled in digital systems.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

10. I believe decentralized identity could improve privacy and control over personal data.

    1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Appendix C

User Study Task

This section contains the full task sheet provided to participants during the user study. The tasks are designed around realistic SSI interactions and tailored to the specific features of each app.

User Study Task

You will be testing two applications:

- Mask Identity (Prototype)
- PrivadoID

Please complete the following tasks in both apps. Each task simulates a real use-case for managing digital identity. Try to complete each one as naturally as possible.

Mask Identity (Prototype)

Your Role: John Appleseed, a university student at the University of Zürich

Task	Example Scenario
Credential Storage & Overview	• View your stored credentials. • Select one to view its detailed information.
Credential Request	Request a new credential: “Certificate for the course Ethics in AI” from the University of Zürich. • Use the following DID for the University of Zürich: did:ethr:0xAA0E8C1F1E262F5F9A9E4B7E520CB5DD7FE • Type: Certificate • Issuer: University of Zürich • Message: Student ID: 18-277-388
Selective Sharing	• Share your university degree with a company. • Ensure your GPA (grade point average) is **not** included in the shared data. • Use the company’s DID:did:ethr:0xAB1F9A3E4C784E3A8D1B9C6A5F12DE88ABCD • Use they company name: Fantasy Inc.
Verification Status	Check if your drivers license credential is still valid
General App understanding	• Find out your DID address. • Find out more information about the application and Decentralized Identity in general.

Privado ID

In this part of the test, you will complete tasks as yourself. You do not need to share any personal information — feel free to skip any steps you are uncomfortable with.

App setup	• Read and accept terms and conditions Choose either: • Create a local account (recommended), or • Connect with an existing crypto wallet
Claim new credential	• Claim the Proof of Liveness credential to prove you are a real person. • This step involves a face scan. If you are uncomfortable sharing your facial data, feel free to skip this step.
Credential Storage & Overview	• View your credentials. • Select one to see its detailed information.
General App Understanding	• Find out your DID address. • Try to learn more about the app and decentralized identity in general.

Appendix D

Questionnaire

This appendix includes the Likert-scale and open-ended questions used to evaluate usability, trust, transparency, and user experience after participants tested each application.

Please write your name:

Pre-Test Questionnaire

Section 1: Demographics

1. Age:

   ☐ Under 18 ☐ 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-54 ☐ 55-64
   ☐ 65-74

2. Gender:

   ☐ Female ☐ Male ☐ Non-binary ☐ Prefer not to say

3. Occupation / Field of Study:

   _____

4. Highest level of education completed:

   ☐ Secondary school ☐ Gymnasium ☐ Apprenticeship
   ☐ Bachelor's degree ☐ Master's or higher

Section 2: Experience with Wallets and Digital Identity

5. Have you used a digital wallet before (e.g., Apple Wallet, Google Wallet, crypto wallets)?

   ☐ Yes ☐ No

6. Have you heard of Self-Sovereign Identity (SSI)?

   ☐ Yes ☐ No

7. Have you used any decentralized identity app (e.g., PrivadoID, PolygonID, Truvera, SelfKey)?

   ☐ Yes ☐ No

Section 3: Initial Trust Perception

Please rate your agreement with the following statements (1 = Strongly Disagree, 5 = Strongly Agree):

8. I generally trust digital identity systems.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

9. I am concerned about how my personal data is handled in digital systems.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

10. I believe decentralized identity could improve privacy and control over personal data.

    1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Post-Test Questionnaire

Section 1. System Usability Scale (SUS)

This is a standard questionnaire that measures the overall usability of an app. Please select the answer that best expresses how you feel about each statement after using the prototype today.

		Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1.	I think I would like to use this tool frequently.	☐	☐	☐	☐	☐
2.	I found the tool unnecessarily complex.	☐	☐	☐	☐	☐
3.	I thought the tool was easy to use.	☐	☐	☐	☐	☐
4.	I think that I would need the support of a technical person to be able to use this system.	☐	☐	☐	☐	☐
5.	I found the various functions in this tool were well integrated.	☐	☐	☐	☐	☐
6.	I thought there was too much inconsistency in this tool.	☐	☐	☐	☐	☐
7.	I would imagine that most people would learn to use this tool very quickly.	☐	☐	☐	☐	☐
8.	I found the tool very cumbersome to use.	☐	☐	☐	☐	☐
9.	I felt very confident using the tool.	☐	☐	☐	☐	☐
10.	I needed to learn a lot of things before I could get going with this tool.	☐	☐	☐	☐	☐

Section 2. Trust

This section focuses on how trustworthy the app felt to you. Please indicate how much you agree with each statement based on your experience. There are positive and negative statements.

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. I felt secure while using the app.	□	□	□	□	□
2. I trusted the information shown in the app.	□	□	□	□	□
3. I was **unsure** whether the credential issuers were legitimate.	□	□	□	□	□
4. I felt confident the app would handle my data responsibly.	□	□	□	□	□
5. The app appeared professional and competent.	□	□	□	□	□

Section 3. Transparency

This section focuses on how clearly the app communicated important information, like how personal data is managed. Please rate how much you agree with each statement. There are positive and negative statements.

		Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1.	I knew who would receive my data.	☐	☐	☐	☐	☐
2.	I understood what data would be shared.	☐	☐	☐	☐	☐
3.	The issuer of the credential was **not** clearly identifiable.	☐	☐	☐	☐	☐
4.	I understood where my data is stored (locally or in the cloud).	☐	☐	☐	☐	☐
5.	The app gave clear feedback when data was shared or received.	☐	☐	☐	☐	☐

Section 4. Additional UX / Emotional Response

This section covers the app's design, navigation, and how it made you feel while using it. Please rate each statement and answer the open questions if you can.

		Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1.	The app felt modern and aesthetically pleasing.	☐	☐	☐	☐	☐
2.	It was mentally tiring to use the app.	☐	☐	☐	☐	☐
3.	I felt in control while using the app.	☐	☐	☐	☐	☐
4.	I recognized familiar concepts or metaphors in the app (like cards, wallet, ID).	☐	☐	☐	☐	☐
5.	I was able to navigate the app easily and find what I needed.	☐	☐	☐	☐	☐

6.	I often made mistakes while using the app.	□	□	□	□	□
7.	The app helped me avoid mistakes or guided me when I was about to make one.	□	□	□	□	□
8.	The app appeared professional and well-made	□	□	□	□	□

1. What did you like about the design or the app in general?

2. What did you not like or find confusing about the design or the app in general?

3. Did any part of the app make you feel frustrated, anxious, or unsure?

4. How would you describe your overall experience in a few words?

5. Any suggestions for improvement?

Appendix E

Answers to Open Questions

This section contains the digitized and aggregated responses to the open-ended questions from the user evaluation. Aside from correcting major grammatical errors, the answers have been kept in their original form to preserve the participants' wording and tone. These responses provide valuable qualitative insights into user impressions, points of confusion, and suggestions for improvement.

Open Questions - Answers

Mask Identity

1. What did you like about the design or the app in general?

- "Modern UI, very easy to use, intuitive navigation, great instructions, good overview of credentials, good color use." (A. Harris)
- The credentials page and details page with all the information and share / delete buttons. And the explanation page. (C. Gebbia)
- The app was very well designed and I found it very aesthetically pleasing. I liked how the different credentials were displayed as cards. This made it very intuitive to use. The combination of intuitive interface and clean design made it seem really trustworthy. (L. Helm)
- It felt secure but also appealing. I liked the gradient and use of colors. (L. Hüni)
- It felt very sleek and well thought out. The design felt fairly intuitive and was kept to a minimum. Once I entered an invalid DID, it notified me instantly. It notified me when something was sent or saved. (S. De Luca)
- It looks friendly but yet not sterile. Not overloaded. Clear structure.
  The logo of the app is well chosen. It is simple and unique and well chosen. (R. Müller)

2. What did you not like or find confusing about the design or the app in general?

- Nothing really, easy to understand and use, good instructions. (A. Harris)
- I need a few moments to orient myself. (G. Müller)
- I would have liked to have the introduction of the app in the beginning (maybe as a quick pop-up). Especially for people not as experienced with the concept of digital ID, this would have helped to get a quick understanding. (L. Helm)
- Nothing really. (L. Hüni)
- I did not understand if and how or whether renewing the expired license was possible. (S. De Luca)
- The personal information is listed under settings. Settings means for me rather technical nature. Menu profile would be suitable perhaps. (R. Müller)

3. Did any part of the app make you feel frustrated, anxious, or unsure?

- not really (A. Harris)
- No (C. Gebbia)
- Not really (G. Müller)
- No (L. Helm)
- Nothing (L. Hüni)
- Not really. The navigation was intuitively and information was readily available. (S. De Luca)
- I felt unsure whether the issuers were actually legitimate. There is no way to verify this! (R. Müller)

4. How would you describe your overall experience in a few words?

- Easy to use and great overview. (A. Harris)
- Clean and good overview. Simple and modern. (C. Gebbia)
- The design was clean and clear. I liked the info-page. It felt like a site I'd trust. (G. Müller)
- Overall it was a great experience and I could imagine using the app in the future. (L. Helm)
- Good experience. Felt secure, logical and appealing. (L. Hüni)
- It was nice and simple. I felt like I knew what I was doing and why. (S. De Luca)

- Stress free, good user experience (R. Müller)

5. Any suggestions for improvement?

- Input validation on the share screen. (C. Gebbia)
- I don't think that some people would intuitively know what DID to search for and where to find it. Confusion between holder DID and certificate DID. (G. Müller)
- As stated in 2. (L. Helm)
- Maybe some more guidance for first time users. (L. Hüni)
- Improving the way DID address is displayed and imported could be improved and simplified. (S. De Luca)
- Perhaps a rollover functionality might be helpful for people with little IT-experience. Different languages might also help! (R. Müller) [rollover = description when hovering on item]

PrivadoID

1. What did you like about the design or the app in general?

- Onboarding process (C. Gebbia)
- I liked the consistency in design language. Colors etc. (L. Helm)
- Pretty boring and simple design. The smileys make it approachable. All in all not bad it feels professional. (L. Hüni)
- There were just a few functionalities but they also just give you very few buttons, which is good. There is no unnecessary clutter. (S. De Luca)
- Modern UI (A. Harris)
- Simple structure. (R. Müller)

2. What did you not like or find confusing about the design or the app in general?

- That "home" is the proofs and not the credentials. (C. Gebbia)
- The app appeared to be "too well designed" for a government issued app. —> too smooth (G. Müller)
- I didn't understand what the different sections of the app were all about. For example "home" and the "credentials" section. I also found the information displayed too complicated for the average user. For example in the tab "credential details". (L. Helm)
- I would like some more guidance in the app. Kind of an introduction which shows the different features. The design could be a little bit more appealing. However, it should stay professional. (L. Hüni)
- The data displayed wasn't named or labeled in a way I would know what it is. (S. De Luca)
- No explanation or tutorial. (A. Harris)

3. Did any part of the app make you feel frustrated, anxious, or unsure?

- No (C. Gebbia)
- No (G. Müller)
- It took too long to load in the beginning. I was also quite confused what the app was all about in general. It lacked some kind of introduction. (L. Helm)
- The page with all the credentials feels a little bit confusing. (L. Hüni)
- Understanding how the data and info I gave the app would be integrated or transmitted and where / through which channels it would go. (S. De Luca)
- The face scanning. Where my data goes. What it is used for. (A. Harris)

4. How would you describe your overall experience in a few words?

- Simple, fast, minimal. (C. Gebbia)
- Very easy but little reassurance of legitimacy. (G. Müller)
- Overall the app looked very clean and aesthetically pleasing but I didn't really grasp the different features it contained. (L. Helm)
- I felt a little bit lost with the functions of the app. (L. Hüni)
- The app seems to work as intended but the whole process is very barebones and opaque. While the UI being usable it's not very friendly and almost looks like a work in progress. (S. De Luca)
- A bit confused as to how it works. (A. Harris)

5. Any suggestions for improvement?

- Credentials details page could be slightly more appealing. Maybe abstract some of the data like "isHuman: true" could be handled with an icon or something. Also better organization instead of just one list with data key/value. (C. Gebbia)
- Inform the user more about the safety concepts. (G. Müller)
- More information in the beginning and maybe an FAQ section. (L. Helm)
- Guide the user more through the app. Show the features and possibilities. And also show more about privacy and security. (L. Hüni)
- Background information about the data and its storage and who handles it and how it's handled and slightly better UI and instructions and how to use it. (S. De Luca)
- More instructions respectively any instructions at all. (A. Harris)