A Universal Framework for Claim Portability in Self-Sovereign Identity Applications

Merel Steenbergen[1] , **Martijn de Vos**[1] , **Johan Pouwelse**[1]
[1]TU Delft
{M.A.Steenbergen}@student.tudelft.nl, {M.A.deVos-1, J.A.Pouwelse}@tudelft.nl

Abstract

Self-sovereign identity (SSI) provides users of the internet control over their own data by letting them store it on their own device or in a decentralized way, such as on a blockchain. The Super App is an SSI application currently under development by the Delft Blockchain Lab, but it still lacks one of the core features of SSI, which is interoperability. In SSI applications, the user will be in control over their identity when an issuer attests to it. Services can request confirmation about the identity of a user through a verifiable claim, to which the user can reply with this attestation. This research first focuses on building a claim portability framework, which means these verifiable claims and attestations can be communicated between the Super App and other applications. This framework is designed using a public key infrastructure, as that is already present in the Super App. Before sending a claim or attestation, it is signed by the sender and encrypted with the public key of the intended receiver for security purposes. The Super App currently lacks infrastructure to assign issuers of attestations, so a Trusted Issuer registry will have to be stored somewhere in the network. To contest the adoption problem that currently exists in many SSI solutions, the usability has been evaluated as it plays a significant part in adoption. For this, some mock-up user interfaces were created and evaluated by users through a survey and some suggestions were made for improvements.

1 Introduction

Personally Identifiable Information (PII) is any data that can be used to identify a user. This could mean a name or a social security number, but it is not limited to that. Users can also be identified from a social media post, a picture, or a username. When the World Wide Web was introduced in 1990, users identified themselves with usernames and passwords, creating a new account for every service. Even though Single Sign-On has reduced the number of passwords per user, passwords are still a major security risk. In 2017, the password manager LastPass analyzed the data of employees of over 30.000 companies using the service and found that the average amount of accounts per employee is 191 [1]. This is because PII storage is still centralized, meaning that if one wants to log in to a service, the username and password are stored in a database owned by the service.

The main disadvantage of this approach is that the service controls the users' PII. As an example, the terms of service of Instagram[1] state the following: "We reserve the right to modify or terminate the Service or your access to the Service for any reason, without notice, at any time, and without liability to you". [2] clearly explains how this might impact end-users: "Because the only online identities most people have are centralized, the removal or deletion of an account effectively erases a person's online identity which they may have spent years cultivating and may be of significant value to them, and impossible to replace."

In addition, these data duplicates ensure that the estimated accumulated cost of identity assurance in the UK exceeds 3.3 billion pounds. CTRL-Shift has estimated that using 'make once, use many times' strategies could reduce this to 150 million pounds [3]. Yet those are not the only costs of centralized data management. Cybercrime and data breaches cost an estimated $450 billion US dollars per year in 2017 [4] and it was estimated to grow to $6 trillion US dollar by 2021 [5].

Self-sovereign identity (SSI) aims to solve the problem stated in the previous paragraph by providing users with complete control over their data. This is achieved with decentralized data management, such as blockchain. In this context, decentralized means user-centric; the user is the only person storing and managing their data. The TrustChain Super App [6] is a mobile application under development by the Delft Blockchain Lab. It aims to create a digital foundational identity. However, it currently cannot transfer data to other applications. This is an essential aspect of SSI to ensure that third parties can request data from a user to confirm their identity.

This research will create a secure and reliable way to transfer data from the Super App to a third party. A possible use case for this is buying alcohol online, as the Super App could be used to confirm that the buyer is actually of legal drinking age. There are some challenges to transferring data outside of the blockchain. These will be explored first in the Problem Description, then some more context on SSI will be given in

[1]Instagram's terms of service 2021

section 4, which motivates the decisions that are made in the following section, where the communication protocol will be designed. The optimal solution will be discussed in the context of the Super App. Then, the usability of muck-up user interfaces will be evaluated and the ethical aspects of the research will be reflected on in the following sections. Finally, the conclusion will contain a brief summary of the problem and solution and elaborate on future research that might be conducted in this field.

The framework that is designed in this paper, will focus on verifiable claim portability. This means that the user will be able to verify claims of applications other than the Super App. The framework will be designed in such a way that the architecture can be reused to solve the problem of full data portability. Full data portability requires extra research towards data storage, but the architecture for actual communication between applications will already exist because of this research.

2 Problem Description

Many SSI applications either aim to replace all current technology or do not have a solution for the data duplication that occurs in the databases of services like social media. They mainly focus on verifying the possession of certain documents, such as a driver's license. To reach a fully decentralized way of managing and storing data, no service should be allowed to store PII. They should rather request the data from the users themselves. Some of the SSI applications do pose a solution for the data duplication problem, those include Sovrin and uPort, which will be discussed in the Related Work section.

The Super App currently does not support the transfer of data across applications. Thus the online identity that a user assembles and stores in an application can only be used within the application itself. To define Self-Sovereign Identity, the ten principles that were devised by Christopher Allen are often used. The sixth of which is Data Portability: "Information and services about identity must be transportable" [7].

The current situation is not desirable as it implies that each application currently in use by end users would have to be replaced with an equivalent in the Super App. As mentioned previously, the average employee has 191 accounts across various platforms. The Super App has been designed to be able to replace most, if not all, of these. Still, it would be more effortless, both for users and developers, to make the Super App collaborate with other applications, rather than making it replace them.

Naturally, one of the complications of transferring PII out of the blockchain is security. Data could be intercepted or possibly even altered by a malicious user, who could reveal the data to anyone or claim to be someone they are not. SSI applications do try to solve this problem by using verifiable claims, which will be explained in the next section. These verifiable claims will also form the basis for the communication protocol that will be designed in this paper.

3 Related work

More Self-Sovereign Identity applications than just the Super App exist. [8] and [9] have made comparisons of some of the implemented solutions. These include uPort, Sovrin, Civic, and ShoCard. Next to those solutions, we will also briefly explore the BlockStack solution.

uPort uses the Ethereum blockchain, which supports the use of smart contracts [10]. These smart contracts are one of the three main components of uPort. The other two are the mobile app and developer libraries. uPort operates on the idea of selective disclosure, meaning that the user chooses the services they want to share their PII with. However, uPort users do have a public JSON profile for the registry, which can compromise the privacy of the user [9]. uPort also has some centralized components, such as the push notification centre.

The main purpose of the application of the **Sovrin** Foundation is user identification and authentication [11]. For this, they created a blockchain with a Decentralized Public Key Infrastructure. Every user has a collection of Decentralized Identifiers (DID) that can be used to find their public keys. Users can use a different DID for each service, which makes sure they are not worth stealing. Sovrin makes use of Verifiable Claims and Zero-Knowledge Proofs. This means that, in order to verify you are old enough to drink, you only need to show proof in the form of verification by the government.

SelfKey is very similar to Sovrin. It uses a blockchain in combination with a Public Key Infrastructure to identify its users. They use verifiable claims that are issued by claim issuers such as the government [12].

Civic makes use of the Bitcoin blockchain. Users can download its mobile application to create an identity. "The Civic App stores a user's PII securely on the user's phone using high-level encryption and biometric locks such as a fingerprint ID" [13]. Civic takes care of the verification of the user's PII. The attestations are stored on the blockchain and can be used by users to verify claims. Identification requests are made with QR codes, which the user can scan to view the request. The user can choose to approve or deny the request after scanning the QR code.

ShoCard uses the Bitcoin blockchain to bind a decentralized identifier to an already existing document, such as a passport. ShoCard does make use of a centralized server to help the user communicate their encrypted data to a relying party [14].

Blockstack consists of three main components: A blockchain to bind digital property like domain names to public keys, a peer network called Atlas, and a decentralized storage system called Gaia [15]. In Blockstack, complex logic is executed off-chain for security and scalability purposes. The user data is not stored on the user's device, but in the Gaia database, which is a combination of Amazon S3, Dropbox, Microsoft Azure, FreeNAS Server, and Google Drive. The responsibility of encryption of data lies with the client-side of the application.

4 Self-Sovereign Identity

There are three customary methods of managing PII, which have been depicted in Figure 1. This section will first explore these methods. Then, the concept of verifiable claims will be explained, which is a key concept of SSI and will play a crucial role in this research. The final subsection explains the drawbacks of verifiable claims.

4.1 Data management models

Isolated User Identity. The most common PII management model is the Isolated User Identity (SILO) Model [16], also known as centralized data management and described in part a of Figure 1. Each service stores the required and requested PII in their own database, which the user can access when they authenticate with a username and password. This approach introduces two main drawbacks, both of which were mentioned before. First, there is a vast amount of data duplication that exists in different databases. Second, users are not in control of their own data as any service that stores it can decide what they wish to do with it. They could choose to delete it, effectively erasing a part of the online identity of their user.

Federated Identity management. The federated identity management model, as shown in Figure 1b, removes some centralization from the SILO model. Users can log in to multiple services using one account, such as their Google or Facebook account. As several services can use the same database, it removes part of the data duplication problem. Furthermore, it is simpler for users to control their data as there is a central overview of services that have access to their account.

However, users are still not in control of their own data, so this is not a suitable final solution. The federation manages and stores the data, hence a breach in their databases compromises the PII of many users and services. Moreover, the federation could delete the one account that users identify themselves with on multiple services. This has an even larger impact than the situation where one service deletes an account as it eliminates a significantly larger portion of the user's online identity.

Self-Sovereign Identity. Figure 1c shows the Self-Sovereign Identity management model. SSI does provide this full control over data to users. The main contributor to this control is the storage of data. Data is stored in a decentralized fashion, such as on a blockchain. Sensitive data that should not be visible to anyone is stored locally on the device of the user. Of this data, not many copies exist, which makes it less complicated for the user to manage and delete data. The user is the only person with the right to grant or revoke access to their PII, they are fully autonomous. All services can make use of the same database, provided they are allowed access by the user.

4.2 Verifiable claims

Verifiable claims (VCs) lie at the heart of SSI solutions. Almost all data is sent through these claims. The key idea is that the user will never have to send sensitive information. Rather, a claim is made, to which the user can answer. The service now has the information they require and the user did not send any sensitive PII over a network.

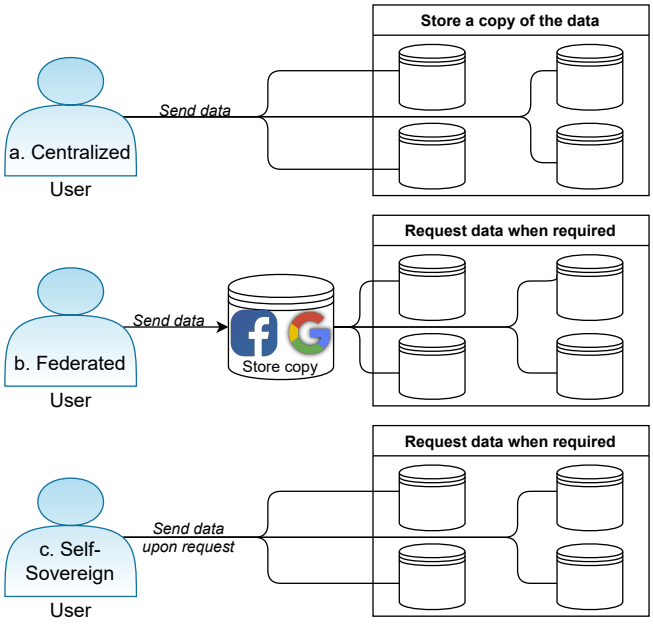


Figure 1: Data management according to the different methods

As depicted in Figure 2, there are three parties involved in the process of creating claims. The first party is the subject; the user of an application and the person that identifies themselves. The main focus of SSI is that the subject is in full control over their identity, deciding which other parties gain or lose access. However, often PII has to be verified or issued by a trusted party, the issuer. An example of an issuer is the government, as can they provide proof of a social security number or a driver's license. The final party is the relying party, which often is a service that requests the subject for identification by making a verifiable claim.

Upon receiving such a verifiable claim, the subject does not have to send PII to prove the claim. The VC acts as a polar question to which the subject can provide an answer. Instead of providing the subject's date of birth to verify they are over eighteen, they provide the attestation that was sent by the government. These signatures are combined with some metadata to ensure they can only be used for this particular claim. This metadata can, among others, contain a name, expiration date, and signature scheme [17].

In this paper, the term claim will mean the request that the relying party makes towards the subject and the term attestation will be used to imply the proof that the subject returns, which has been issued by the issuer.

Only forwarding attestations has the advantage that no actual PII is sent over a network. If a malicious user were to get hold of the data they would not get any information about the subject. There exists a trade-off between identity and privacy: The more information is disclosed about a user, the less anonymous they are. Even though SSI applications aim to enable users to identify themselves, they should still be designed with privacy in mind. This is because SSI solutions handle sensitive data, such as banking and health records [11].

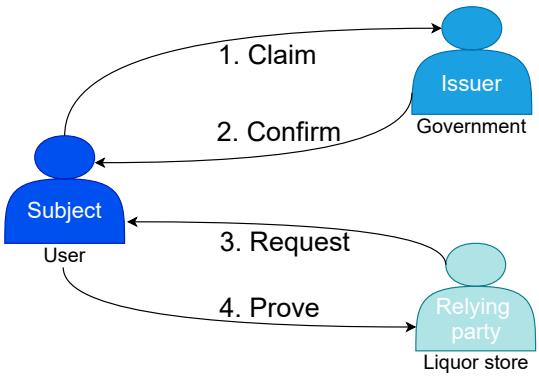


Figure 2: Parties involved in attesting data

4.3 Limited autonomy

The drawback of verifiable claims is that they are limited in the amount of data they are able to contain. Using only a polar question does not allow for any additional data to be sent. This implies that VCs alone will not be sufficient if SSI aims to replace all centralized services. Those services store more PII than can be requested through a verifiable claim. Almost every service that requires users to register with an account, stores the name of a user. This results in a great amount of duplication of stored data. However, it is hard, if not impossible, to make a verifiable claim about the name of a user when that name is not known to the service.

In many SSI solutions, this is solved by using an identifier for a user that provides access to the data storage. The SSI solution Blockstack[2] stores public key identifiers on the blockchain and offers off-chain storage for other data, such as PII. For the off-chain database, several cloud-computing applications are used, such as Dropbox and Google Drive. The advantage of this is that they are very fast in comparison with a blockchain. With this approach, it is a client-side responsibility to ensure data is properly formatted and encrypted if need be. This way, a service can request access to the data they need, without storing it locally.

However, requesting that data requires more communication than is feasible with VCs. Still, claim portability is a key step towards full data portability, so this research will represent a universal architecture for the portability of verifiable claims. Further research should be conducted towards full data portability, as will be discussed in section 10.

5 Our claim portability framework

In order to profit from data deduplication, we need to radically rethink the interoperability aspects of SSI. Adaption is one of the most prevalent problems of SSI. To combat this, the application should be easy to use for both developers and users. Therefore, we will devise a universal claim portability framework for the sharing of verifiable claims with a focus on usability for both users and developers. This communication protocol will be designed for general blockchain SSI solutions. When the design of the protocol has been finished, it will be discussed in the context of the Super App. For designing this framework, it will be assumed that the issuer and subject both make use of the Super App, while the relying party is a service using another application.

Certain decisions need to be made when designing this framework. The Super App currently does not support roles for trusted issuers, but to use the VCs as they were explained earlier, it will need to. The first subsection will explain what architecture is needed for this. Furthermore, security is a key concern of SSI, as the application will be working with sensitive data such as banking information. Security measures will be explored in subsection 5.2. Afterwards, data storage will be discussed as this sensitive data cannot be stored everywhere. Finally, there will be a subsection on the metadata that the claims contain.

5.1 Trusted Accreditation and Trusted Issuers

As previously mentioned, the Super App currently does not support issuers for claims. This is a necessary addition before VCs can replace most currently existing identifying mechanisms. The infrastructure needed for this includes two registries: A Trusted Accreditation (TA) Organization Registry and a Trusted Issuer (TI) Organization Registry. The TA is an organization that has been authorized to accredit organizations to become TIs. A TI is an organization such as the government or a university that may issue certain types of VCs, such as the possession of a driver's license or diploma. The two registries contain the information about the organizations, These registries should be stored publicly, in the peer-to-peer (P2P) network, such that everyone is able to verify them.

5.2 Smart Contracts and Public Key Infrastructures

The VCs cannot be sent in plaintext, as that would make the data too effortlessly obtainable for malicious users. Many different ways exist to ensure security during communication. Often used in SSI solutions are Smart Contracts and Public Key Infrastructures (PKI). Smart contracts are small computer programs that execute automatically on a blockchain. The logic is performed and displayed on the blockchain, while the operations are completely autonomous [18]. Sometimes these two are even combined and a public key is included in the smart contract.

Sovrin uses a PKI for encryption of VCs [11], while uPort [10] uses smart contracts. Both implementations are suitable options. However, since not all blockchains support smart contacts, PKIs are the more universal solution. Therefore our framework will be implemented using a PKI. In the Super App, a public key infrastructure already exists, as it is built upon IPv8[3]. This can be used for the encryption and signing of VCs.

The idea behind the encryption, as shown in Figure 3, is quite simple. Each user in the network has a public key and a private key. When a relying party wants to send a VC, they sign it with their own private key (1). This way, it can be verified who sent the VC. To make sure only the intended receiver

[2]https://docs.stacks.co/build-apps/guides/data-storage

[3]https://github.com/Tribler/kotlin-ipv8/

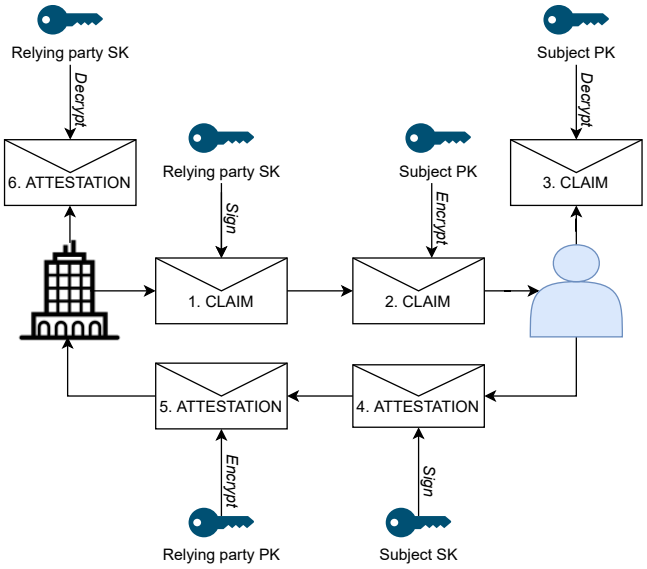


Figure 3: Signing and encrypting of claims and attestations

can use the claim, it is also encrypted with the public key of the subject (2). Now it can only be decrypted with the private key of the subject (3). When the subject wants to respond to the claim, they sign the corresponding attestation with their private key (4) and encrypt it with the public key of the relying party (5). Finally, the relying party can now decrypt the attestation with their private key and the information can be used (6).

5.3 Data storage

PII storage is an important aspect of designing an SSI protocol. Blockchains are distributed and secure, but hard to scale. As an example, Bitcoin can only handle roughly 7 transactions per second [19]. Furthermore, if there is a lot of information on the chain, the look-up time can be very high. Therefore, it is not suitable to store all information on-chain. Additionally, sensitive information should not be stored on-chain, even if blockchain were scalable and the data was encrypted, as encryption might be broken with quantum computers when given enough time. As the transactions cannot be removed from the chain, only non-sensitive data should be stored on-chain to protect the privacy of the users. With our framework, storage of PII on the user's device is a suitable solution. This is because services can obtain the knowledge they require through claims and thus never need to see the actual data. If the framework expands to handle data access requests, the decision on data storage will play a more significant part. More about this expansion of the framework can be read in section 10.

PII is not the only information that should be managed by our framework, the key pairs need to be stored as well. Public keys are usually stored on-chain, as they should be available to everyone. We will adopt this approach in our framework. Other than the subject, no one should have access to the private key. Therefore, the most suitable storage for the private keys is the device of the user, which usually is a smartphone. The smartphone is portable and widely used, which makes it a good option as well. In 2018, 84% of the Dutch citizens had access to a smartphone with internet connection [20]. Storing data on only one device poses some threats of loss of keys upon loss of the device. There are some solutions to retrieve data in this case. However, the problem of data resilience is out of the scope of this research and will not be discussed any further.

5.4 Metadata

In section 4 it was already mentioned that an attestation includes some metadata. This metadata should at least contain a validity period or expiration time and an identifier of pending transactions. This makes sure that malicious users cannot take advantage of unused or lost claims. This validity period should depend on the average response time. When multiple transactions are in process between a subject and relying party, the identifier will make a distinction between the VCs that have been confirmed and the VCs that have been declined.

Some architectures allow VCs that have not been issued by a TI. If this is the case, the metadata should also contain a value indicating whether the VC can be answered by a subject-issued attestation. In this case, the VC usually gets verified by other users in the network. However, since most claims will have to be issued by a TI, this framework will only support attestations that were issued by a TI. This removes the need for this value in the metadata of the claim.

6 Implementation details

This section will describe the framework specifically for the Super App and include figures and screenshots to explain it. It can serve as a guideline for implementing the framework in the Super App.

6.1 IPv8

The Super App is built upon IPv8[4], which is a library for creating distributed applications, based on a P2P-overlay. IPv8 includes a PKI that is ready for use. Every peer in the network has a key pair, generated with Curve25519.

IPv8 already includes an attestation service that could be used for this framework. The attestation flow is as follows: "Peer 1 and 2 can see each other and have no existing attributes, then Peer 1 requests attestation of an attribute by Peer 2 and Peer 2 attests to the requested attribute. Finally, Peer 1 checks its attributes to confirm successful attestation" [21]. In this case, Peer 2 can be an arbitrary node in the network. To add the notion of Trusted Issuers, the registries of TIs and TAs still have to be added.

6.1.1 Library

The main focus of this research is to enable other applications to communicate with the peers in the network. To do this, these applications need a key pair and thus have to be a peer in the network. Therefore, they will need to use IPv8. Since IPv8 is not very easy to integrate for developers, the application will consist of two main parts: A library that developers

[4]https://www.tribler.org/IPv8/

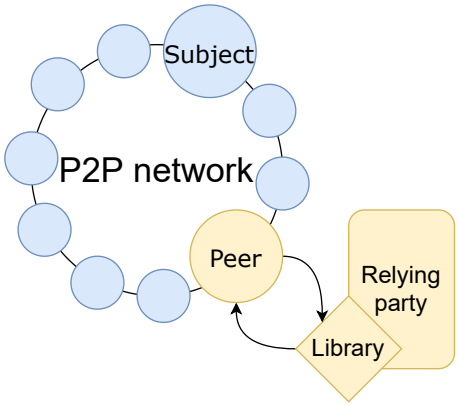


Figure 4: The connectivity of the library in the P2P network

can easily integrate into their application, and an application in the Super App itself.

Developers can easily integrate a library in their application, which improves developer usability. The library will reside in the application of the relying party. Since it connects to IPv8, the relying party will now also become a peer in the network and can thus communicate with other peers to request data from them. This connectivity is described in Figure 4. In this figure, the yellow components all belong to the relying party. The library is integrated into their application and connects to the P2P network, which enables it to communicate with possible subjects and make claims about their identity.

6.2 Super App registries

The application in the Super App will hold the TI registry, TA registry, a claim registry, and an attestation registry. The TI and TA registries will no longer be discussed here, as they were already explained in subsection 5.1. The claim and attestations registries are overviews of all the claims that relying parties have requested from the user and all the attestations that have been made by issuers.

A decision that should be made in the design of the claim registry is whether services are allowed to store attestations or can only request them for one-time use. For users, it would be preferable if services are allowed to store attestations, as it means they will not have to accept data requests every time they want to use a service. In this case, the claims will show an expiration date to the user. The subject will always have an option to revoke the claim, which takes away the right of the relying party to store the attestations. This is an aspect of the user's right to be forgotten according to the GDPR. However, it does imply that the relying party will have to make a new claim when the data is needed again.

To ensure that users are in full control over their data, only one copy should exist at a time, which should be in the user's control. As we think security and persistence are more important than this small impact on user comfort, our framework will only allow single-use claims.

7 Usability

Many research papers and white papers about SSI do not explain concretely how they will replace popular applications such as social media. They mainly focus on verifying the fact that a user has a certain document, such as a driver's license. They seem to imply that all applications will eventually make use of the SSI solution, but do not describe how the shift to a full SSI environment will take place. Adoption is an ongoing issue in SSI applications, so every SSI solution should be designed and implemented with adoption in mind. Usability is very important for adoption, so we will discuss how this will be implemented in the Super App. We make the distinction between developer and user usability.

7.1 Developer usability

For developers, it is paramount that the SSI application is easy to integrate into their own application, especially since many of them work for companies. The main focus of most companies is making money, and as the saying goes: Time is money. Eventually, businesses can save money when they use the Super App, as they will no longer have to store much data and thus do not have to pay for storage.

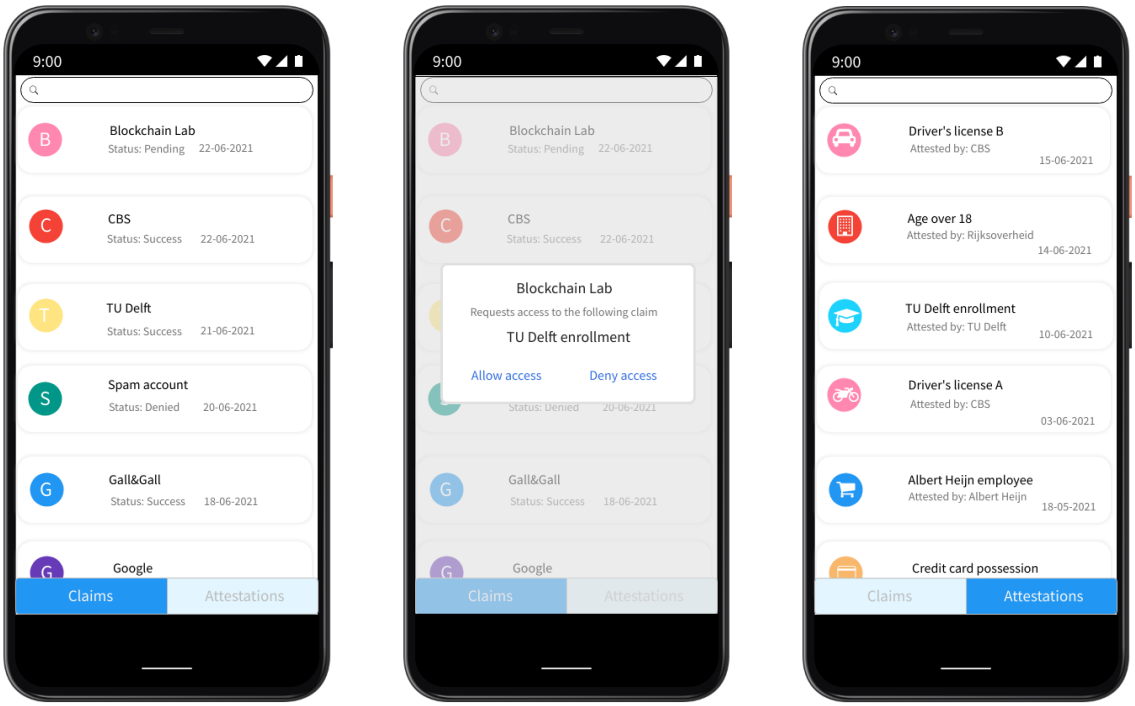
To make sure companies will take the step to adopt the SSI solution, it needs to be easy to integrate such that a developer does not take too much time to do it. Good examples of successful services that can be integrated into other applications are payment methods such as iDeal or PayPal. They have created APIs or libraries that can communicate with the payment provider. To make the developer usability as high as possible, a library has been designed as discussed in subsubsection 6.1.1.

7.2 User usability

91% of users of the internet know it is unsafe to reuse passwords, but 61% still admit to doing exactly that [1]. This indicates that security is not a priority for most users. They will only adopt an application if it adds to the user experience, which means the advantages should outweigh the disadvantages for a user. In order to minimize the disadvantages, the usability should be very high. [22] has defined the main necessities to create an application that users can easily utilize. These are learnability, efficiency, user retention over time, error rate, and satisfaction. In short, the learning curve should not be too steep, the process of using the application should be efficient and a user should not be able to make many mistakes. For our framework, that means that the actions should be explained clearly and concisely and the number of actions the user should take should be limited.

7.3 Usability review

Some mock-up user interfaces were created for this framework, as shown in Figure 5. These mock-ups include the claim and attestation registries that will reside in the Super App, and a survey was created to evaluate their usability. The participants range from elderly with little technological experience to developers of applications. The survey is based on [23] and starts by asking the participants about the expected goal of the application, based on the mock-up user interfaces.



(a) Claim overview (b) Notification for claim request (c) Attestation overview

Figure 5: Mockups for the user interface of the claim registry

Afterwards, the participants have answered some questions about the navigation and clarity of the separate screens. Finally, some questions of the survey handle the distinction and connection between separate screens. The results will be analysed in this section.

First, participants were shown the mock-ups without any context and were asked to predict the goal of the application. Most users were fairly close, with answers like 'password manager' and 'show facts about you, for example to get in a bar'. Next, participants were asked how they would try to find more information about a claim or an attestation and how they would navigate between the two pages. An open question was also asked to evaluate if the correlation between the pages is clear. As can be seen in Table 1, this was not the case. After speaking to the participants, it became apparent that this was due to the explanation of the application. Some users did not understand it fully, especially since English is not their first language. Therefore, the pages should be given a title and an option to gain more information.

Question	Correct	Incorrect
Application goal	73%	27%
More claim information	95%	5%
More attestation information	89%	11%
Navigation between pages	89%	11%
Correlation of pages	37%	63%
Meaning of notification	66%	34%

Table 1: Open question correctness

Furthermore, after explanations were offered about pages or the application itself, the clarity of the user interfaces was evaluated based on that explanation. The results can be viewed in Table 2. Based on this table, it is clear that the claim overview needs some improvements, which will probably affect the overall impression and clarity as well.

Participants were able to give suggestions to improve the user interfaces. First, many non-native English speaking participants did not know the meaning of the word 'attestation' and did not know the difference between claims and attestations. After this feedback, it seems better to rename the pages. Instead of claims, *requests* is a better option as this reflects the objective of the page better. Based on the responses of participants, attestations should be renamed to *Your data*, as many participants never used the word attestations in the feedback and used 'data' instead. Therefore, from now on, the terms *request* and *user data* will be used to indicate claims and attestations, respectively.

Other suggestions were to add buttons for users to get more information on the different sections on the pages. In the list of requests, users are missing information on what user data has been shared exactly. To make looking for certain requests or user data less complicated, participants want options for ordering them based on name, date, organization, or even category. In the request overview, the letters should be replaced for logos or icons as users associate those with organizations.

In the notification, participants are especially missing the reason that the organization is requesting data, so a short explanation or button to show this information should be added. Participants would also like an option to delay the choice,

Question	Unclear (1)	Not very clear (2)	Neutral (3)	Clear (4)	Completely clear (5)	Mean
Complete UI	0%	5,4%	24,3%	64,9%	5,4%	3,7
Claim overview	0%	16,2%	29,7%	35,1%	18,9%	3,6
Attestation overview	0%	5,4%	10,8%	56,8%	27%	4,1
Options on notification	0%	5,4%	13,5%	29,7%	51,4%	4,3

Table 2: Scale question answers

however this is not feasible. Requests have a short expiration time to make sure malicious users can not misuse them. Users will often need to answer requests based on actions they took, so the request can be denied if a needs more information first. This should be made clear to users in the extra information that will be added.

8 Responsible Research

To validate my research, it is important to reflect upon the proper acknowledgment of related work, reproducibility, privacy, and integrity. Related work is cited using the IEEE citation method. This does not only include research papers, but also white papers, technical reports, and GitHub pages.

8.1 Reproducibility

Proper citations are of significant value for the reproducibility of the work. As this paper focuses on designing a framework, the most important factors that should be reproducible are the design decisions that were made based on the literature.

The code for the framework has not been implemented yet. If it ever is, that should happen in an open-source fashion, such that it can easily be cloned and executed by anyone that wishes to review the implementation. For the research to be reproducible, it should also be tested with the same data and produce the same results. This can easily be done by cloning the code and running the application. The test data will not be completely identical, as the user gets assigned a public key based on their device. Still, the results should be similar enough to verify the functionality of the application.

8.2 Privacy

Identifying applications such as SSI solutions are dealing with sensitive data. Since the General Data Protection Regulation (GDPR)[5] went into effect in 2016, users have the right to request, delete or alter their personal data that is stored at an organization. The aim of SSI aligns perfectly with the GDPR as it aims to give users full control over their data, while still abstracting the technical details for usability.

8.3 Integrity

The results of the usability survey that was evaluated in subsection 7.3 have been summarized. To ensure data integrity, all the results can be requested by sending the author an email. This allows reviewers to make their own conclusions about the reviews and the product. The results are anonymous to protect the privacy of the participants of the survey.

[5]European Commission: GDPR

9 Conclusions and discussion

This research presented a universal solution for portability for verifiable claims in Self-Sovereign Identity applications with a focus on usability. Interoperability is reached by enabling communication with other applications. This ensures that all currently existing applications do not have to be replaced, but can integrate SSI in their application, which prevents a complete application infrastructure transition.

This framework is designed with a Public Key Infrastructure (PKI) because those are common in SSI applications and therefore the most universal solution. As the Super App has been built upon IPv8, it also includes a PKI that can be used for creating the framework. The keys are used to sign and encrypt messages which are sent between the Super App and the relying party to ensure security.

The usability of the framework has been evaluated using screenshots of the user interface. It has been evaluated by 39 potential users. From the survey, it became apparent that not all aspects of the interfaces are clear and some suggestions were made to improve them.

SSI solutions are currently not yet widely used. Some applications have relative success and offer services from several providers. Still, the optimal solution is a solution used by everyone and everything. To achieve this, the application needs to be universally applicable and effortless to use. Users will not trade their current applications without a good reason. This framework makes it easy for existing applications to use SSI without extra effort for users. The next step is to implement this framework in the Super App, such that it can hopefully reach its full potential as an SSI application.

10 Future work

The next step to take is to achieve full data portability. This can be done by using verifiable credentials in addition to verifiable claims. In a verifiable credentials architecture, PII is stored in a secure database that can also be reached by other services. An example of such a database is the Gaia data storage that BlockStack uses, which is a combination of services like Dropbox and Google Drive.

When a service wants to request data from the user, they send an authentication request. If the user decides to authenticate the application, they reply with a confirmation and a public key. This public key can be used to decrypt the PII in the data storage. This approach would also enable commercial applications like Twitter and Instagram to use the SSI solution. All the user data, including tweets, can be stored in the database that the user controls, but the application can still quickly request access to show the tweets on the timeline.

The claim portability framework can be used to create a data portability framework. The verifiable claim creation and

transmission can be used to send the authentication request and confirmation as well. The aspect that does need further research, is data storage. Even though the data in the storage is encrypted, the risk of failures and weaknesses should still be minimized.

So while this research paper presented a key step towards full data portability, data storage is still a missing link in the chain that needs to be researched in order to find the final solution.

References

[1] LastPass Enterprise, "The password exposé. 8 truths about the threats - and opportunities – of employee passwords," tech. rep., LastPass, 2017.

[2] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," March 2017.

[3] A. Mitchell and J. Smith, "Economics of identity. the size and potential of the uk market for identity assurance," tech. rep., The Open Identity Exchange / Ctrl-Shift, October 2015.

[4] Hiscox, "The hiscox cyber readiness report," 2017. Data retrieved from https://www.hiscox.com/documents/brokers/cyber-readiness-report.pdf.

[5] S. Morgan, "Cyberwarfare in the c-suite," 2021. Data retrieved from https://1c7fab3im83f5gqiow2qqs2k-wpengine.netdna-ssl.com/wp-content/uploads/2021/01/Cyberwarfare-2021-Report.pdf.

[6] Delft Blockchain Lab, "TrustChain SuperApp." GitHub, 2021. Retrieved from https://github.com/Tribler/trustchain-superapp.

[7] K. Wagner, B. Némethi, E. Renieris, P. Lang, E. Brunet, and E. Holst, "Self-sovereign identity: A position paper on blockchain enabled identity and the road ahead.," pp. 13–14, October 2018.

[8] P. Dunphy and F. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE Security Privacy*, vol. 16, 01 2018.

[9] M. Shuaib, S. Daud, and S. Alam, "Self-sovereign identity framework development in compliance with self sovereign identity principles using components," *International Journal of Modern Agriculture*, vol. 10, p. 2021, 05 2021.

[10] D. C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, "uport: A platform for self-sovereign identity," tech. rep., uPort, October 2016.

[11] The Sovrin Foundation, "A protocol and token for self-sovereign identity and decentralized trust," tech. rep., Sovrin, January 2018.

[12] The SelfKey Foundation, "Selfkey whitepaper," tech. rep., The SelfKey Foundation, September 2017.

[13] Civic Technologies, "Civic whitepaper," tech. rep., Civic, 2017.

[14] SITA, "Travel identity of the future," tech. rep., Sita and ShoCard, May 2016.

[15] M. Ali, R. Shea, J. Nelson, and M. J. Freedman, "Blockstack: A new decentralized internet," tech. rep., Blockstack, May 2017.

[16] M. S. Ferdous, F. Chowdhury, and M. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 1–1, 07 2019.

[17] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Computer Science Review*, vol. 30, pp. 80–86, 2018.

[18] Z. Diebold, "Self-sovereign identity using smart contracts on the ethereum blockchain," Master's thesis, University of Dublin, Trinirty College, Dublin, 5 2017.

[19] S. Goswami, "Scalability analysis of blockchains through blockchain simulation," Master's thesis, University of Nevada, Las Vegas, 5 2017.

[20] Eurostat, "Mobile internet access," 2019. Data retrieved from Eurostat Data Explorer (link).

[21] "IPv8 Documentation." readthedocs.io, 2021. Retrieved from https://py-ipv8.readthedocs.io/en/latest/.

[22] X. Ferre, N. Juristo, H. Windl, and L. Constantine, "Usability basics for software developers.," *Software, IEEE*, vol. 18, pp. 22 – 29, 02 2001.

[23] Y. G. Ji, J. H. Park, C. Lee, and M. H. Yun, "A usability checklist for the usability evaluation of mobile phone user interface," *International Journal of Human-Computer Interaction*, vol. 20, no. 3, pp. 207–231, 2006.