Towards a Disaster Resilient Self-Sovereign Identity

Kalin Kostadinov , Martijn de Vos , Johan Pouwelse
k.k.kostadinov@student.tudelft.nl , m.a.devos-1@tudelft.nl
Delft University of Technology

Abstract

1 Introduction

Every person on the Internet uses at least one digital identity. And service providers rely on them for building trust with their users. Unfortunately, the creators of the Internet have not designed a unified identity layer. Thus, service providers need to handle authentication and authorization themselves [2] which explains why every service has at least one identity management system. However, those systems control users' identities, so identity owners cannot administer their data.

In recent years, identity management has become a big concern for governments which has led to a large amount of research and regulations in the field [3]. There is a need for a novel identity management system, and its formal description stands in the middle of all the work [4]. It promises to not take control over an identity from its rightful owner and achieves this by satisfying the requirements for Self-Sovereign Identity [1]. SSI allows every identity holder to store and manage their data. For that, they need to use resources under their jurisdiction.

There are already several implementations that cover part of SSI's properties [8], and they have matured over the past couple of years. However, the biggest obstacle which prevents them all from going mainstream is the problem of adoption add reference to the adoption paper. Self-Sovereign Identity management systems have one of two problems in this regard. Part of them relies on global blockchains that contain the entire transaction history of all users. In this case, global consensus is a must that prevents such systems from operating in several situations. For example, in an offline setting or when transaction times need to be as low as possible. Other SSI managers either use local blockchains or do not use a blockchain at all. Such systems are fully distributed and need no global consensus, thus solving the problem of the former group. However, they do not employ any mechanism for disaster resilience, and in case users lose access to their digital identity, they cannot recover it. All in all, availability suffers.

There is a need for a solution to the data resilience problem of fully distributed SSI management systems. It will allow those systems to outweigh the other SSI managers when it comes to service availability. Thus, data resilience as a sub-problem of availability is a research area that is worthwhile exploring.

The following research question will be at the center of this work: ***How to make fully distributed Self-Sovereign Identity management systems disaster resilient?***

The Delft Blockchain Lab develops one of the Self-Sovereign Identity management systems, called IPv8 [6]. It is arguably the most sophisticated SSI management system. However, the issue with IPv8 is that it does not offer long-term data resilience, thus not offering a mechanism for recovery from identity loss. Every user has its blockchain, called TrustChain [5], for managing their identity. And Trustchain allows IPv8 to work as a fully distributed system. The idea behind this design decision is that users have more control over their own identity if they are the only ones physically possessing their data blocks.

Mobile applications are the most effective way of hosting an identity management system like IPv8. However, mobile devices are not reliable enough. Thus, it is not clear how users are supposed to recover their identities when access to them is lost. IPv8 falls within the group of SSI managers that use local consensus. Consequently, it suffers from the problem this paper is trying to solve. That is why I am using IPv8 as a platform to develop a solution for my research question.

Since I am using IPv8 as a base for improvements, I have explored its implementation in Kotlin for the super app [7], which the Delft Blockchain Lab is currently working on. My goal was to assess the application and find a suitable approach for integrating my implementation.

First, this paper will define two possible solutions and argue about which is the better one. Then, it will introduce the design of an algorithm used for solving the problem. Later, it will lay out the technical details and an example use for the solution. In the end, it will discuss the reproducibility of the contribution, derive conclusions, and propose some ideas for future work.

2 Problem Description

It is clear that to make a system resilient to data loss, a protocol adding redundancy is in need because identities have to be in at least two separate locations. Redundancy, however, adds some complexity and overhead to the system. Also, it calls for a caching mechanism that, in an offline setting, allows temporary storage of transactions before synchronizing them with other system deployments. And if there are multiple nodes having control over the same identity, there needs to be a mechanism for access revocation. Two ideas emerge from these observations.

The first one is to keep the SSI management system on a master server, controlled by the identity holder. A central node that is under the jurisdiction of the identity owner will add some unwanted overhead. Also, there needs to be a caching mechanism. It is mandatory for the storing of offline transactions before committing them to the central node. However, the benefit is that users will easily revoke remote access, quickly transfer control to other devices and reliably restore lost identity access.

The second idea is to reproduce the blockchain from the knowledge of other users about the lost identity. Blockchain recreation, however, increases complexity because there might be some offline users during the rebuilding process. Also, some might not be honest about previous transactions, and others might not even exist anymore. The benefit here is that there is no need for a caching mechanism, but the revocation will only be possible through peers, which ignore the revoked node. Privacy is also a concern in this instance. It is not desirable to keep identity information, even if it is encrypted, on untrusted nodes.

After an evaluation of both solutions, the first one looks more suitable for providing data resilience. The reason is that a central node might run on a machine connected to the wall. Such a device does not rely on battery size and network coverage. Thus, compared to a smartphone, it seems to have unlimited resources.

The second solution also needs an algorithm that continuously tries to make data backups available. It will add an enormous amount of excess traffic and waste valuable resources. Phone storage devoted to supporting the backup system will become unusable to the owner of the mobile device. Another issue is the need for a revocation mechanism. It will rely on a distributed hash table algorithm for gossiping information about mobile devices with revoked access to a specific identity.

In conclusion, to implement the first solution, the following questions need to be answered:

- How to allow transactions when there is no access to the central node?
- How to deal with cached transactions when they do not get committed to the central node and get lost?
- How to make cached transactions legally valid?

3 Caching Legally Valid Transactions

4 Implementation and Results

5 Responsible Research

6 Discussion

7 Conclusions and Future Work

A further development that goes beyond the goals of this research project will be the creation of emergency access ”terminals” that will be available at border control, for instance. They will allow someone access to their identity manager with restricted controls if their other SSI managers are not available. Those emergency ”terminals” should only allow for verification of attestations.

References

[1] Christopher Allen. The path to self-sovereign identity. 2016.

[2] Gergely Alpár, Jaap-Henk Hoepman, and Johanneke Siljee. The identity crisis. security, privacy and usability issues in identity management. 2011.

[3] European Commission. Regulation (eu) 2016/679 of the european parliament and of the council. 2016.

[4] Md Sadek Ferdous, Farida Chowdhury, and Madini O. Alassafi. In search of self-sovereign identity leveraging blockchain technology. *IEEE Access*, 7:103059–103079, 2019.

[5] Pim Otte, Martijn de Vos, and Johan Pouwelse. Trustchain: A sybil-resistant scalable blockchain. *Future Generation Computer Systems: the international journal of grid computing: theory, methods and applications*, 107:770–780, June 2020.

[6] Quinten Stokkink, Dick Epema, and Johan Pouwelse. A truly self-sovereign identity system. 2020.

[7] Tribler. Trustchain super app.

[8] Dirk van Bokkem, Rico Hageman, Gijs Koning, Tat Luat Nguyen, and Naqib Zarin. Self-sovereign identity solutions: The necessity of blockchain technology. 04 2019.