Proof of Concept for a Blockchain-based Self-Sovereign Identity system

Blockchain Engineering CS4160

Mart Meijerink s1097296
Ruben Starmans 4141792
Mathijs Hoogland 4237676
Konrad Kleeberger 4748476
Santhos Baala Ramalingam Santhanakrishnan 4740270

February 2, 2018

Abstract

This technical report describes the development of a proof of concept of a blockchain-based, self-sovereign identity system. The system allows users to make claims about their identity, get an attestation for these claims from an authority (the government for example), and then use their attested claims to prove they are allowed to use a service offered by a provider.

The system makes use of blockchain technology to publish the attestations made by authorities. An authority publishes the public parameters of a zero-knowledge proof (ZKP) and the key of the claim it is attesting. By publishing this on the blockchain, the authority acknowledges the user has the claimed attribute. After which the user can proof the ZKP to a provider, which verifies the proof with the information on the blockchain.

1 Introduction

Traditional identity management systems are centralised. Users need to trust companies to handle their data correctly. Facebook, for example, has a really large database of users who have all entered their personal information which is then uploaded to servers managed by Facebook. But all this information gathered in a centralised manner gives them too much power and might result in a violation of privacy guidelines. The rise of blockchain technology has resulted in decentralisation of many domains, also decentralised identification schemes are actively being developed.

In this report, we describe a proof of concept self-sovereign identity system in which the identity management and thus the power is in the hands of the individual. We first introduce the concept of self-sovereign identity in Section 2. This is followed by a description of the architecture designed to implement a proof of concept in Section 3. The system relies on Blockchain technology to distribute identity attributes and in Section 4 the underlying structure of our Blockchain is described. Section 5 describes how the proof of concept preserves the privacy of users by implementing zero-knowledge proofs (ZKP). In Section 6, the conclusion on the implementation of the proof of concept can be found, as well as pointers for future work.

2 Self-Sovereign Identity

This section describes the concept of and the ideas behind self-sovereign identity, what it is, how it can be used, and the concepts implemented in the system.

2.1 Concept

Identity is a representation of certain attributes of the whole independent existence. The digital identity could be online Accounts which are formed by one person or digital certificates which are linked to one. Since the beginning of the world wide

web, there were attempts to unify certain accounts around the web under one identity. This kind of management is also called identity management. The problem with these attempts in the past was that there was always a central trusted authority managing the identity like in the 'login with Facebook' example. With blockchain technology, it is possible to have decentralised management. As the blockchain is built with cryptographic methods it also provides data protection and new ways of sharing details like Zero Knowledge Proofs. Furthermore, it provides low entry barriers as everyone can participate. This form of identity is called self-sovereign identity [2].

2.2 State of the art

There are currently two projects who aim to deploy self-sovereign identity globally. Sovrin uses a permissioned ledger approach which means that there are many organisations which sign a contract with the Sovrin foundation and therefore can manage the identity of the user. The user can freely choose which organisation he trusts with his data and is able to change the organisation at any time [4]. Uport, on the other hand, is built on the permissionless Ethereum blockchain. Ethereum is used for signing claims (attributes) about an identity or performing actions. The actual data can be saved off the chain [5].

2.3 Principles of Self-Sovereign Identity

There are several principles of a self-sovereign identity, as explained in [1]. The *existence* of the papers prototype is given by the sheer existence of attributes which represent a real identity. The users have full *control* over their identity as they can only share the attributes they want to share. Using the Zero knowledge proof also guarantees that even the shared data is not revealed. The *access* to the data is in the power of the users as the secret and the claims are stored locally and only the public parts are published on the blockchain. Despite that, there is no data stored in a proprietary format. The used algorithms for encryption are publicly available what guarantees *transparency* of the process. Even when an authority shuts down the user has *portability* to move on to another authority to claim the attributes. The prototype includes an interactive accepting mechanism. This keeps the *consent* in the hand of the users. Furthermore, the system has *minimisation* implemented as the claims only share as little information as possible about the user.

3 Architecture

This section describes the system architecture's core setup, as shown in Figure 1 in Appendix A. The core exists of three main building blocks, namely 1) Identity, 2) Claim, and 3) Attestation. Besides these is a Service, bundling one or more claims. The proposed architecture could be used by authorities to transition from physical documents to digital attestations as it recognises the current role of governments in being a trusted provider in the identity business. However, it also offers the possibility for third parties to become authorities for certain claims. A bank, for example, might be perfectly able to attest a user's financial solvency. This architecture recognises this opportunity, but the proof of concept does not yet implement this possibility.

3.1 Identity

An Identity models a person or organisation. An identity can take three roles. Being, 1) a User, modelling a natural person able to make claims about his identity; 2) an Authority, modelling an authority which can attest Claims; or, 3) a Provider, offering services which users can request.

Users can make claims about their identity. These claims can be *e.g.* "HAS_DRIVERS_LICENSE" or "18+". As long as these claims are true or false statements. Before a claim can be used, it has to be verified. To this end, a user requests an Attestation from an authority. If the authority gives out the attestation, the user receives the parameters to prove his claim. Now, the user can prove his claim when needed for a service it wants to use from any provider.

Upon receiving a request to attest a user's claim, the authority will vet that claim against its records. In case of a driver's license for example, the authority will confirm that the requesting user actu-

ally possesses a driver's license. For a user claiming to be over eighteen years old, the authority will confirm this based on his birthday. When a claim is correct, the authority will generate a zero-knowledge proof (ZKP) and share the necessary parameters with the user and publish the public parameters on the blockchain linked to the attested claim. Zero-knowledge proofs are explained in depth in Section 5.

Providers are companies offering certain services or selling products. For these services it might be needed the user needs certain attributes. Like a driver's license in case the user wants to rent a car, or to be over eighteen when buying alcohol. Therefore, the provider answers the request of a user with a request to provide a proof for the necessary claims. The user creates a zero-knowledge challenge-response string to prove his attributes and sends this along with the location of the public parameters of the ZKP on the blockchain. The provider will verify the received proof of the user when it acknowledges the authority.

3.2 Claim

A Claim models an attribute of a user. A claim is something the user possesses and needs to be acknowledged by an authority. Without an attestation, a claim is worthless as it cannot be verified. The system models a claim as a true/false statement. This way it can be proven using zero-knowledge proofs.

3.3 Attestation

An Attestation models the attestation made by an authority. This authority could be the government for instance. The authority knows the underlying attribute of the user with which it can validate a claim. For example, upon receiving the request to attest a user is over eighteen, the authority will verify this based on the user's birthday. When the user is over eighteen, the authority issues an attestation. This attestation is a ZKP of which the authority publishes the public parameters and the key of the attested claim. After publishing, the user receives the secret parameter of the ZKP along with the block hash where the public parameters can be found on the blockchain.

From this point forward, the user can create a proof of his claim and send this together with the block hash to a provider. The provider can look up the attestation on the blockchain and verify the proof using the public parameters. When the verification of the claim succeeds, the provider can provide the requested service to the user.

4 Blockchain Structure

This section describes how we designed our blockchain, what our design considerations were, and how we implemented our blockchain in our final deliverable.

4.1 Design

We used the code presented in the blog post *Learn Blockchains by Building One* [7] as the starting point for our design. This is a blockchain implemented with `Python`, using `HTTP` requests to communicate with all nodes in the network. We decided to go for this minimalistic approach, since we wanted to focus on our self-sovereign identity implementation.

4.2 Design considerations

Proof-of-work Our proof-of-work consists of finding a hash value that ends with 5 zeros. We choose this number since it enables us to mine blocks relatively quick (roughly 10 seconds), and therefore can demonstrate the functionality of our application.

Message space The message space of our blockchain has no constraints. Authority nodes use it to store the ZKP, but in theory every data type can be stored.

Mining reward Nodes are currently not incentivised to mine blocks. Since it is quite easy to find a proof-of-work, this is currently not necessary. Every node is able to mine and publish messages to the blockchain.

4.3 Implementation

We designed a demo consisting out of 3 nodes (a normal user, an authority, and a provider). Docker-compose is used to spin run these nodes separately and enables them to communicate with each other.

5 Zero Knowledge Proof

In order to prove that the user indeed possesses the claims attested, a naive way for the provider would be to have the authority authorise the user and send the approval back to the provider. But this protocol severely limits the capabilities of the system by requiring all the parties to be online[1], centralises the control to the authority (which can be compromised), has the risk of having to pass the identity over the wire to the authority and is not scale-able. To address the problems mentioned previously, a mathematical tool called zero knowledge proofs (ZKP) [6] is used, through which the provider can assert that the user indeed possesses the attestation with a very high probability. The blockchain is used as the medium for publishing the attestations, inheriting its scalability and security properties which make the attestations free from tampering in addition to allowing the parties to be offline and temporally independent. The implementation for the zero knowledge proof is a separate utility which classes can simply import. Because of this it can be swapped out for a different approach very easily. In the project, we have used the Schnorr's Identification Protocol [6] based on finite groups in elliptic curves, implemented using the PETLIB Python library [3].

5.1 Generating the attestation

At the authority's end, after the claims are verified, a random[2] elliptic curve G is chosen, with a generator g, of prime order o. The curve was chosen randomly because the curves were all proven to be secure [3]. Then a random secret value x is chosen within o. The transformation $h = x * g$ and the curve number are published onto the blockchain and x is securely send to the user along with the block address of the attestation. All parties look up the known values g and o when G is given and also agree that the $*$ operation is multiplication.

[1] Multiple authorities could be involved for various claims.

[2] From the set of curves provided by PETLIB for speed and security.

5.2 Proof generation and verification

At the user's end, with the knowledge of G, h and x, corresponding to an attestation, a commitment $r = k * g$ is generated, where k is a random value in the group. Also, the parameters r, g, and h are digested using SHA512 as e, and a signature is created according Equation 1.

$$s = (k + (x * e)) \mod o \tag{1}$$

The commitment r, the digest e and the signature s are sent along with the block hash address to the provider as the ZKP of the attestation.

In order to verify the proof's correctness, the commitment r is simply reconstructed at the provider end from s, e and the block hash is looked up for the values of G and h. It is then asserted that the equality in Equation 2 is satisfied.

$$r = ((s * g) + (-e * h)) \tag{2}$$

What makes the ZKP safe is the non-triviality of the problem. To increase the difficulty, the provider could request for proof multiple times from the user, have the authority re-attest the claims periodically, and also the system could be extended to assign reputation values to the authority.

6 Conclusion

In this report we have described a self-sovereign identity system which is built from the ground up. For this purpose a simple blockchain structure was built in Python to accommodate this system. We have also elaborated on the structure of the system which is simple and to the point. The self-sovereign identity system is a working concept which allows a user to request attestation of a certain attribute by a recognised authority. The public parameters of the zero knowledge proof linked to this attestation can then be published onto the blockchain. Next, a provider can verify a user has a certain attribute without that user giving away any additional information.

The system is very versatile in its maintenance as the used technologies can be adapted very easily. For example, the blockchain on which the attestations are published, can be exchanged for another type of blockchain. The type of encryption and also the creation and verification of the zero knowledge proof can be swapped out as these are separated modules in the structure of the system.

6.1 Future Work

An addition we would like to build in the future is the implementation of identification through biometrics. This was out of scope for this project and now anyone that has access to the machine with the secret value of the attestation can use it to prove corresponding attribute. The addition of biometric access control could eliminate this problem.

Future work should also focus on support of multiple authorities. The architecture supports this extension, but further research is necessary to establish a viable and scalable way to implement multiple authorities. A possibility would be to set up a public key infrastructure (PKI) inspired scheme.

References

[1] C. Allen. The Path to Self-Sovereign Identity. `https://www.coindesk.com/path-self-sovereign-identity/`. Accessed: 2018-01-24.

[2] D. Baars. Towards Self-Sovereign Identity Using Blockchain Technology. Master's thesis, University of Twente, 2016.

[3] G. Danezis. Petlib documentation. `http://petlib.readthedocs.io/en/latest/`. Accessed: 2018-01-16.

[4] S. Foundation. Sovrin Provisional Trust Framework, June 2017.

[5] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. uPort: A Platform for Self-Sovereign Identity, February 2017.

[6] N. P. Smart. *Cryptography made simple*. Springer, 2016.

[7] D. van Flymen. Learn blockchains by building one. `https://hackernoon.com/learn-blockchains-by-building-one-117428612f46`. Accessed: 2017-12-16.

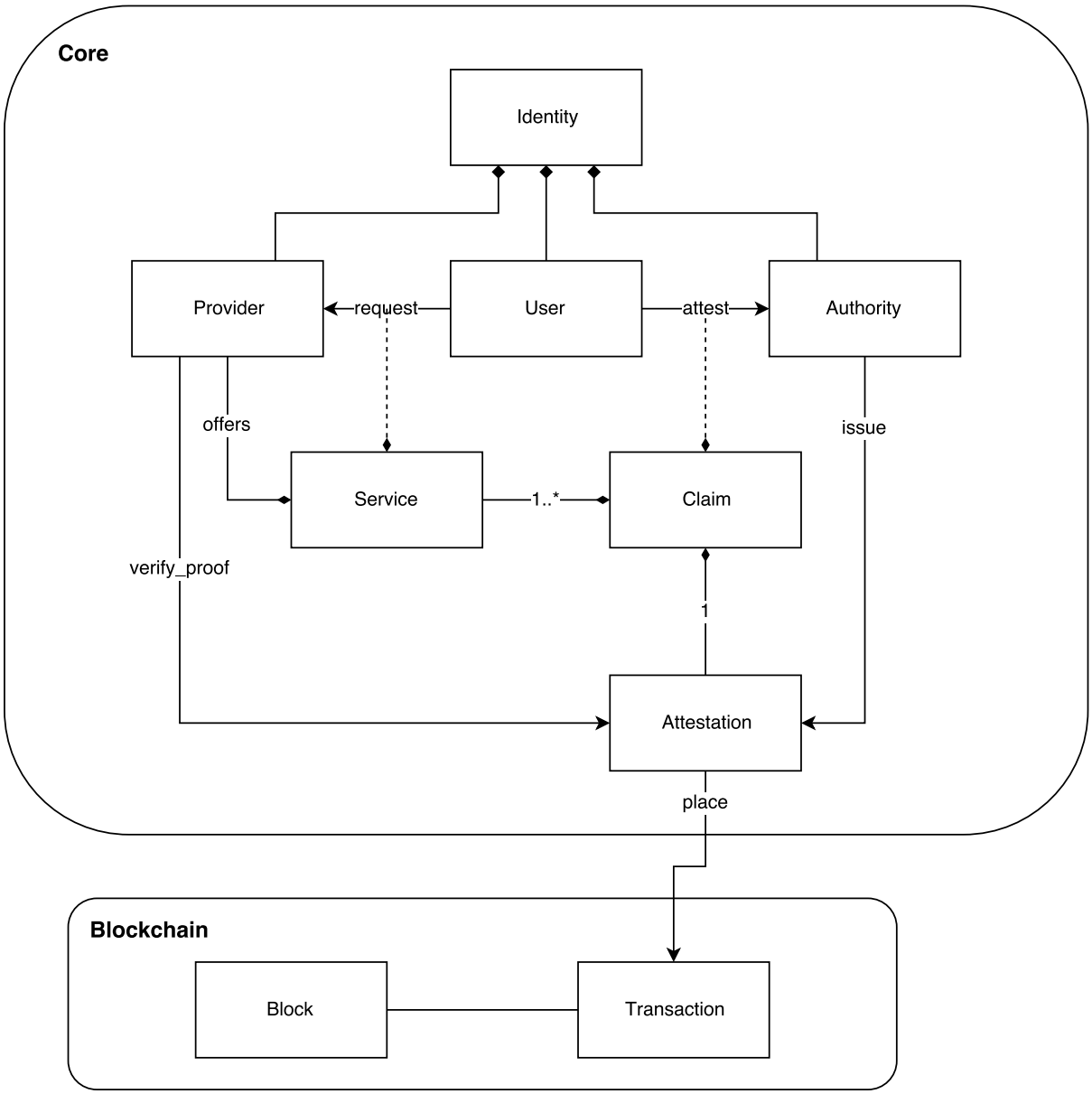

Figure 1: Self-sovereign identity system's architecture diagram

A Architecture Diagram

Figure 1 shows the architecture diagram of the self-sovereign identity system.