Computer Science Review 30 (2018) 80–86

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Survey

A survey on essential components of a self-sovereign identity

Alexander Mühle, Andreas Grüner, Tatiana Gayvoronskaya, Christoph Meinel

Hasso Plattner Institute, Potsdam, Germany

ARTICLE INFO

Article history:
Received 30 April 2018
Received in revised form 8 October 2018
Accepted 15 October 2018
Available online 25 October 2018

ABSTRACT

This paper provides an overview of the Self-Sovereign Identity (SSI) concept, focusing on four different components that we identified as essential to the architecture. Self-Sovereign Identity is enabled by the new development of blockchain technology. Through the trustless, decentralised database that is provided by a blockchain, classic Identity Management registration processes can be replaced.

We start off by giving a simple overview of blockchain based SSI, introducing an architecture overview as well as relevant actors in such a system. We further distinguish two major approaches, namely the Identifier Registry Model and its extension the Claim Registry Model.

Subsequently we discuss identifiers in such a system, presenting past research in the area and current approaches in SSI in the context of Zooko's Triangle. As the user of an SSI has to be linked with his digital identifier we also discuss authentication solutions.

Most central to the concept of an SSI are the verifiable claims that are presented to relying parties. Resources in the field are only loosely connected. We will provide a more coherent view of verifiable claims in regards to blockchain based SSI and clarify differences in the used terminology.

Storage solutions for the verifiable claims, both on- and off-chain, are presented with their advantages and disadvantages.

Contents

1. Introduction

Blockchain technology has experienced tremendous hype in recent years and is touted as a transformative evolution in distributed systems [1]. Satoshi Nakamoto is seen as father of the technology for introducing *Bitcoin: A peer-to-peer electronic cash system* [2]. By applying the concept of trustless timestamping proposed by Haber and Stornetta [3] to a decentralised setting and combining it with a chain of Proof-of-Work [4,5] the so called Nakamoto consensus protocol was established.

The computational resources invested in the Proof-of-Work solutions are equivalent to votes on the correct version of the blockchain, so as long as more than 50% of the computational resources are in control of honest nodes, an eventual consistency can be achieved [6].

This decentralised consensus protocol has seen application in numerous fields, one of them identity management.

The management of identities has also experienced increased interest due to the ever growing need of digital identities, as a large part of peoples lifes is spent online, interacting with services.

E-mail addresses: alexander.muehle@hpi.de (A. Mühle), andreas.gruener@hpi.de (A. Grüner), tatiana.gayvoronskaya@hpi.de (T. Gayvoronskaya), christoph.meinel@hpi.de (C. Meinel).

https://doi.org/10.1016/j.cosrev.2018.10.002
1574-0137/

Security	Controllability	Portability
Protection	Existence	Interoperability
Persistance	Control	Transparency
Minimisation	Consent	Access
		Portability

Fig. 1. Christopher Allen's Ten Principles of Self-Sovereign Identity summarised by the Sovrin Foundation [9].

A digital identity can be simply described as a means for people to prove electronically that they are who they say they are and distinguish different entities from one another.

Although the term "*Self-Sovereign Identity*" (SSI) is still only loosely defined, a few key properties of the concept have emerged. In essence it is an identity management system which allows individuals to fully own and manage their digital identity. The World Wide Web Consortium (W3C) working group on verifiable claims states that in a self-sovereign identity system users exist independently from services [7]. This highlights the contrast to current identity management which either relies on a number of large identity providers such as Facebook (Facebook Connect) and Google (Google Sign-In) or the user has to create new digital identities at each individual service provider.

Christopher Allen proposed *Ten Principles of Self-Sovereign Identity* [8] which laid out the requirements for a system implementing the self-sovereign identity concept. These Principles were further grouped into the three categories *security*, *controllability*, and *portability* in a whitepaper by the Sovrin Foundation [9] as pictured in Fig. 1.

Essentially *security* can be boiled down to the protection of personal user data and the limiting of data exposure to the minimum required to fulfill a function. Additionally a persistent identity was named as a security requirement. Persistence in this context however should not contradict a "right to be forgotten" according to Allen. This right to be forgotten could also be grouped into the *controllability* category as both *control* and *consent* should extend to the removal of the identity not only the creation and access.

Another essential requirement for an SSI system is the portability of the identity. Allowing the user to use their identity wherever they want and being independent of any particular identity provider.

Although there are a large number of projects and initiatives concentrated on Self-Sovereign Identity, both the terminology and understanding of architectures differs widely.

New innovations come primarily from private ventures or individuals volunteering in working groups. While this leads to a lot of interest in the wider public, the documentation of such ideas is either very practical or only for advertisement purposes limiting their scientific usefulness.

This paper's objective is to give an overview and deeper understanding of the concept of SSI as well as the current state of the art. For this purpose we will look at four basic components: identification, authentication, verifiable claims, and attribute storage, needed in a Self-Sovereign Identity system.

Before we start going into detail about the different components we will first provide a high level overview of the Self-Sovereign Identity (SSI) Architecture. After this general overview in Section 2 we will present the essential components of such a system, starting with the identifier to be chosen for identities in the system in Section 3. Further the authentication of the identity will be discussed in Section 4. The concept of verifiable claims and their integral role in the SSI system as well as the possibility for reputation systems will be reviewed in Section 5. For privacy and scalability considerations we will also discuss storage approaches for use in a Self-Sovereign Identity system in Section 6.

2. Self-Sovereign identity architecture

In contrast to most previous identity management systems where the service provider was at the center of the identity model, SSI is user centric. In Fig. 2 the relation between the different actors of the system can be observed. The claim issuer issues (at least part of) the identity by attesting to certain attributes of the user. This identity is controlled by the user himself. Any relying party that needs to identify the user will be presented with the parts of the user controlled identity relevant to him. In order to accept the identity, the relying party needs to have a trustful relationship with the claim issuer.

The basis of this new architecture type is the distributed ledger of the blockchain. In Fig. 3 the relation between the different components of a typical SSI architecture are laid out. The blockchain acts as replacement for the registration authority in classic identity management systems. In this paper we will call this blockchain function the *identifier registry*. Here the pairing of **identification** and **authentication** is maintained. The identifier as well as the **verifiable claims** are directly managed by the user.

The identifier is tied to the specific user by use of an authentication method such as asymmetric cryptography. By establishing a pairing of identifier and public key on the blockchain the identifier can be verified by anyone reading the blockchain by posing a challenge to the user himself or a delegate of the user.

A distinction can be made between subject and holder in some cases, i.e guardians to underaged individuals or attorney client relations. In the following we will, for simplicity, assume that the holder is indeed the subject of any claims and will refer to him as user.

The actual identity claim is stored in the user controlled **storage**, typically off-chain for privacy considerations. The relying party, also called claim-verifier, can then compare the publicly available identifier with the identifier in the claim that has been handed to him by the user. After authenticating the user with the authentication method presented in the public blockchain, the claim itself can be verified and accepted or rejected by the relying party.

We will describe this architecture as *Identifier Registry Model*. A very popular competing model can be described as *Claim Registry Model*. In that model the blockchain not only functions as a registry for the identifiers of an identity but also to hold the cryptographic fingerprints of all the associated claims of an identity. This model can be seen as an extension to the Identifier Registry Model.

In this process no information about the user has to be stored at either the issuer or the verifier. Only the trust between the issuer and verifier has to be established beforehand. As described in this section, the SSI architecture relies on the mapping of an identifier to a specific authentication method that is recorded on the blockchain. In the next two sections we will discuss how this identifier and its namespace is chosen as well as the authentication methods used.

3. Identification

Bryce Wilcox-O'Hearn published a widely cited article on namespaces in computer systems in 2001. In it he laid out what is now known as Zooko's Triangle [10]. According to his assessment it was impossible (or highly unlikely) that someone would be able to design a system in which identifiers could be chosen in a distributed fashion but at the same time being both secure and human-readable (see Fig. 4).

In this context distributed means without the need for a central registration and verification process, while secure refers to the identifiers being securely unique (collision free).

The identifiers that are presented in this section can be grouped into three different categories. Firstly the identifiers based on

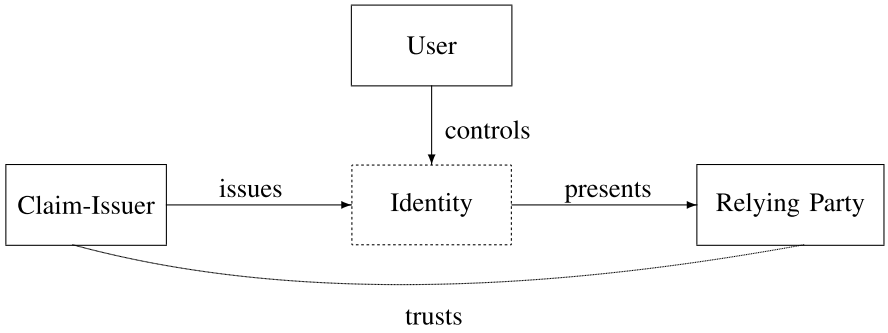

Fig. 2. Self-Sovereign Identity Actors.

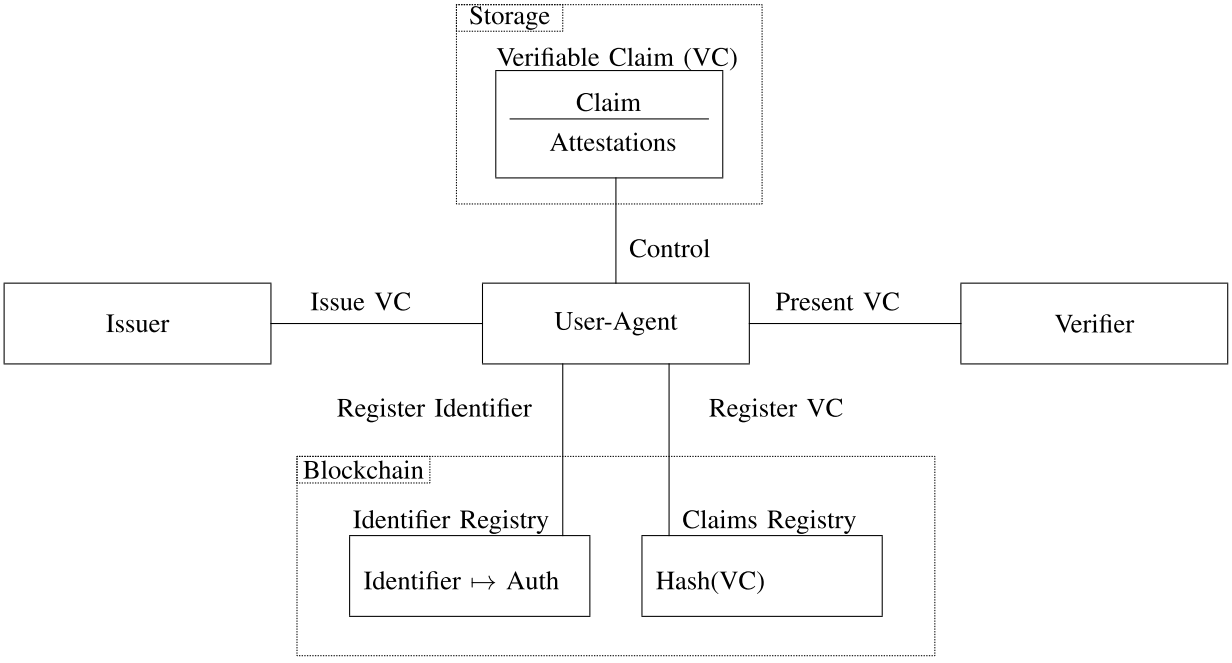

Fig. 3. Self-Sovereign Identity Architecture.

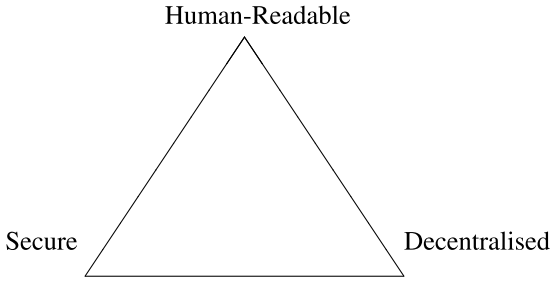

Fig. 4. Zooko's Triangle.

random number generation that rely on probabilities to avoid collisions. Secondly the centralised identifiers that utilise a registration authority in order to assign identifiers and prevent collisions. Finally we will discuss how the blockchain technology can help merge the best aspects of both these approaches.

Already in the 1980s the need for a globally unique identifier became apparent [11]. The Universally Unique Identifier (UUID or GUID) [12] does not require a central registration process but rather lets users generate their own identifiers, therefore partially fulfilling the decentralisation requirement formulated by Zooko's Triangle. In the UUID versions 1 and 2 the uniqueness is guaranteed by including node specific information such as the users MAC address which are, unless tampered with, uniquely assigned by the manufacturer of the network card and the IEEE registration authority [13]. This means in those versions there is still a centralised component while version 4 is using large random or pseudo-random numbers to avoid collisions. There are a total of 2^{122} possible version 4 UUIDs making a collision, assuming no implementation errors, highly unlikely to the point that it can be ignored [14]. This means UUIDs can be considered secure in this context and do not require a central registration authority. Apart from being non-human readable in the sense that no human could realistically remember specific UUIDs, they are also not completely decentralised as the verification process of key–value pairs using UUIDs typically requires a trusted third party for verification.

This is where public/private key pairs hold a significant advantage over UUIDs as identifiers in an SSI. In contrast to a UUID a public/private key pair would not require a trusted third party for verification as it is self-authenticating.

A distinction between self-authenticating and non-self-authenticating key–value pairs helps in understanding Zooko's argument. Self-authenticating schemes such as secure hash algorithms or public/private keypairs can create key–value pairs collision free (to current knowledge) and verify them without third party input but are typically non-human readable identifiers.

In non-self-authenticating schemes however there needs to be trust placed in a third party, assigning and verifying the name–value pairs.

X.509 is a commonly used public key certificate standard, utilised i.e for TLS/SSL [15]. Although the public–private key pair used in a X.509 certificate is self-authenticating, the mapping of a human readable *Distinguished Name* to a specific public key is not. For this mapping a centralised *Certificate Authority* has to be trusted to correctly assign and store the pairing of name and public key. This centralisation carries significant risk though. Either through attacks [16] or coercion [17] the central authority can be compromised.

However, even more decentralised solutions such as PGP [18] that do not rely on central entities for verification defacto utilise a quasi-centralised approach to assign human readable identifiers by using email-addresses. These are issued by a number of centralised providers that ultimately rely on the Internet Corporation for Assigned Names and Numbers (ICANN) to assign domain names without collisions.

Up until this point, Zooko's Triangle hypothesis held up. Either identifiers were not human readable or part of the decentralisation requirement was not fulfilled. From 2011 on a number of name services on the blockchain appeared, "squaring" Zooko's Triangle. With the distributed ledger technology it is possible to choose a human-readable identifier in a decentralised fashion as well as assign and verify name–value pairs without third party input.

In contrast to previous decentralised human-readable namespaces (i.e. as initially used in Freenet [19]) that were unsafe, the consensus protocol of the blockchain and the global view of the system can guarantee that once a name–value pair has been established it cannot be changed without the correct authentication and most importantly the same identifier cannot be assigned more than once. As there is no central authority assigning name–value pairs however, there need to be other mechanisms.

The first name service built on Bitcoin called Namecoin [20] as well as a later competitor Emercoin [21] used first come first serve logic to assign name–value pairs. This policy however causes problems such as squatting of names, which was further escalated by the lack of centralised control. Kalodner et al. found in their study of the Namecoin namespace design that of the 120,000 registered domain names, only 28 were not squatted or had non-trivial content [22]. They argue that because the names are human readable they are naturally scarce and will therefore hold some market value compared to the essentially infinite non-human readable identifiers such as hashes of keys or the public key to a private key.

Carl Ellison stated in his 1996 paper on *Establishing Identity Without Certification Authorities* that "*it is clear that there is no such thing as a universal, global name space with names meaningful to all possible users and that there never will be*" [23]. Ellison reasoned that there are simply too many names for a human to remember and attach meaning to.

These assessments were utilised by the Ethereum Nameservice (ENS) [24] which implemented a decentralised bidding process to reduce the problem of squatters.

The Self-Sovereign Identity system uPort [25] uses an Ethereum smart contract address as persistent identifier for a users identity. The address is derived from the public key of the creator of the smart contract. Since this identifier is non-human readable, Christian Lundkvist of uPort sees ENS as a viable naming layer to map the non-human readable uPort ID to a human readable address [26]. Blockstack [27] similarly uses its blockchain Name System to implement a naming service with human readable names that a Blockstack identity can be linked to for their system.

W3C decentralised identifiers [28] can be seen as an even higher level naming scheme, similar to URNs. They resulted from an effort by a number of working groups investigating decentralised name systems. A decentralised identifier (DID) is comprised of a scheme as well as a method and method specific identifier. The method closely resembles the namespace component of an URN. Each distinct blockchain or rather each identity registry (there can be multiple per blockchain, i.e uPort, Civic [29], SelfKey [30] all operate on Ethereum) constitutes its own namespace while the blockchain specific identifiers such as a uPort ID or ENS name specify the actual identity addressed by the DID. An example for such a DID path would be: did:examplechain:123456789

The Decentralised Identity Foundation is developing a universal resolver for these DID paths. Currently Sovrin [31], Bitcoin [2], Blockstack [27], uPort [25], Interplanetary Identifiers [32], and Veres One [33] are supported by the resolver with implemented drivers. The resolver uses the method type to decide which driver to use and uses the method specific identifier to resolve to the DID document stored on the specified Blockchain. The DID document or its equivalent in other systems is the key to the decentralised identity.

In it the authentication method is defined to bind the specified identifier to an identity that is in control of a secret key or other data used in the authentication.

4. Authentication

In a Self-Sovereign Identity system authentication is typically done with the use of a public/private key pair where the public key is stored as value of the identifier on the blockchain. This concept has been described as *Decentralised Public Key Infrastructure* [34,35]. Thanks to the zero knowledge proof properties of the asymmetric cryptography it is possible to prove that a given user is indeed in control of the identity with the public key stored on the blockchain. Most popular Self-Sovereign Identity systems use a asymmetric cryptography authentication protocol.

This poses the question of how the user should hold the private key associated with his key pair. Blockstack uses probably the simplest solution where the keys are stored with the device that the identity was created on and the user himself is responsible for key recovery and mobility. To make this process somewhat more usable mnemonic phrases, typically of 12 words, are used as seed to generate the keys. Using those phrases it is possible to recreate the private key, reducing the effort needed to move keys from one system to the other.

The most used solution in the space currently however is utilising smartphones for key storage. This has the advantage of being more portable than other solutions. The challenge that is being posed by the relying party is communicated to the smartphone via a QR code displayed on the login page and the response directly sent from the smartphone to the designated endpoint. This visual communication removes the need for physical connections and therefore hardware support that would be needed for alternative mobile solutions such as SmartCards. David Chadwick already stated in 1999 that "*smart cards are beneficial in some scenarios [...] in some user environments, the costs and inconveniences clearly outweigh the potential benefits of using smart cards*" [36]. Especially the need to equip workstations with card readers was seen as a major hindrance.

This is however not the only way authentication on the blockchain can be realised. Buldas et al. from Guardtime proposed a hash sequence authentication method for use in their blockchain system [37]. Their aim was to make their infrastructure more quantum computing resistant under the assumption that hash functions would stay secure in a quantum computing environment. The concept of hash sequence authentication has first been proposed by Lamport in 1981 [38].

Another authentication method that has seen interest is the use of biometric systems. However most biometric cryptosystems

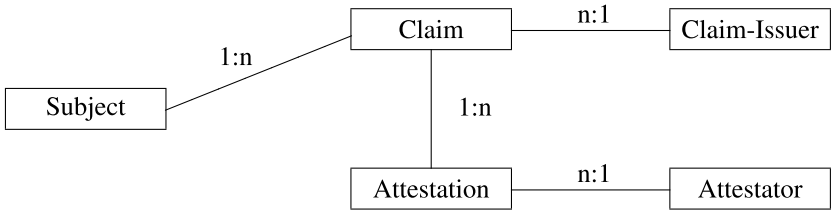

Fig. 5. Relation between components in a verifiable claim.

need biometric dependent information (helper data) which could potentially reveal significant information about the original biometric template [39].

The W3C DID working group proposes the use of external biometric services in combination with a cryptographic hash of the biometric templates.

In theory any authentication method could be used through an identification service endpoint as defined in the W3C Verifiable Claims Working Group specification draft [40], however a self-authenticating method such as public key cryptography or hash sequences do not need to rely on any third party endpoints, eliminating yet another point of centralisation.

As the authentication in such a case relies on a secret held by the end-user it would be beneficial to provide him with a way for key recovery/replacement. In the DID scheme this is done by separating authentication from authorisation allowing others to also change the DID document, i.e. changing the authentication key after the private key was lost.

uPort uses a quorum based key recovery where the holder logic includes a way for previously selected delegates to vote on replacing the public/private key pair of the user.

Key recovery seems to be a necessity for a working SSI system, since key losses are inevitable as the experience from bitcoin and other cryptocurrencies shows. In Bitcoin's case up to a quarter of all current coins have been lost due to unrecoverable private keys [41].

5. Verifiable claims

At the center of the Self-Sovereign Identity concept lay the *verifiable claims*. The first clarification that is necessary in this context is between a *claim* and a verifiable claim. A claim in itself is only a statement about a specific subject. A *credential*, which some differentiate from claims [42], describes a number of claims together with their meta data such as issuer and validity period.

Verifiable claims are verifiable through a signature of an attestation issuer that has either issued the claim himself or can attest the correctness of it. An *attestation* can be seen as a proof in form of a signature attesting to a certain claim and meta data needed for verification such as name, validity period and signature scheme.

The verifiable claims themselves have to be associated with a subject, typically by including the subject identifier. This can be observed in Fig. 5 where the relation of different components and actors in a verifiable claim is shown. In addition to the subject, a verifiable claim should hold information about one or more actual claims as well as some meta data. The claim is issued by exactly one claim issuer. Similar to X.509 certificates meta data in a verifiable claim could include a validity period, the identity of the issuer and algorithms used for signature/encryption. To make the claim verifiable and trustable, the issuer has to sign the claim with a well known key. This is shown in Fig. 5 where each claim can have multiple attestations and each attestation has one attestator.

There are mainly two different ways for claims and attestations to be linked to a users identity. uPort which operates on the Ethereum network utilises smart contracts to keep a registry for claims on the blockchain. In the registry they maintain a mapping of user to hashes of claims that are stored off-chain. Through this fingerprint the integrity of a claim can be verified by relying parties. More specifically the timestamping property of the blockchain is utilised to prevent secret modification of a claim and its signature. This architecture however only allows the user to add new claims to his identifier unless a more complex access management is implemented for the uPort registry. A registry model inspired by the uPort registry is being standardised in the Ethereum Improvement Proposal (EIP) ERC780 [43]. In the proposed Ethereum Claims Registry the writing of claims is not limited to the owner of the identity but issuers can directly add new claims and also revoke them in the registry.

The W3C VCWG data model on the other hand does not utilise a claim registry. They only rely on the blockchain for the mapping between an identifier and an authentication method. By including the identifier in the claim and having the issuer sign it, already secures against tampering from outside sources, however not against tampering by the issuer of whoever holds the signing key of the issuer. When colluding with the holder of the verifiable claim changes to the claim would go undetected and backdating or similar attacks could be done.

However, W3C's approach is very privacy preserving in the fact that not even the existence of new claims can be derived from blockchain changes but it also does not leverage the blockchains ability to trustfully timestamp items. Claims that have been altered after first issuance would need to be updated in the uPort registry, which would be recorded on the blockchain, protecting against tampering by the claims issuer or anyone in control of the signing key.

Another advantage that the registry model provides is the ease of revocation. As there is a "central" (but physically decentralised) location for all claims it would be possible to extend the registry with a revocation mechanism.

In comparison to current service centric/centralised identity solutions a relying party does not have to only trust a single issuance of a claim. Rather through the aggregation of multiple attestations for a claim, a more overarching and more decentralised trust model can be formed. This allows for relying parties to employ their own local confidence in certain attestators, depending on their individual relation. Working systems such as uPort which have their first real world applications [44,45], although designed to support such reputation aggregation, so far only utilise the SSI as a way for single attestation claim verification in practice.

On a slightly higher level there is also the PGP like aggregation of multiple claims, not only attestations, which can be used to form a reputation model for an identity.

6. Storage

6.1. Public

Although most data in SSI is stored off-chain some of the data is essential to have on-chain. Most importantly the already mentioned authentication such as a public key is typically included in a public fashion. In the end it is up to the user's discretion to decide what information he wants to publicly reveal and what he wants

to control more closely. Both Blockstack and uPort have public profiles which not only include signing keys but also names and profile pictures. Especially Blockstack provides use-cases for public disclosure of information. Specifically social media accounts or PGP keys that need to be publicly available to realise their full potential are data that can be securely stored on the blockchain.

6.2. Private

In a lot of cases a user does not want to disclose claims about himself though. In most cases the privacy of the user has to be preserved. For this purpose most claims are stored off-chain not publicly available and either secured by the previously discussed claim registry model or simply linked by the identifier defined in the identifier registry.

Just as the public disclose of information, the user is also in control of where to store the claims. The most trustless way would be in a directly user controlled environment such as hardware in possession of the user. One such example would be the SelfKey project which utilises a users smartphone to store claims. This however poses some serious problems too. Namely data security both against data loss and data theft. The lack of data redundancy when locally storing claims on mobile devices as well as the security of the device itself have to be taken into consideration.

Blockstack on the other hand opted for centralised storage providers such as Amazon S3, Dropbox and Google Drive [27]. This helps prevent potential data loss as these systems are highly redundant. To minimise the impact of attacks on these systems, they are used in conjunction with each other, spreading the claim data over multiple providers.

Through the use of decentralised storage systems such as IPFS [46] uPort wants to minimise the reliance on centralised entities even more. IPFS is a peer-to-peer distributed file system based on distributed hash table technology and is only one example of decentralised storage a user can utilise.

7. Future work

As we have hinted at in Section 5, we consider the possibility for a reputation system for each individual claim an interesting future topic. Through the aggregation of multiple attestations, as well as weighting of different attestations a more complex than binary claim reputation might be realised. In the same vain as a reputation model for a verifiable claim, the reputation for the identity as a whole could be derived from all verifiable claims associated with a given identity.

8. Conclusion

In the age of increasing digital interactions and analysis of user data, the concept of Self-Sovereign Identities has gained a large amount of interest. It promises its users more control and a more user-centric experience that, in contrast to previous user-centric efforts, does not have to rely on any centralised entities. The concept of verifiable claims has been extended by the Identity Registry Model as well as the Claim Registry Model. These decentralised registries were enabled by blockchain technology and although not a necessity the storage can be decentralised too. This only leaves the claim-issuers and their position of trust as centralised entities in the system.

In this paper the architecture of Self-Sovereign Identity systems has been presented as well as terms further clarified. Most importantly an analysis of essential components of such a system was provided.

Conflict of interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

Acknowledgments

This work was partially supported by the German Federal Printing Office (Bundesdruckerei GmbH).

References

[1] Meinel, Gayvoronskaya, Schnjakin, Blockchain: Hype oder innovation, 2018, ISBN: 978-3-86956-394-7. [Online]. Available: https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/10314/file/tbhpi113.pdf.
[2] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.
[3] S. Haber, W.S. Stornetta, How to time-stamp a digital document, in: Conference on the Theory and Application of Cryptography, Springer, 1990, pp. 437–455.
[4] C. Dwork, M. Naor, Pricing via processing or combatting junk mail, in: Annual International Cryptology Conference, Springer, 1992, pp. 139–147.
[5] A. Back, et al., Hashcash-a denial of service counter-measure, 2002. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf.
[6] M. Vukolic, Eventually returning to strong consistency, IEEE Data Eng. Bull. 39 (1) (2016) 39–44.
[7] W3C VCWG. Verifiable Claims Working Group Frequently Asked Questions. Accessed: 08/10/18. [Online]. Available: https://w3c.github.io/webpayments-ig/VCTF/charter/faq.html.
[8] C. Allen, The path to self-sovereign identity, 2016. Accessed: 31/1/2018. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.
[9] A. Tobin, D. Reed, The inevitable rise of self-sovereign identity, Sovrin Found. (2016) accessed: 08/10/18. [Online]. Available: https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf.
[10] Z. Wilcox-O'Hearn, Names: Decentralized, secure, human-meaningful: Choose two, 2003. [Online]. Available: https://web.archive.org/web/20011020191610/http://zooko.com/distnames.html.
[11] P.H. Levine, The apollo domain distributed file system, in: Distributed Operating Systems, Springer, 1987, pp. 241–260.
[12] Leach, Mealling, Salz, A universally unique identifier (uuid) urn namespace, RFC4122. [Online]. Available: https://tools.ietf.org/html/rfc4122.
[13] IEEE Standards Association. Registration authority. http://standards.ieee.org/develop/regauth/. Accessed: 08/10/18.
[14] P. Jesus, C. Baquero, P.S. Almeida, Id generation in mobile environments, 2006. [Online]. Available: https://repositorium.sdum.uminho.pt/bitstream/1822/36065/1/1159.pdf.
[15] Cooper, Dzambasow, Hesse, Joseph, Nicholas, Internet x.509 public key infrastructure: Certification path building, RFC4158, 2005. [Online]. Available: https://tools.ietf.org/html/rfc4158.
[16] J.R. Prins, B.U. Cybercrime, Diginotar certificate authority breachoperation black tulip, Fox-IT, November, 2011. [Online]. Available: https://www.rijksoverheid.nl/ministeries/ministerie-van-binnenlandse-zaken-en-koninkrijksrelaties/documenten/rapporten/2011/09/05/diginotar-public-report-version-1.
[17] C. Soghoian, S. Stamm, Certified lies: Detecting and defeating government interception attacks against ssl (short paper), in: International Conference on Financial Cryptography and Data Security, Springer, 2011, pp. 250–259.
[18] Callas, Donnerhacke, Finney, Shaw, Thayer, Openpgp message format, RFC4880. [Online]. Available: https://tools.ietf.org/html/rfc4880.
[19] I. Clarke, O. Sandberg, B. Wiley, T.W. Hong, Freenet: A distributed anonymous information storage and retrieval system, in: Designing privacy enhancing technologies, Springer, 2001, pp. 46–66.
[20] Namecoin. https://namecoin.org. Accessed: 08/10/18.
[21] Emercoin. https://emercoin.com. Accessed: 29/01/18.
[22] H.A. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, A. Narayanan, An empirical study of namecoin and lessons for decentralized namespace design, in: WEIS, 2015.
[23] C. Ellison, et al., Establishing identity without certification authorities, in: USENIX Security Symposium, 1996, pp. 67–76.
[24] Ethereum Name Service. https://ens.domains. Accessed: 08/10/18.
[25] Consensys. https://uport.me. Accessed: 08/10/18.
[26] C. Lundkvist, What is a uport identity. Accessed: 8/10/2018. [Online]. Available: https://www.reddit.com/r/ethereum/comments/5wi6cl/what_is_a_uport_identity/deaki14.

[27] M. Ali, J.C. Nelson, R. Shea, M.J. Freedman, Blockstack: A global naming and storage system secured by blockchains, in: USENIX Annual Technical Conference, 2016, pp. 181–194.
[28] Drummond, Reed, Manu, Sporny, Dave, Longley, Christopher, Allen, Ryan, Grant, Markus, Sabadello, Decentralized identifiers (dids) v0.7: Data model and syntaxes for decentralized identifiers, W3C, Tech. Rep., 01, 2018.
[29] Civic technologies. https://civic.com. Accessed: 08/10/18.
[30] Selfkey network. https://selfkey.org. Accessed: 08/10/18.
[31] Sovrin foundation. https://sovrin.org. Accessed: 08/10/18.
[32] J. Holt, https://github.com/jonnycrunch/ipid. Accessed: 08/10/18.
[33] Digital bazaar. https://veres.one. Accessed: 08/10/18.
[34] C. Fromknecht, D. Velicanu, S. Yakoubov, A decentralized public key infrastructure with identity retention. 2014.
[35] Allen, Brock, Buterin, Callas, Dorje, Lundkvist, Kravchenko, Nelson, Reed, Sabadello, Slepak, Thorp, Wood, Decentralized public key in- frastructure: A white paper from rebooting the web of trust, Rebooting the Web of Trust, 2015.
[36] D. Chadwick, Smart cards aren't always the smart choice, Computer 32 (12) (1999) 142–143.
[37] A. Buldas, R. Laanoja, A. Truu, Efficient quantum-immune keyless signatures with identity. IACR Cryptology ePrint Archive, Vol. 2014, p. 321, 2014.
[38] L. Lamport, Password authentication with insecure communication, Commun. ACM 24 (11) (1981) 770–772.
[39] C. Rathgeb, A. Uhl, A survey on biometric cryptosystems and cancelable biometrics, EURASIP J. Inf. Secur. 2011 (1) (2011) 3.
[40] D. Burnett, M. Sporny, D. Longley, G. Kellogg, Verifiable Claims Data Model. https://www.w3.org/TR/2017/WD-verifiable-claims-data-model-20170803/. Accessed: 08/10/18.
[41] J.J. Roberts, N. Rapp, Nearly 4 million bitcoins lost forever, new study says, 2017, accessed: 8/10/2018. [Online]. Available: http://fortune.com/2017/11/25/lost-bitcoins/.
[42] M. Sporny, https://github.com/w3c/vc-data-model/issues/112. Accessed: 08/10/18.
[43] J. Torstensson, Ethereum claims registry, ERC 780, accessed: 08/10/18. [Online]. Available: https://github.com/ethereum/EIPs/issues/780.
[44] City of zug. Accessed: 8/10/2018. [Online]. Available: http://www.stadtzug.ch/de/bevoelkerung/dienste/digitaleid/?action=showthema&themenbereich_id=1587&thema_id=5295.
[45] N. Benes, Announcing gnosis olympia. https://blog.gnosis.pm/announcing-gnosis-olympia-5fb7e16dd259. Accessed: 08/10/18.
[46] J. Benet, Ipfs-content addressed, versioned, p2p file system, arXiv preprint arXiv:1407.3561, 2014.