Technische Hochschule
Brandenburg
University of
Applied Sciences

Self-sovereign Identity: Development of an Implementation-based Evaluation Framework for Verifiable Credential SDKs

Brandenburg University of Applied Sciences
Department of Economics

Master's Thesis

submitted by

Philipp Bolte

September 1, 2021

First supervisor:	Prof. Dr. rer. nat. Vera G. Meister
Second supervisor:	Jonas Jetschni, M.Sc.

Statutory Declaration

I hereby attest that I have written this thesis independently without any outside help and that I have used only the sources cited.

Brandenburg, September 1, 2021

Philipp Bolte

Abstract

Diese LaTeX-Vorlage ist für Berichte, Bachelor- sowie Masterarbeiten gedacht. Natürlich ist sie nicht perfekt und jede Art der Verbesserung wird dankend angenommen. Bei Fragen zur Verwendung oder Anregungen zur Verbesserung können Sie mir diese gern an markus.brandt1992@gmail.com senden.

An entsprechender Stelle werden Beispiele für die Verwendung von Abkürzungen, Zitaten, Abbildungen, Tabellen und die Einbettung von Code gegeben.

Contents

List of Figures

List of Tables

List of Abbreviations

CA	Certificate Authority
DID	Decentralized Identifier
IDP	Identity Provider
IIW	Internet Identity Workshop
SDK	Software Development Kit
SSO	Single sign-on
SSI	Self-sovereign Identity
VC	Verifiable Credential
VP	Verifiable Presentation

1 Introduction

The Internet has become a cornerstone of coexistence in today's world. With over 4.66 billion Internet users worldwide [Jo21], it determines how we communicate, think, inform ourselves, and interact with one another. As a result, huge networks of people are being created in which different cultures are coming closer together and knowledge is being shared like never before. A central enabler for the functioning of such a digitalized society are digital identities [Li20].

Over the course of our lives, we generate a large amount of digital identities from a wide variety of services, including Facebook, Twitter, WhatsApp, GitHub, LinkedIn, and many more. They represent us in this digital realm, are part of our personality, and allow us to identify ourselves online. Because of the way we manage digital identities in the current era, users mostly own separate identities for each service or go through centralized, federated identity providers like Google or Facebook. As a result of these key developments, silos of identity data emerged, which are problematic concerning efficiency, security, and privacy. This creates a dependency towards the services that have full control over the identity data. This makes it difficult for users to control how services exploit this power for their own interests. In addition, various data leaks and hacks in which sensitive user data became public show that the current approaches are not suitable for the problems of these modern times [Sw21]. [Eh21, pp. 2-3]

In contrast, the SSI paradigm takes a new approach by giving users full control of their digital identities through various novel approaches [FCA19, p. 103059]. This work examines this new approach from a developer's point of view to test its practical applicability. In the next sections, the scope, related work and the research approach will be discussed.

1.1 Scope of Work

For a successful realization of Self-sovereign Identity (SSI) concepts, the existence of good solutions for developers is critical. This ensures that the barriers to a successful adoption of SSI are kept to a minimum, simplifying and speeding up the entire process. A good toolset and developer experience is thus a key enabler for SSI.

With this in mind, an overview of the most important solutions[1] in the SSI space is

[1] synonymous to Software Development Kits (SDKs), libraries, frameworks, and platforms

established throughout the thesis. To scope the work accordingly, this work looks at the solutions in terms of how closely they can map the lifecycle of a Verifiable Credential (VC). This decision was made due to VCs being a key artefact in SSI as they hold the actual verifiable data, e.g. vaccination status or birthdate, of a subject [SLC19]. The overview is intended to serve as an entry point for developers to get a general view of the capabilities of existing solutions and to give starting points for further research.

Furthermore, a use case agnostic reference implementation is presented that implements four of the presented solutions based on the lifecycle. It can serve developers as a basis for their own work, but above all enables practical validation and the gathering of experience during its development. This way, the knowledge gained flows directly into a new evaluation framework, which, in addition to other software selection frameworks, can provide concrete help in selecting the most suitable solution from the developer's point of view. In addition, it can reveal shortcomings in current solutions that need to be addressed for successful adoption of SSI in practical use cases. So the objective of this work, besides the scientific contributions, is to generate added value for the whole ecosystem.

1.2 Related Work

At the current time, there does not appear to be any comparable work that addresses the topic in a manner corresponding to section 1.1. The most similar is [NJ20] who have developed a mobile wallet based on uPort that covers login, VC issuance as well as verification. Based on the experience gained, an evaluation of uPort has been made as well. However, uPort is currently no longer being developed, and the assessment is also based on only a fraction of the VC lifecycle and basic principles for SSI.

Another paper by [Ku20] defines a comprehensive evaluation framework from an enterprise perspective that, compared to other papers, also covers aspects such as user experience, technology and compliance. It is characterized by a wide range of questions that are used for the evaluation of 43 solutions. However, the list of solutions considered is outdated and missing important players (see e.g. MATTR and Trinsic). Furthermore, the assessment does not provide any practical guidance for developers. A clear analysis of the SSI-relevant features, e.g. with regard to the VC lifecycle, does not exist.

Otherwise, many papers seem to focus on theoretical foundations or evaluation of existing solution based on two things: (i) architecture [GMM18] concerning privacy [Be19], performance [Bo20], use case [Ku20], various variations [Al16, Sp21, AL20, Bo20, FCA19, Ca05] of SSI principles [Bo19, Bo20, DT20, DP18, FCA19, SNE20], and (ii) the interoperability between those systems [Ho20, Jo20]. This clearly shows that there is a deficit in terms of works that look at existing solutions based on their practical features and applicability from a developer's point of view. This thesis addresses some of these gaps and thus clearly contributes to the field of research.

1.3 Methodology

The process for achieving the objectives from section 1.1 can be divided into four basic steps: gain theoretical foundation, create solutions overview, develop reference implementation, and define the evaluation framework (see figure 1.1).

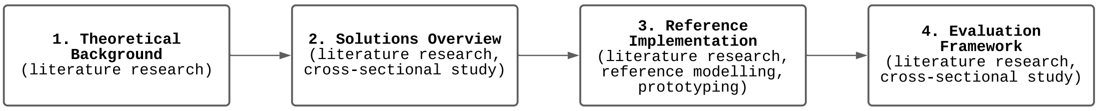

1. Theoretical Background
(literature research)
2. Solutions Overview
(literature research, cross-sectional study)
3. Reference Implementation
(literature research, reference modelling, prototyping)
4. Evaluation Framework
(literature research, cross-sectional study)

Figure 1.1: Research Approach

For this purpose, various methods of business and information systems engineering according to [WH07] are applied. Through a literature research, an overview of existing papers and books is created, which on the one hand serves for building a theoretical foundation, but also represents the basis for all other steps.

To find the literature, keywords such as "Self-sovereign Identity" and the following complex query based on [Bo19] were used at Google Scholar:

```
("Self-sovereign identity" OR "Self sovereign identity ")
OR (
    ("block-chain" OR "blockchain") AND ("identity management")
    AND ("solution" OR "implementation" OR "review" OR "survey")
    AND ("verifiable credentials" OR "decentralized identifiers")
)
```

Furthermore, the method of cross-sectional studies, mainly in the form of expert questionnaires, is used. A clearly defined set of questions is used to validate the solutions overview, but also to identify evaluation criteria for the evaluation framework. More details on the selection of experts and the questions are described in section X.X. For the creation of the reference implementation, the methods of reference modelling and prototyping are used. In combination, they allow the development of a software prototype that represents a particular problem in a simplified way and whose analysis can contribute to the discovery of new knowledge. This is especially interesting for the development of the evaluation framework. Moreover, it should also be noted that this thesis follows the general idea of generating real-world artefacts defined by [He07] as part of design science research.

To conclude the methodology, a combination of research approach and applied methods defined previously is reflected in the following research questions:

1. What libraries, platforms, or SDKs are available for implementing Verifiable Credentials?
2. Which SDKs do experts from the field recommend using?
3. Which criteria for evaluating Verifiable Credential SDKs can be derived after developing a reference implementation?

2 Self-sovereign Identity

Health care, social security, education, access to financial services — this is just a small list of requirements that are essential for a decent life and are usually taken for granted by people in the Western world. Yet, there are more than 1.1 billion people worldwide who cannot provide identification and thus cannot access the most basic services. Digital identities could make a significant contribution towards solving this problem and giving people the chance to participate in society on a more equal playing field. [Wo17]

As already mentioned in chapter 1, however, current implementations of such digital identities are insufficient for the problems of our modern times. [SNA21] divided these problems into four categories:

1. Data Ownership and Governance
2. Password-Based Authentication
3. Fragmented Identity Data
4. Data Breaches and Identity Fraud

The former describes the fact that users have no ownership over their digital identities and thus cannot exercise any control over them. Service providers often take advantage of this and use collected data to create comprehensive profiles of their users and thus sell tailored advertising space on corresponding marketplaces for high figures. The lack of control also means that service providers can temporarily or permanently deny users access to their digital identity at any time. At the beginning of 2021, this led to much discussion as the account of former U.S. President Donald Trump was permanently banned from Twitter. One of the central concerns was whether service providers have too much power over users' liberties [No21]. In addition, given the frequent and often repeated use of weak passwords, the heavy reliance on password-based authentication is a security risk that may lead to identity theft. If users want to protect themselves, they need to use different and complex passwords for each of their accounts, which quickly becomes a complicated undertaking without a password manager. A study by the password manager LastPass, for example, found that a business customer manages an average of 191 passwords [St17]. While the use of such tools greatly simplifies the management of passwords, they too can pose a major security risk and do not completely protect the user [OR20, Or21, To21]. Alternatives such as Single sign-on (SSO), where users authenticate to other service providers using for example their Google account, can solve this problem but lead to even greater dependency and centralization. The third issue involves identity data being spread across a large set of service providers, making it difficult to maintain. As a result, duplicates, errors, and outdated data sets are common. The lack of open standards also complicates interoperability between providers, which could theoretically be used to retrieve, move, or delete personal data. Efforts like the Data Transfer Project founded by Microsoft, Google, Twitter, and Facebook try to simplify the transfer of data between providers, but after more than 3 years show very few actual successes [Mi20, Ho21, Lo20] and are being criticized for pushing small competitors even further behind [BC18, p. 15]. [SNA21, pp. 2-3]

One of the biggest problems, however, are data breaches. In June 2021 alone, there were 235 breaches with 1.16 billion stolen records, with a total of 18.9 billion records stolen in 1,785 breaches in the first half of 2021 [Ri21]. Looking at the past, there have been quite a few major hacks [Sw21], including:

- Yahoo (2013): 3 billion accounts
- Marriott (2018): 500 million customer records
- Alibaba (2019): 1.1 billion entries
- LinkedIn (2021): 700 million accounts

A survey of 413 people by [Ma21] found that 73% of participants had been affected by at least one, but an average of 5.3 data breaches. In addition, the majority blamed themselves for the breaches, with only 14% aware that service providers were responsible.

These are decades-old problems that were already critically discussed by Kim Cameron in 2005. Cameron, who last worked as Chief Architect of Identity at Microsoft from 1999 to 2019, wrote the following on a blog article [Ca05]:

> *"The Internet was built without a way to know who and what you are connecting to. This limits what we can do with it and exposes us to growing dangers. If we do nothing, we will face rapidly proliferating episodes of theft and deception that will cumulatively erode public trust in the Internet."*

Cameron attributes these problems to the lack of an identity layer on the Internet, which has resulted in many services having to find their own solutions. He calls this a *patchwork of identity one-offs*, which fundamentally still exists today and is difficult to resolve. The reason for this, he says, is a lack of consensus and an unwillingness to give up too much control over identity data. A solution for this is, according to him, an *identity metasystem* that abstracts away deeper complexities similar to hardware drivers or TCP/IP and only loosely couples digital identities to the systems. Such an open identity layer could only be successful if it fulfilled the seven laws of identity defined by Cameron. These include criteria such as user control, consent, pluralism and minimal disclosure. [Ca05]

Over the years, these ideas, among others like [Ma12, id14, Al16], gave rise to the concept of Self-sovereign Identity (SSI). It is intended to eliminate the shortcomings of today's established concepts by placing the users in the center and giving them back complete control over their identity data. A user can decide what, to whom and how much data is shared without being dependent on a central authority. The emergence of blockchain technology and various new standards in recent years gave a new boost to implement SSI in reality. [St21, pp. 6-7; To17, pp. 8-9]

SSI is an entirely new approach to digital identities on the Internet and is seen as a paradigm shift that deeply affects the infrastructure and power distribution of the Internet [PR21, p. 3]. For a more profound look at the topic, this chapter takes a closer look at Self-sovereign Identity. To do so, the concept of identity and the different types of identities will be discussed first. This is followed by a historical look at the different stages of digital identities, taking a closer look at the previous concepts of SSI. After a basic foundation has been built, standards that have been established in recent years and are intended to make SSI feasible in reality are described. Finally, the SSI architecture with its components and roles will be looked at.

2.1 Identity

What defines a human being? One would probably get various answers to this question, such as its name, gender, place of residence, profession, hobbies, religion, charitable activities, party affiliation or even a combination of all these characteristics. [CK01, p. 206] describes in his work that a person's identity is not just a single, fixed construct, but consists of several partial identities. Thus, depending on the context in which a person finds itself, it takes on one of its various partial identities, which represents it as a human being more or less. For example, a partial identity for health care consists of its medical history, while the partial identity towards work contains received certificates. Nevertheless, these different parts of the identity are not necessarily considered separately, as they can also overlap in certain aspects of information. It is important to mention that a person decides which information to share at which time towards which entity. In figure 2.1 the concept of partial identities is illustrated exemplarily by a person Alice.

An important balancing act is to disclose the right amount of data to maintain anonymity, but also to provide the other person with the necessary information. For the purchase of a water, the kiosk vendor should not ask for any personal data, whereas verification of age when buying alcohol is a valid reason for information disclosure. In reality, official documents, such as state identification documents, or sometimes unofficial documents, such as customer cards, are usually used for such proof of identity. Here, users have full control over their documents as they are under their control, and only they can decide self-sovereignly whom and when to show them. Official identification documents are also produced and standardized to ensure the highest possible level of security and interoperability. Other countries can verify such documents without explicitly contacting authorities, simply by looking at the document. Confidence in the validity of the data arises from the fact that the verifying party trusts the authority issuing the document. [St21, p. 6]

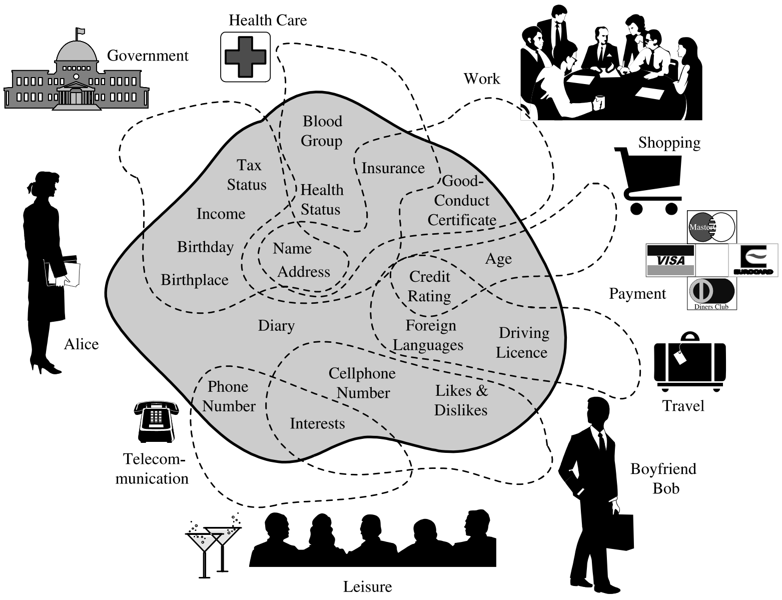


Figure 2.1: Partial Identities of Alice extracted from [CK01]

As a result of the increasing digitalization of various branches of life, many processes are shifting to the digital world. Digital identities, which are similar to analogue identities in terms of their basic idea, are now being used for interacting with digital services. They allow entities, such as people or objects, to authenticate themselves online through certain attributes and thus prove their identity [MG20, p. 103; Bu20]. A more precise definition is given by Cameron [Ca05], who defines digital identity as *"A set of assertions that a digital subject makes about itself or another digital subject"*. In this context, a digital subject is *"A person or thing represented or existing in the digital realm which is being described or dealt with."* and the attributes mentioned can be represented in the form of claims, which are defined as *"An assertion of the truth of something, typically one which is disputed or doubted"*. The problem is that analogue identities and their documents usually have no or not widely accepted [Kr19, Ko21] digital representations that could be used as a digital identity. From this emerged the patchwork of identity one-offs described in chapter 2, resulting in a divergence of digital identities from their original counterparts concerning their characteristics. To better understand this development, the next section describes the different stages of digital identities in more detail. [St21, p. 10; Eh21, p. 2]

2.2 Stages

As indicated in the last section, Allen's work [Al16] has had a major influence on what is today considered Self-sovereign Identity and has been cited in over 100 works according to Google Scholar. According to him, digital identities, or online identities, have gone through four major stages since the beginning of the internet. These are examined in more detail below and show which developments led to the emergence of SSI.

2.2.1 Centralized Identity

Centralized identities are identities that are issued and verified by a single party or hierarchy. The oldest examples of this are IANA (1988) for the administration of IP addresses, ICANN (1998) for domain names and Certificate Authorities (CAs), which play a major role today, particularly in connection with SSL certificates. Especially with the latter, the hierarchical structure of CAs becomes obvious when one looks at an SSL certificate in the browser. Here, a root authority allows another organization to manage its own hierarchy, while at all times the root authority has full control. This is highly critical for numerous reasons. For example, one entity has complete control over identities and can delete them at any time or even issue false identities. The latter can happen both willingly and unwillingly as a result of a hack. Due to the centralized nature of such authorities, they and thus also the complete hierarchy (chain of trust) are targets of attack, which has been shown in recent years [Bo12]. Just like these organizations, due to the lack of an identity layer, all services on the internet developed similar centralized solutions (see chapter 2. This manifests itself above all in the various accounts that an internet user has to manage for various services. Again, users have little control over their data. [Al16]

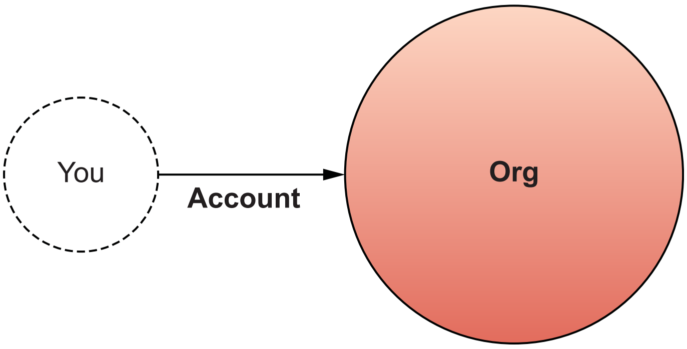


Figure 2.2: Relationship in centralized identities extracted from [PR21, p. 7]

In addition, the user has to manage the abundance of login credentials efficiently and securely. However, it is also a challenge for the services, as they have to store a large amount of sensitive data securely and in compliance with data protection laws. Nevertheless, the beneficiaries here are the services, as they can act flexibly and independently of third parties and have full control over the data. [Eh21, p. 6]

2.2.2 Federated Identity

The second stage of development is represented by the so-called federated identities, which were intended to break down the hierarchies based on a single authority. Here, various commercial organizations developed a model in which control was to be divided between several federated authorities. One of the first projects in this area was Microsoft's Passport in 1999, where Microsoft created a single, federated identity for users that could be used on multiple sites. However, this unification came with the price that Microsoft was now at the center of the federation and could thus exert full control. Other efforts, such as Liberty Alliance Project, founded in 2001, attempted to create an actual federation between multiple companies in which control was distributed among them. The result, however, was a kind of oligarchy in which users still had no control over their data. In the end, the sites remained authorities. [Al16]

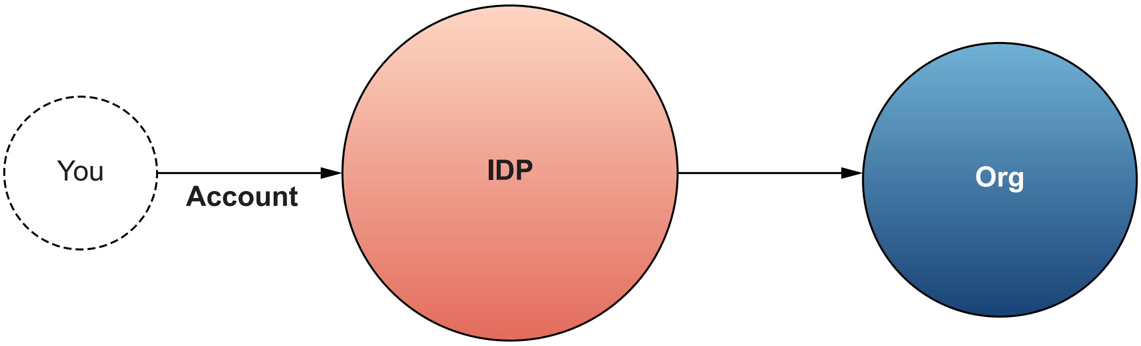


Figure 2.3: Relationships in federated identities extracted from [PR21, p. 8]

Nevertheless, this type of digital identity is advantageous in that users do not have to manage an identity/ account for each service and companies have less administrative effort. The identity provider, e.g., Microsoft, acts as the issuer and owner of the data and is thus the central point of contact if a user wants to log on to another service of the federation. The user therefore has no control over his data and is dependent on the continued existence of the identity provider. Due to the abundance of sensitive data, it is possible for the identity provider Identity Provider (IDP) to aggregate information from various areas in order to create user profiles, which in itself can lead to various problems. [Eh21, pp. 6 - 7]

2.2.3 User-Centric Identity

The goal of user-centric identity is to make federations obsolete and allow the individual to assert control over their identities across multiple authorities [Al16]. The foundations for this, according to [Al16], lie in [JHF03], in which a *persistent online identity* to be integrated directly into the architecture of the Internet was proposed, making federations unnecessary. One of their central demands was that users should have the right to control their own digital identity. This includes, among other things, the ability to decide what information is collected as part of their digital identity and who has access to which parts. Earlier approaches such as Microsoft's Passport or the Liberty Alliance Project were unable to meet these requirements because, as stated by [JHF03], they were too business-oriented and thus too focused on the privatization of information. According to them, everyone's digital identities should be a public good that should not be tied to the financial interests of a private company, as their commercial interest may not overlap with those of society.

These thoughts were guiding and influenced various future organizations and initiatives. One influential organization in this area has been the Internet Identity Workshop (IIW), which grew out of efforts by the Identity Commons and the Identity Gang. The IIW community played a major role in shaping what is understood by user-centric identity and supported key standards such as OpenID (2005), OpenID 2.0 (2006), OAuth (2010), and OpenID Connect (2014). [Al16] summarizes the focus of these efforts with the terms user consent and interoperability, which were non-existent or difficult to implement in previous models. These protocols have also been able to achieve significant success when considering the abundance of social logins from for example Facebook, Google, GitHub and Microsoft, which have taken a central position on various websites [PR21, p. 8]. Nevertheless, the original approach of user-centric identities could not be realized further. Like in previous approaches, the identity data and thus absolute control remain with the SSO providers who register them. [Al16] mentions OpenID as an example, which theoretically allows users to set up their own OpenID providers. However, the complexity is so great that in reality this option is hardly ever used. Accordingly, the original problems that user-centric identities were supposed to solve could only be partially solved, since central, mostly private actors have maintained their authority over identity data. Fundamentally, user-centric identities are still federated identities that are now merely interoperable, which is why some literature [Eh21, PR21] does not list them separately. [Al16]

2.2.4 Self-sovereign Identity

[Al16] refers to Self-sovereign Identity as the next and most current stage of digital identities, which is intended to solve the issues of all previous stages. In contrast to user-centric identities, users are not only at the center of the identity process, but should also be able to completely own and manage their identities. [PR21, p. 12] describes this as a "[...] shift in control from the centers of the network [...] to the edges of the network [...]", according to which all users interact directly with each other in a self-sovereign manner as peers. This evolution can be seen in figure 2.4.

Apparent here is the new element *registry*, which is used as a (decentralized) public key infrastructure [PR21, p. 89]. A more detailed explanation of this is given in section 2.3. To further describe the character of SSI, [Al16] defined 10 principles, with which he connects to previous works like the "Laws of Identity" by [Ca05]. These are:

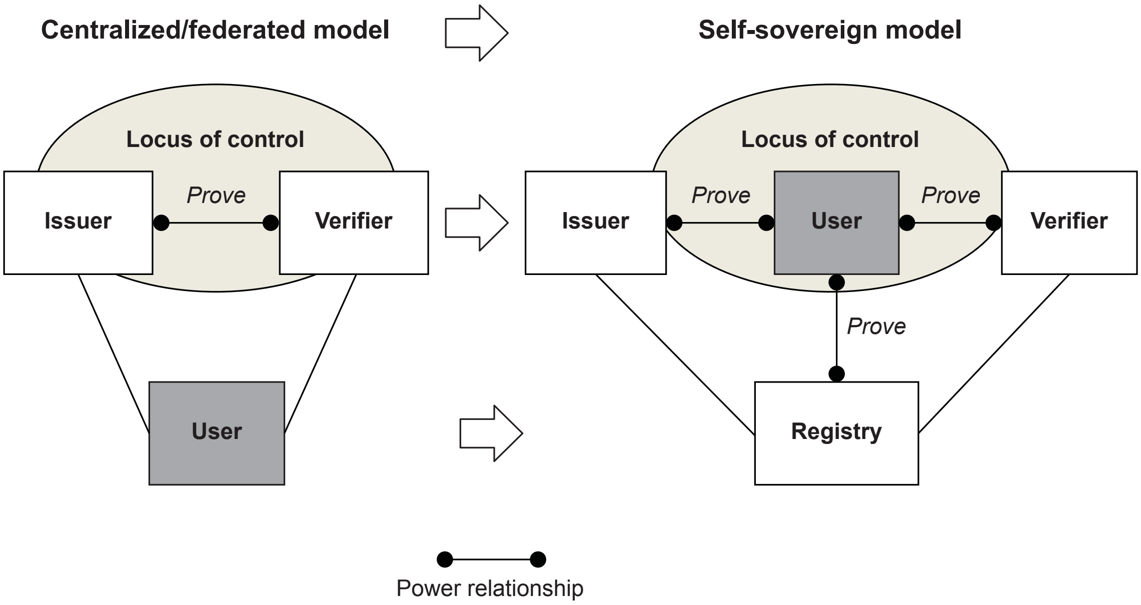


Figure 2.4: Shift of control with SSI extracted from [PR21, p. 12]

1. Existence: “Users must have an independent existence.”
2. Control: “Users must control their identities.”
3. Access: “Users must have access to their own data.”
4. Transparency: “Systems and algorithms must be transparent.”
5. Persistence: “Identities must be long-lived.”
6. Portability: “Information and services about identity must be transportable.”
7. Interoperability: “Identities should be as widely usable as possible.”
8. Consent: “Users must agree to the use of their identity.”
9. Minimalization: “Disclosure of claims must be minimized.”
10. Protection: “The rights of users must be protected.”

SSI, according to [Al16], has its origins in the term “Sovereign Source Authority”, which originated in [Ma12]. In this work, Marlinspike attributes to every human being the right to an identity, which is hindered by tight state structures. In the same year, work began on the Open Mustard Seed by Patric Deegan, which was intended to give users control over their digital identity in a decentralized system. This later resulted in the Windhover Principles (2014), under which the term Self-sovereign Identity appeared [id14, Hu14]. These state, among other things, the following: [Al16]

> *“Individuals [...] should have control over their digital identities and personal data ensuring trust in our communications, and the integrity of the data we share and transact with. [...] Individuals, not social networks,*

> *governments, or corporations, should control their identity credentials and personal data."*

Over the course of the following years, SSI in connection with blockchain technology was frequently also being discussed in the IIW community as well, and various ideas were being developed. This eventually led to some official agencies taking a closer look at this topic. For example, the U.S. Department of Homeland Security Science & Technology division published a report in 2015 in which it addressed the previously discussed topics by the IIW. The EU and countries such as China and Korea have also recognized the potential. In order to make SSI implementable in reality, various new standards have been defined over the years in the W3C, among others, which will be discussed in more detail in the next section. [PR21, p. 6]

2.3 Standards

In this section, the two most important standards *Decentralized Identifier (DID)* and *Verifiable Credential (VC)* will be discussed in more detail, as they are the basis for SSI and various subsequent standards.

2.3.1 Decentralized Identifier

Throughout history, humans have built up various imaginary networks in which they have to identify and address themselves or objects. Be it physical, mutual networks, in which one identifies itself with one's name or be it postal or telephone networks in which it is the addresses or the telephone numbers. In the age of the Internet, various others have been introduced, such as IP addresses, e-mail addresses, domain names or usernames in social networks. Consequently, there are a large number of identifier systems, which can vary greatly in their nature and place of application. Zooko Wilcox-O'Hearn published an article on this subject in 2001 [WO01], in which he describes a trilemma in identifier systems. According to this, an identifier can probably have at most two of the following properties: [PR21, pp. 183-186]

1. Human-readable: Identifiers have semantic meaning in human language and thus have low entropy.
2. Secure: Identifiers are unique and thus bound to a single entity. Spoofing and impersonation should not be possible as well.
3. Distributed: The namespace of the identifier is not managed by any central authority. Identifiers can be generated and resolved independently.

According to this, e-mail addresses, domain names and user names, for example, are human-readable and secure, but do not fulfil the distributed criterion. However, this is precisely what is needed for an SSI ecosystem in which users can own and manage their identity in a self-determined and sovereign manner. With the advent of blockchain technology, however, first solutions emerged that sought to break this problem and thus Zooko's Triangle. This involved the concept of decentralized domain services (see e.g. Namecoin or ENS), with which human-readable, secure and distributed identifiers can be generated. However, these are usually tied to the underlying blockchain and have intrinsic value due to their human-readable nature, which is why domains/ identifiers are often registered and held in such systems without actual use [Ka15], questioning the utility of the system.

To meet the requirements related to identifiers in an SSI system, the W3C standard *Decentralized Identifier* has been defined. These are *globally unique* and *location-independent* identifiers that can be generated autonomously by entities without central authorities and provide the ability to prove control over them through cryptographic evidence. Regarding Zooko's Triangle, DIDs don't attempt to be human-readable and thus satisfy the properties secure and distributed [PR21, p. 185]. [Sp21]

The characteristics of a DID can be summarized in the following points [PR21, p. 160]:

1. Persistent: DIDs have no intrinsic expiration date and do not need to change.
2. Resolvable: DIDs are resolvable to retrieve additional metadata.
3. Cryptographically Verifiable: The owner of a DID can cryptographically prove control over it at any time. This is enabled by a public and private key pair being assigned to a DID.
4. Decentralized: A DID *can* be issued/ generated independently of a central authority.

To better visualize how DIDs work, figure 2.5 provides the DID architecture with all its components and relations.

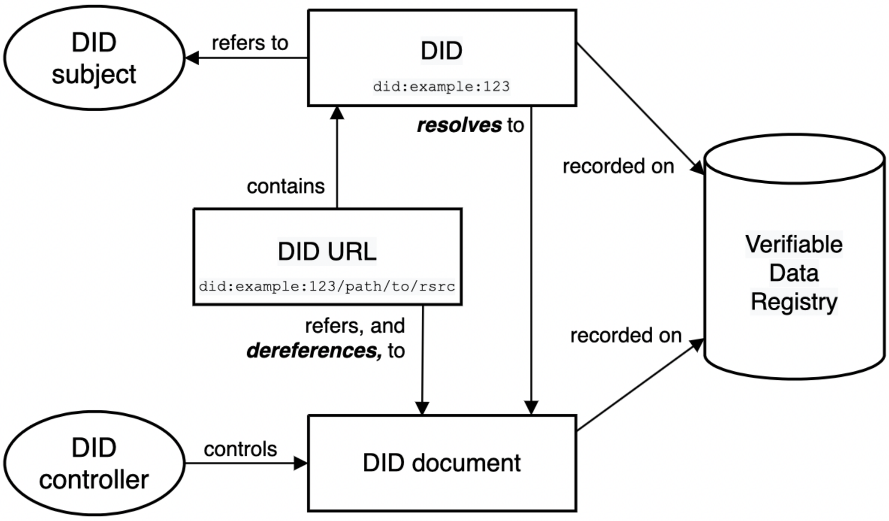


Figure 2.5: DID architecture extracted from [Sp21] (TODO: VECTOR!)

At the top is the DID subject, which can be any entity and is represented by the DID. The DID itself, for example `did:example:123456789abcdefghi`, is the actual identifier, which consists of three parts. The first part `did` describes the identifier schema, `example` the DID method and the third part `123456789abcdefghi` a DID method-specific identifier which can be used to resolve the DID document according to the DID method. Since DIDs are location-independent, they *can* be recorded in different Verifiable Data Registries, for example blockchains or decentralized file systems. The DID method defines any mechanisms for creating, resolving, updating and deactivating DIDs and their DID document that may be recorded on a specific data registry. Examples of DID methods that rely on blockchains or layers built on top of them as a Verifiable Data Registry include `did:ethr`, `did:btcr`, and `did:ion` which are in contrast to static DID methods like `did:key` which do not require a Verifiable Data Registry and usually wrap the public key from which the DID document can be derived [PR21, p. 171]. The DID document contains various metadata about the associated DID and define things like verification methods, public keys, and possible service endpoints for interactions with the subject. Additionally, paths can be attached to DIDs to address specific resources within the DID document. These are the so-called DID URLs. An example DID document can be found in listing 2.1. [Sp21]

```
{
    "@context": [
      "https://www.w3.org/ns/did/v1",
      "https://w3id.org/security/suites/ed25519–2020/v1"
    ]
    "id": "did:example:123456789abcdefghi",
    "authentication": [{
      "id": "did:example:123456789abcdefghi#keys–1",
      "type": "Ed25519VerificationKey2020",
      "controller": "did:example:123456789abcdefghi",
      "publicKeyMultibase": "
   zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
    }]
}
```

Listing 2.1: DID document example extracted from [Sp21]

Lastly, figure 2.5 includes the DID controller. This is another entity that is authorized to make changes to the DID document. Most of the time the DID controller is also the DID subject, but in some cases these entities can be different (see for example parent-child relationship). [Sp21]

In conclusion, the DID specification is a W3C standard that enables decentralized identifiers for SSI ecosystems. The next section presents Verifiable Credentials, which, in conjunction with DIDs, form the basic building blocks of SSI.

2.3.2 Verifiable Credentials

Now that a standard has been created with DIDs, with which entities can independently generate unique and authority-independent identifiers, there is still a need for a data model with which, in combination with DIDs, identity data can be represented in a standardized way. For this purpose, the W3C defined the Verifiable Credential (VC) standard. This defines VCs as a set of claims stated by an issuer in a tamper-evident way, allowing integrity and authorship to be cryptographically verified. A claim is thereby defined, similarly to subsection 2.1 by Cameron, as an *"[...] assertion made about a subject."* where a subject is a *"[...] thing about which claims are made."*. A VC is written in JSON-LD and consists of three basic components, which are visualized in figure 2.6. [SLC19]

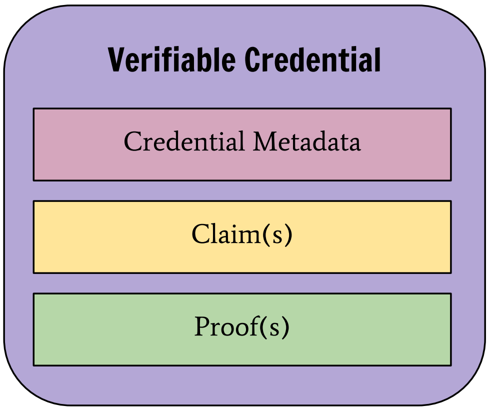


Figure 2.6: Components of VC data model extracted from [SLC19]

The credential metadata can define various properties of the VC including its issuer, an expiration date, credential types, or a revocation mechanism that can be used to check whether the issuer has revoked the credential. Thereafter, a set of claims can be defined by the issuer, which contains statements about the subject of the VC. Finally, cryptographic proofs can be attached and used to verify the validity of the credential's contents. The standard distinguishes between two types of proofs: [SLC19]

1. External Proof: The contents of the credential are wrapped, thus converted into a different, cryptographically verifiable format. A well-known example of this are JSON web tokens, which are used today in many identity systems for the transfer of claims between multiple parties, providing a certain compatibility to existing systems. Effectively, the set of claims is represented in a digital signature, the JSON web signature. Since these were developed for the JSON format, proofs can only refer to an entire credential and not to individual attribute sets [He20b]. [SLC19]

2. Embedded Proof: The proof is contained in the data and is therefore JOSN-LD native, which makes pre- or post-processing of the data unnecessary [SLC19]. Such so-called Linked Data Proofs use Linked Data Signatures, which can

create proof chains on the basis of the semantic structure of JSON-LD. This enables proofing on an attribute basis, rather than per credential as in external proofs. This high amount of flexibility also creates room for other technological possibilities, such as zero knowledge proofs. [He20b]

In listing 2.2 is an exemplary JSON-LD document, which is leveraging the Verifiable Credentials Data Model, attesting the credential subject a bachelors degree.

```
{
   "@context":[
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
   ],
   "type":[
       "VerifiableCredential",
       "UniversityDegreeCredential"
    ],
    "issuer":{
        "id":"did:key:z6MkhMLpju5tqtbd54BSv7Sq2oRWQo6n..."
    },
    "issuanceDate":"2021—05—26T08:33:40.681Z",
    "credentialSubject":{
       "id":"did:key:z6MkhMLpju5tqtbd54BSv7Sq2oRWQo6n...",
       "type":"BachelorDegree",
       "name":"Bachelor of Science and Arts"
    },
    "proof":{
        "type":"Ed25519Signature2018",
        "created":"2021—05—26T08:33:40Z",
        "jws":"eyJhbGciOiJFZERTQSIsImI2NCI6ZmFsc2UsImNyaXQ...",
        "proofPurpose":"assertionMethod",
        "verificationMethod":"did:key:
   z6MkhMLpju5tqtbd54BSv7Sq2oRWQo6njMEywrbWAGAp3442#z6MkhM..."
    }
}
```

Listing 2.2: Example of a Bachelors degree as a Verifiable Credential

Listing 2.2 also shows the basic building blocks described earlier. The document starts with a context definition to reference the semantic vocabulary. This is followed by a definition of the credential types, the issuer and the issuing time (credentials metadata). After that follows the actual claim in the object `credentialSubject`, where the subject and the corresponding degree are defined. At this point it also becomes clear how the DID and VC specifications are intertwined: both issuer and subject are defined by their DID. The public private key pair belonging the issuer's

DID becomes relevant especially in the next point: the proof. Here, the issuer uses the private key coupled to its DID to generate the Linked Data signature and thus makes the credential verifiable.

Another important part of the standard are Verifiable Presentations (VPs). If a holder of a Verifiable Credential wants to present it to someone, it can combine one or more VCs in a Verifiable Presentation without invalidating the cryptographic proofs. This approach has several advantages. On the one hand, the owner of the credential can specify granularly which credentials it wants to disclose and, at the same time, a type of proof of ownership can be provided. This becomes particularly clear if one considers the structure of such a VP in figure 2.7. [SLC19]

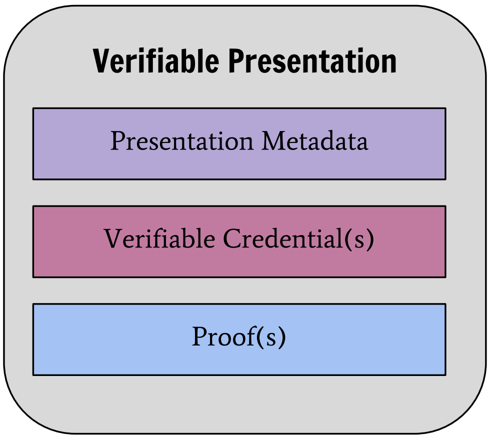


Figure 2.7: Components of a Verifiable Presentation extracted from [SLC19]

Once again, metadata is defined at the beginning, which can include attributes like the context and types. This is followed by the VCs to be presented, which are listed directly one after the other without any changes. Finally, a cryptographic proof follows, which the owner of the credentials generates with the private key of its DID. This protects the integrity of the presentation and at the same time certifies that the credentials are actually presented by the owner of the DID. The inclusion of the attributes `challenge` and `domain` in the proof can also provide protection against replay attacks, in which an attacker presents the intercepted presentation again to another verifier without authorization. [SLC19]

Having introduced Decentralized Identifiers and Verifiable Credentials as the backbone of an SSI ecosystem, the next section will specify a few things in more detail.

2.4 Architecture

In this section, various functional aspects of SSI are described in more detail. For this purpose, the roles and interactions in a SSI system are described, followed by a general look at the technology stack.

2.4.1 Roles

In an SSI ecosystem, there are three basic roles that participants can occupy: issuer, verifier, and holder. They have already been briefly described in subsection 2.3.2 and are therefore an integral part of the VC standard. The three roles are briefly presented below: [PR21, pp. 25-26; SLC19]

1. *Issuer*: An entity that makes statements within a VC about a subject. Such an entity can be organizations such as governments, universities, but also private individuals or objects such as sensors. An issuer transmits VCs to holders.
2. *Holder*: An entity that requests or receives VCs from issuers and manages them in a credential repository/ digital wallet. However, a holder may not always be the (credential) subject. Examples of these cases include a parent (holder) holding VCs for its child (subject) or a friend (holder) filling a prescription at the pharmacy for its sick friend (subject). Holders can also generate Presentations from Verifiable Credentials and show them to a verifier.
3. *Verifier*: An entity that wants to verify certain attributes or claims of a subject. It may receive these in the form of VP, which may contain those claims from one or more VCs. However, holders have control at all times over which attributes are passed to the verifier.

The roles and their relations are often called the trust triangle, as it describes how trust is formed in an SSI ecosystem. Like in the real world, trust in the credentials comes from a verifier trusting the issuer. The figure 2.8 shows this triangle, but also visualizes how the roles interact with VCs, which is why this process is also called the Verifiable Credential Lifecycle. [PR21, pp. 25-26; SLC19]

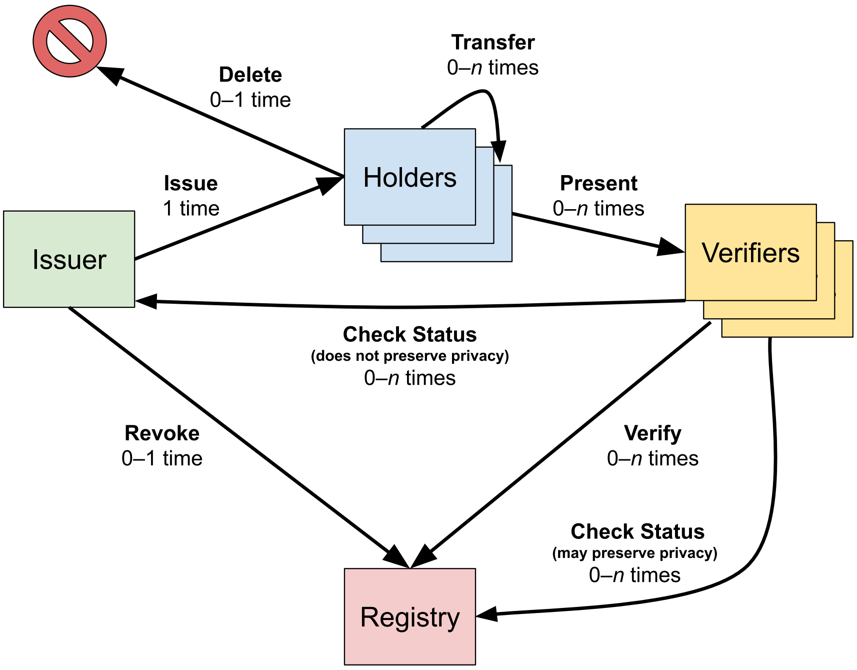


Figure 2.8: Verifiable Credential Lifecycle edited and extracted from [SLC19]

The lifecycle shows which phases a Verifiable Credential goes through and which roles perform which actions in these phases. At this point, the process is described using a Verifiable Credential representing a bachelor's degree as an example. Here, the issuer is a university, the holder is an alumnus, and the verifier is a potential employer. The process is as follows: [SLC19]

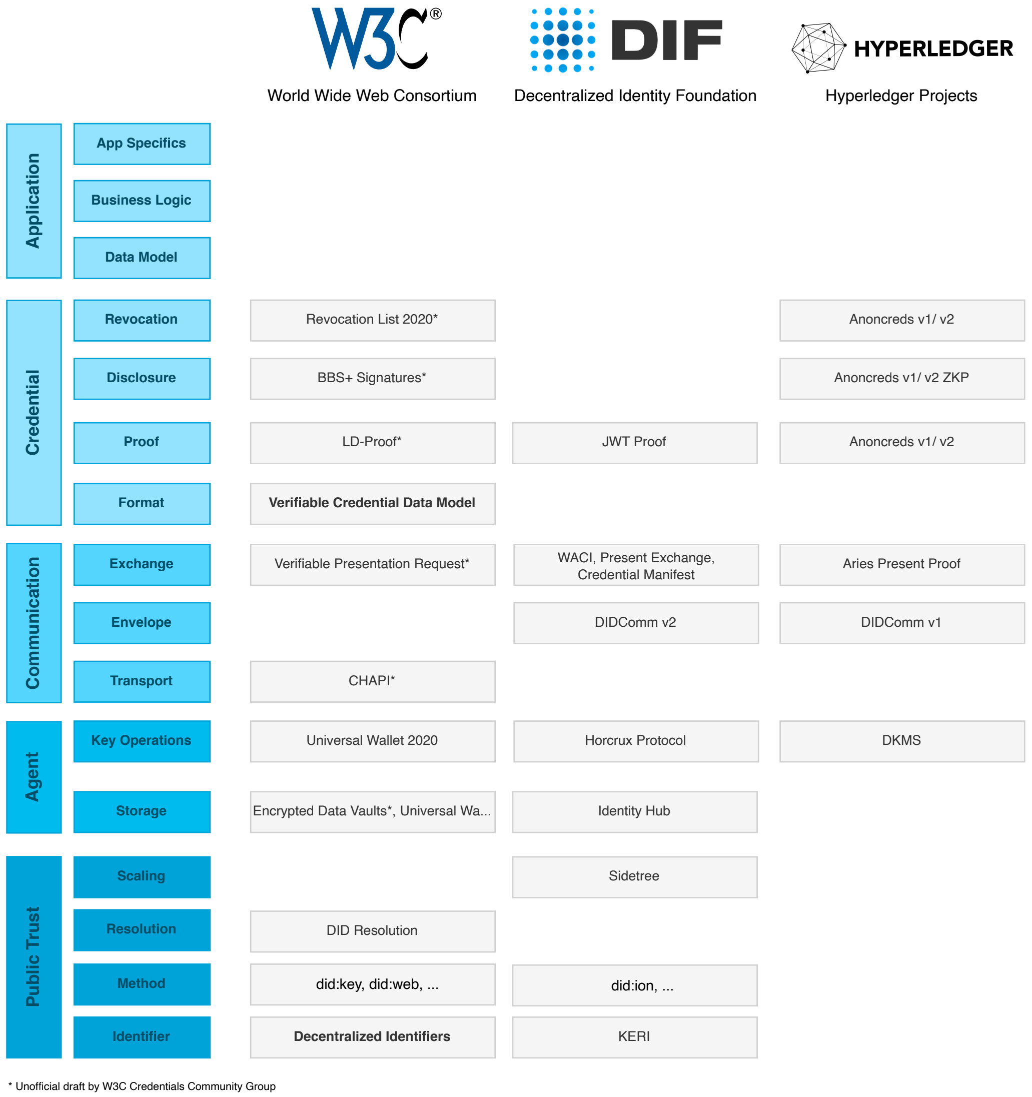
1. *Issue*: The now alumnus has successfully defended its thesis. Its university then issues a Verifiable Credential to the DID of the alumnus with its own DID. As holder and subject, the alumnus stores the VC in its digital wallet.
2. *Transfer*: The alumnus can transfer this VC to another holder, e.g., if he authorizes a friend to go to a governmental authority for him with the degree.
3. *Present*: The alumnus presents the VC of the bachelor's degree, optionally inside a VP, to the potential employer as part of his application in order to have his degree verified.
4. *Verify & Check Status*: The potential employer checks the authenticity of the credential. This includes firstly checking that the credential meets the standard, the proofs are valid and is not revoked. To check the proof, the employer must resolve the DID documents of the DIDs in the credential to obtain the public keys. For this, depending on the DID methods used, the employer may need to query a verifiable data registry (see subsection 2.3.1.
5. *Revoke*: If the university wants to revoke and thus invalidate a VC for some reason, it can do so. Depending on the implementation, this is also done with some kind of decentralized or central registry. One of them is described later on in subsection 2.5.3.
6. *Delete*: A holder can delete a VC from its digital wallet at any time, which does not affect its overall validity.

This process can be applied to any other use case and creates a system through defined standards, technologies and the described trust model, which in its basic characteristics also takes place in real interactions and where trust can form between entities.

2.4.2 Technology Stack

Looking at the SSI technology stack, figure 2.9 provides an overview that divides it into five basic layers. This is based on previous work from the Decentralized Identity Foundation and the Trust over IP organization [He20a**?** , Da21]. A notable change is that here the communication layer has been broken out of the agent layer. Even though it is mostly used by agents, it's not part of the agents itself, but rather an implemented software module leveraging the communication layer.

The public trust layer is the baseline layer and thus forms the basis for all other layers above it. The aim here is to create a public trust registry that includes, for



		World Wide Web Consortium	Decentralized Identity Foundation	Hyperledger Projects
Application	App Specifics			
	Business Logic			
	Data Model			
Credential	Revocation	Revocation List 2020*		Anoncreds v1/ v2
	Disclosure	BBS+ Signatures*		Anoncreds v1/ v2 ZKP
	Proof	LD-Proof*	JWT Proof	Anoncreds v1/ v2
	Format	**Verifiable Credential Data Model**		
Communication	Exchange	Verifiable Presentation Request*	WACI, Present Exchange, Credential Manifest	Aries Present Proof
	Envelope		DIDComm v2	DIDComm v1
	Transport	CHAPI*		
Agent	Key Operations	Universal Wallet 2020	Horcrux Protocol	DKMS
	Storage	Encrypted Data Vaults*, Universal Wa...	Identity Hub	
Public Trust	Scaling		Sidetree	
	Resolution	DID Resolution		
	Method	did:key, did:web, ...	did:ion, ...	
	Identifier	**Decentralized Identifiers**	KERI	

* Unofficial draft by W3C Credentials Community Group

Figure 2.9: SSI technology stack, standards, and efforts based on [He20a, Yi21, Da21]

example, DIDs and their DID methods and thus serves as a (decentralized) public key infrastructure. As already mentioned, this may or may not imply the need for technologies like blockchains or decentralized file systems. In the subsequent agent layer, which fundamentally allows an entity to *"[...] take actions, perform communications, store information, and track usage of the digital wallet."* [PR21, p. 192] and thus handles tasks related to storing VCs and DIDs and performing key operations. This includes for example the generation of proofs, but also the deactivation or generation of new key pairs. On top of this is the communication layer, which handles the communication between agents. This includes transport, envelope, and credential exchange standards and protocols. On the fourth level is the credential layer, which includes all standards and technology used in the credentials' data model, such as formats, types of proofs, disclosure, and revocation. The top level is the application layer, which creates user applications based on the underlying layers that cover and implement specific use cases. This includes concrete data models for the credentials, but also business and application-specific logic and technology. [He20a, Yi21, Da21, PR21]

In addition to the general structure and elements of the layers, figure 2.9 contains concrete standards and community efforts which provide the layers with a technological basis. A differentiation is made here between the concrete efforts of the three most important organizations in this area. It should be noted at this point that this overview does not claim to be complete and is merely illustrative.

2.5 Recent Developments

As mentioned in the last subsection and in figure 2.9, there are various community efforts that try to fill the gaps in the SSI stack. Therefore, three of the most important efforts are examined in more detail in the following subsections. The selection was made by observations of the reference implementation and the last Internet Identity Workshop in April 2021.

2.5.1 DIDComm

DIDComm, or DID communication, is a standard for secure, asynchronous, peer-to-peer communication between agents based on the DID standard. It is managed by the DID-Comm Working Group of the Decentralized Identity Foundation [Ha21] and originated from efforts of the Hyperledger project [Ha19]. Messages are agnostic of the transport medium, so existing protocols such as HTTP, Bluetooth, NFC, or even QR codes can be used. Moreover, the standard focuses on machine-readable messages, which enables a broader mass of use cases where all kinds of entities can exchange any kind of encrypted messages. [PR21, pp. 96-97]

If an entity A wants to send a message to entity B, it prepares a JSON message and retrieves the DID document of B's DID. Out of this, it needs two pieces of information: A messaging endpoint and the public key. With the latter, A can encrypt the message so that only B can decrypt it. In addition, A attaches a signature that it created with its own private key. This allows B to verify the integrity and origin of the message. Depending on the messaging endpoint and the transport route, the message is either delivered directly or scheduled for several hops via intermediaries. The standard provides various routing-specific information for this. If B now receives the message, it can decrypt it with its private key and verify the signature with As public key. If everything is correct, B can reply in the analogous way to As's procedure. The specification describes its goals in eight points: [Ha21]

1. Secure: Temper-proof using cryptography.
2. Private: Intermediaries don't know who is when communicating about what. Senders can be anonymous.

3. Decentralized: Trust is based on keys derived from control of DIDs
4. Transport-agnostic: Usable with any transport protocol. No matter if simplex, duplex, synchronous, asynchronous, online, or offline.
5. Routable: Messages can be routed like email through any kind of infrastructure.
6. Interoperable: Independent of hardware or software.
7. Extensible: Easily extensible by developers.
8. Efficient: Low resource requirements.

To better understand the structure of a DIDcomm message, listing 2.3 shows a plaintext version.

```
{
    "typ": "application/didcomm-plain+json",
    "id": "1234567890",
    "type": "<message-type-uri>",
    "from": "did:example:alice",
    "to": ["did:example:bob"],
    "created_time": 1516269022,
    "expires_time": 1516385931,
    "body": {
      "messagespecificattribute": "and its value"
    }
}
```

Listing 2.3: Plaintext DIDComm message extracted from [Ha21]

Subsequently, `type` describes the media type of the message, i.e. whether it is unencrypted, encrypted, and or signed. This is relevant for the corresponding library how to handle the content. DIDComm relies here on some JSON Web Algorithms from the JOSE (Javascript Object Signing and Encryption) family, which standardizes these cryptographic operations. `id` is the message ID and identifies the message exactly. Next, `type` describes what kind of message is in plaintext so that it can be handled correctly at the application level later. The next two attributes `from` and `to` define the DID of the sender and the DIDs of the recipients, followed by timestamps defining the creation and expiration time of the message. All attributes up to this point are the so-called message header, which is followed by the `body` containing the actual message. [Ha21]

The specification is much more detailed in many points, but this will not be considered further here with regard to the scope of this work. DIDComm as a secure communication method over DIDs is considered one of the most promising specifications in this area [PR21, p. 97] and can significantly contribute to how entities can exchange simple messages or even VCs peer-to-peer. Nevertheless, DIDComm is still relatively new, so its toolset and adoption is still relatively small.

2.5.2 BBS+

With regard to subsection 2.2.4, Allen describes in his ten principles for SSI, among other things, the principle of “minimization”, according to which the number of released claims should be kept as low as possible. This is also known as selective disclosures, according to which a user can keep certain attributes of a credential secret, which corresponds to the blackening of documents in the analog world. The next step to this approach are so-called zero knowledge proofs, where the actual attribute is not shared, but an assertion of a value that confirms what the verifier wants to know (predicates). For example, if an age check takes place, the verifier does not actually need to know the exact age but only the fact whether the person is over 18 or not. There are two basic approaches to this in the community: Camenisch-Lysyanskaya signatures and BBS+ signatures. Camenisch-Lysyanskaya signatures were one of the first implementations in the form of Anoncreds v1, but were dependent on published schemes for each credential on a ledger and therefore not standard-compliant. Furthermore, they were accompanied by large keys and large credentials that were costly to generate [Zu21]. [Yo21, pp. 17-18]

In the fall of 2020, MATTR announced BBS+ LD proofs that promised the same benefits while maintaining compatibility with the VC specification and reducing credentials and signature size. In addition, signatures were significantly faster to generate and the dependency on a ledger was not needed any more. [Zu21]

Table 2.1: Comparison of BBS+ and CL signatures (based on MA20b, He20c)

Domain	Criterion	BBS+ Signatures[1]	CL Signatures
Size	Private Key	32 Bytes	256 Bytes
	Public Key	96 Bytes	$771 + \frac{257}{msg}$ Bytes
	Signature	112 Bytes	672 Bytes
	Proof	$368 + \frac{32}{hidden\_msg}$ Bytes	$696 + \frac{74}{msg}$ Bytes
Performance	Key Generation	1 ms	8.8 sec
	Signing[2]	2.58 ms	93 ms
	Verifying[2]	5.23 ms	11 ms
	Proof Generation[2]	10.6 ms	13 ms

[1] BLS12-381 elliptic curve

[2] For 10 messages

Technically, BBS+ LD proofs rely on a combination of Linked Data proofs, the JSON-LD credential schema and BBS+ signatures. This combination allows generating proofs for presentations that can contain only a subset of attributes of the original VCs without affecting the semantic expressiveness and cryptographic integrity. Attribute filtering is performed using JSON-LD framing, while BBS+ is used to generate a so-called multi-message signature. As the name suggests, instead of one signature of one message, here a signature is composed of an array of messages. This ultimately allows the resulting signature to be derived by the holder so that it can only represent a part of the attributes. Unlike Camenisch-Lysyanskaya signatures, current implementations of BBS+ do not yet allow the use of predicates to enable comprehensive zero knowledge proofs. Cryptographically, however, such support is possible, but has not been the focus of efforts to date. Alternatively, the values of such predicates can be incorporated directly by the issuer as a separate attribute in the VC, which can then be disclosed by the holder through selective disclosures. The associated specification "BBS+ Signatures 2020" [LS21b] is currently an unofficial draft, which is managed by the W3C Credentials Community Group. [Yo21, pp. 18-20]

2.5.3 RevocationList2020

An elementary part of the VC lifecycle is the revocation step (see subsection 2.4.1). It allows issuers of VCs to revoke them at any time due to various reasons. Such reasons may be, for example, the expiration of a credential whose expiration date was not known at the time of issuance (e.g., office building access card), or incorrectly issued credentials (e.g., fraud).

The VC standard describes this process step and some important requirements, but does not define a standard for it. It only provides that there could be so-called "revocation registries" which could be part of the verifiable data registry and some privacy considerations concerning data leakage and correlation [SLC19]. Since such a registry can be located at any storage location, as already described in subsection 2.3.1 and 2.3.2, the writer thinks that a distinction could be made at this point between on-chain (on a blockchain) and off-chain (off a blockchain) storage solutions for revocation. An example of the former is Anoncreds v1, which implements such a registry on-chain that can be used to retrieve values used for the revocation process [Ha18, Hy18]. In contrast to this are off-chain solutions such as the Revocation List 2020 specification (see figure 2.10), which is managed by the W3C Credentials Community Group and currently an unofficial draft [LS21a].

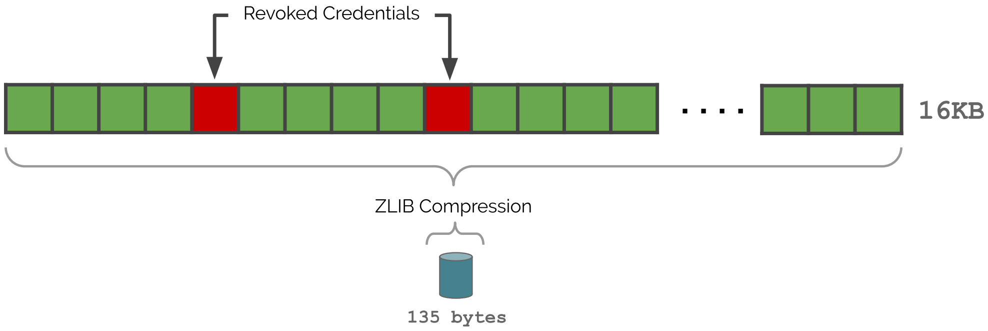


Figure 2.10: Workings of Revocation List 2020 (extracted from [LS21a])

It is a revocation mechanism based on a compressed bitstring that is compatible with existing architectures on the Internet. The issuer of VCs maintains and publishes its own revocation list (bitstring) for its credentials, which contains their status. Each credential is assigned a position in the list whose bit value describes its status (1 = revoked, 0 = not revoked). The specification advises a default bit string size of 16 KB, which can then hold the status of 131,072 credentials. However, since in most cases the majority of credentials has not been revoked and therefore have the value 0, the size can be reduced to a few hundred bytes using compression methods such as ZLIB. If a verifier now wants to retrieve the status of a credential, it retrieves the revocation list from the issuer, which contains the status of over 100,000 credentials. It can decompress this and check the position from the credential in the bit string to check the revocation status. The advantages of this approach can be summarized as follows: [LS21a]

- Efficient: The compression of the bit string uses little storage space and bandwidth.

- Privacy: The status of a credential can be hidden among several others. This makes correlation difficult for issuers, since an entire list is retrieved at any time, instead of the state of a single credential. If higher privacy is needed, the length of the bit string can be increased accordingly. Furthermore, only a minimum of information is published, so no data is published outside of an anonymous status.

- Compatibility: The list can be hosted by an issuer directly or distributed through a content distribution network, which further increases privacy since the issuer no longer handles requests directly.

For its implementation, the issuer must publish the compressed bitstring in a Verifiable Credential signed by itself. The data model can be seen exemplary in listing 2.4.

```
{
    "@context": [...],
    "id": "https://example.com/credentials/status/3",
    "type": ["VerifiableCredential", "
   RevocationList2020Credential"],
    "issuer": "did:example:12345",
    "credentialSubject": {
      "id": "https://dmv.example.gov/credentials/status/3#94567",
      "type": "RevocationList2020",
      "encodedList": "H4sIAAAAAAAAAA-3
   BMQEAAADCoPVPbQsvoAAAAAAAAAAAAAAA..."
    },
    "proof": { ... }
}
```

Listing 2.4: Example RevocationList2020 credentials (edited and extracted from [LS21a])

The specification requires that this credential must contain a `id` that references the id within the credential whose status is being represented. In addition, the type `RevocationList2020Credential` must be included, as well as the type and encoded list under `credentialSubject`. Correspondingly, the specification also describes how the published revocation list can be correctly referenced as a revocation method in an VC issued by the issuer. This is illustrated in listing 2.5.

```
{
    "@context": [...],
    "id": "https://example.com/credentials/23894672394",
    "type": ["VerifiableCredential"],
    "issuer": "did:example:12345",
    "credentialStatus": {
      "id": "https://dmv.example.gov/credentials/status/3#94567"
      "type": "RevocationList2020Status",
      "revocationListIndex": "94567",
      "revocationListCredential": "https://example.com/credentia
    },
    "credentialSubject": {
      "id": "did:example:6789",
       "type": "Person"
    },
    "proof": { ... }
}
```

Listing 2.5: Example VC referencing a RevocationList2020 credential (edited and extracted from [LS21a])

In 2.5, the object `credentialStatus` is the primary object to be considered. In this, the `id` is found in the first place, which corresponds to the id in the `credentialSubject` of the RevocationList2020 credential. The `type` attribute clearly indicates that this credential uses the RevocationList2020 revocation method, followed by the index of the revocation status of the credential in the list (`revocationListIndex`) and a URL leading directly to the VC containing the encoded revocation list (`revocationListCredential`). [LS21a]

RevocationList2020 is a promising approach to implement privacy-preserving revocation of VCs. Despite the status of the specification, there are already possibilities to use this revocation method in production. For example, MATTR offers it to its customers on its SSI platform [MA20a], but it can also be implemented independently using open-source libraries such as `vc-revocation-list-2020` [Di21] from Digital Bazaar.

3 Expert Questionnaire

3.1 Expert Selection

3.2 Questionnaire

3.2.1 Solutions Overview Draft

3.2.2 Questions

3.3 Results

4 Reference Implementation

4.1 Requirements

4.2 Base Implementation

4.3 Architecture

4.4 Solution Integration

4.4.1 Solution Selection

4.4.2 MATTR

4.4.3 Trinsic

4.4.4 Veramo

4.4.5 Azure AD

4.5 Results

5 Evaluation Framework

5.1 Requirements

5.2 Criteria & Questions

5.3 Results

6 Conclusion

Bibliography

[Al16] Allen, Christopher: , The Path to Self-Sovereign Identity, April 2016.

[AL20] Allende López, Marcos: Self-Sovereign Identity: The Future of Identity: Self-Sovereignity, Digital Wallets, and Blockchain. Inter-American Development Bank, September 2020.

[BC18] Borgogno, Oscar; Colangelo, Giuseppe: Data Sharing and Interoperability Through APIs: Insights from European Regulatory Strategy. SSRN Electronic Journal, 2018.

[Be19] Bernabe, J. Bernal; Canovas, J. L.; Hernandez-Ramos, J. L.; Moreno, R. Torres; Skarmeta, A.: Privacy-Preserving Solutions for Blockchain: Review and Challenges. IEEE Access, 7:164908–164940, 2019. Conference Name: IEEE Access.

[Bo12] Borchers, Detlef: , Der Diginotar-SSL-Gau und seine Folgen, January 2012.

[Bo19] van Bokkem, Dirk; Hageman, Rico; Koning, Gijs; Nguyen, Luat; Zarin, Naqib: Self-Sovereign Identity Solutions: The Necessity of Blockchain Technology. arXiv:1904.12816 [cs], April 2019. arXiv: 1904.12816.

[Bo20] Bouras, Mohammed Amine; Lu, Qinghua; Zhang, Fan; Wan, Yueliang; Zhang, Tao; Ning, Huansheng: Distributed Ledger Technology for eHealth Identity Privacy: State of The Art and Future Perspective. Sensors, 20(2):483, January 2020.

[Bu20] Bundesdruckerei: , So funktionieren digitale Identitäten, March 2020.

[Ca05] Cameron, Kim: , The Laws of Identity, May 2005.

[CK01] Clauß, Sebastian; Köhntopp, Marit: Identity management and its support of multilateral security. Computer Networks, 37(2):205–219, October 2001.

[Da21] Davie, Matthew; Gisolfi, Dan; Hardman, Daniel; Jordan, John; O'Donnell, Darrell; Reed, Drummond; Deventer, Oskar van: , 0289: The Trust Over IP Stack, May 2021.

[Di21] Digital Bazaar: , vc-revocation-list-2020, May 2021. original-date: 2020-04-21T21:56:17Z.

[DP18] Dunphy, Paul; Petitcolas, Fabien A.P.: A First Look at Identity Management Schemes on the Blockchain. IEEE Security & Privacy, 16(4):20–29, July 2018.

[DT20] Dib, Omar; Toumi, Khalifa: Decentralized Identity Systems: Architecture, Challenges, Solutions and Future Directions. Annals of Emerging Technologies in Computing, 4(5):19–40, December 2020.

[Eh21] Ehrlich, Tobias; Richter, Daniel; Meisel, Michael; Anke, Jürgen: Self-Sovereign Identity als Grundlage für universell einsetzbare digitale Identitäten. HMD Praxis der Wirtschaftsinformatik, February 2021.

[FCA19] Ferdous, Md Sadek; Chowdhury, Farida; Alassafi, Madini O.: In Search of Self-Sovereign Identity Leveraging Blockchain Technology. IEEE Access, 7:103059–103079, 2019.

[GMM18] Grüner, Andreas; Mühle, Alexander; Meinel, Christoph: On the Relevance of Blockchain in Identity Management. arXiv:1807.08136 [cs], July 2018. arXiv: 1807.08136.

[Ha18] Hardman, Daniel: , 0011: Credential Revocation, 2018.

[Ha19] Hardman, Daniel: , Aries RFC 0005: DID Communication, November 2019. original-date: 2019-05-08T16:49:20Z.

[Ha21] Hardman, Daniel: , DIDComm Messaging Specification, 2021.

[He07] Hevner, Alan R: A Three Cycle View of Design Science Research. 19:7, 2007.

[He20a] Heck, Rouven: , SSI Architecture Stack, 2020.

[He20b] Helmy, Nader: , JWT vs Linked Data Proofs: comparing VC assertion formats, August 2020.

[He20c] Helmy, Nader: , A solution for privacy-preserving verifiable credentials, August 2020.

[Ho20] Homeland Security: , Preventing Forgery & Counterfeiting of Certificates and Licenses – Phase 1 Interoperability Plug Fest Test Plan, May 2020.

[Ho21] Hollington, Jesse: , In a Surprising Twist, Apple Just Launched a Tool to Transfer iCloud Photos to Google Photos (But There's a Catch), March 2021. Section: News.

[Hu14] Hub Culture: , HubID First to Deploy Windhover Principles and Framework for Digital Identity, Trust and Open Data, October 2014.

[Hy18] Hyperledger: , How Credential Revocation Works — Hyperledger Indy SDK documentation, 2018.

[id14] idcubed.org: , ID3 - idcubed.org - The Windhover Principles for Digital Identity, Trust, and Data, November 2014.

[JHF03] Jordan, Ken; Hauser, Jan; Foster, Steven: The Augmented Social Network: Building identity and trust into the next-generation Internet. First Monday, 8(8), August 2003.

[Jo20] John, Anil: , DHS SVIP Blockchain/DLT/SSI Cohort - Multi-Product Phase 1 Interop Artifacts/ Scaffolding / Information, December 2020.

[Jo21] Johnson, Joseph: , Internet users in the world 2021, July 2021.

[Ka15] Kalodner, Harry; Carlsten, Miles; Ellenbogen, Paul; Bonneau, Joseph; Narayanan, Arvind: , An empirical study of Namecoin and lessons for decentralized namespace design, 2015.

[Ko21] Koppenhöfer, Laura: , Kabinettsbeschluss: Handy-Personalausweis ab September, October 2021.

[Kr19] Krempl, Stefan: , E-Government-Studie: Bundesbürger nutzen Personalausweis mit eID kaum, October 2019.

[Ku20] Kuperberg, Michael: Blockchain-Based Identity Management: A Survey From the Enterprise and Ecosystem Perspective. IEEE Transactions on Engineering Management, 67(4):1008–1027, November 2020.

[Li20] Liu, Yang; He, Debiao; Obaidat, Mohammad S.; Kumar, Neeraj; Khan, Muhammad Khurram; Raymond Choo, Kim-Kwang: Blockchain-based identity management systems: A review. Journal of Network and Computer Applications, 166, September 2020.

[Lo20] Lomas, Natasha: , Facebook's photo porting tool adds support for Dropbox and Koofr, March 2020.

[LS21a] Longley, Dave; Sporny, Manu: , Revocation List 2020, April 2021.

[LS21b] Looker, Tobias; Steele, Orie: , BBS+ Signatures 2020, June 2021.

[Ma12] Marlinspike, Moxie: , What is "Sovereign Source Authority"?, February 2012.

[MA20a] MATTR: , Adding support for revocation of Verifiable Credentials, October 2020.

[MA20b] MATTR: , Intro to ZKPs using BBS+ signatures, July 2020.

[Ma21] Mayer, Peter; Zou, Yixin; Schaub, Florian; Aviv, Adam J: "Now I'm a bit angry:" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them. USENIX Security Symposium, 30:18, 2021.

[MG20] Meinel, Christoph; Gayvoronskaya, Tatiana: Blockchain: Hype oder Innovation. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020.

[Mi20] Minor, Jens: , Google Fotos: Praktisches Export-Werkzeug - so lassen sich alle Facebook-Fotos & Videos zu Google übertragen - GWB, December 2020.

[NJ20] Naik, Nitin; Jenkins, Paul: uPort Open-Source Identity Management System: An Assessment of Self-Sovereign Identity and User-Centric Data Platform Built on Blockchain. In: 2020 IEEE International Symposium on Systems Engineering (ISSE). IEEE, Vienna, Austria, pp. 1–7, October 2020.

[No21] Noor, Poppy: , Should we celebrate Trump's Twitter ban? Five free speech experts weigh in, January 2021. Section: US news.

[OR20] Oesch, Sean; Ruoti, Scott: That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers. In: USENIX Security Symposium. pp. 2165–2182, 2020.

[Or21] Ormandy, Tavis: , Password Managers., June 2021.

[PR21] Preukschat, Alex; Reed, Drummond: Self-Sovereign Identity: Decentralized digital identity and verifiable credentials. Manning, 1 edition, May 2021.

[Ri21] Risk Based Security: , Data Breach QuickView - June 2021, July 2021. Section: Featured.

[SLC19] Sporny, Manu; Longley, Dave; Chadwick, David: , Verifiable Credentials Data Model 1.0, November 2019.

[SNA21] Soltani, Reza; Nguyen, Uyen Trang; An, Aijun: A Survey of Self-Sovereign Identity Ecosystem. Security and Communication Networks, 2021:1–26, July 2021.

[SNE20] Satybaldy, Abylay; Nowostawski, Mariusz; Ellingsen, Jørgen: Self-Sovereign Identity Systems: Evaluation Framework. In (Friedewald, Michael; Önen, Melek; Lievens, Eva; Krenn, Stephan; Fricker, Samuel, eds): Privacy and Identity Management. Data for Better Living: AI and Privacy, volume 576, pp. 447–461. Springer International Publishing, Cham, 2020. Series Title: IFIP Advances in Information and Communication Technology.

[Sp21] Sporny, Manu; Longley, Dave; Sabadello, Markus; Reed, Drummond; Steele, Orie; Allen, Christopher: , Decentralized Identifiers (DIDs) v1.0, March 2021.

[St17] Steel, Amber: , LastPass Reveals 8 Truths about Passwords in the New Password Exposé, November 2017.

[St21] Strüker, Dr Jens; Urbach, Dr Nils; Guggenberger, Tobias; Lautenschlager, Jonathan; Ruhland, Nicolas; Sedlmeir, Johannes; Stoetzer, Jens-Christian; Völter, Fabiane: Grundlagen, Anwendungen und Potenziale portabler digitaler Identitäten. p. 52, June 2021.

[Sw21] Swinhoe, Dan: , The 15 biggest data breaches of the 21st century, January 2021.

[To17] Tobin, Andrew; Reed, Drummond; Windley, Foreword Phillip J; Foundation, Sovrin: The Inevitable Rise of Self-Sovereign Identity. p. 24, 2017.

[To21] Toth, Marek: , You should turn off autofill in your password manager | Marek Tóth, July 2021.

[WH07] Wilde, Thomas; Hess, Thomas: Forschungsmethoden der Wirtschaftsinformatik. p. 8, 2007.

[WO01] Wilcox-O'Hearn, Zooko: , Names: Distributed, Secure, Human-Readable: Choose Two, October 2001.

[Wo17] World Bank: , 1.1 Billion 'Invisible' People without ID are Priority for new High Level Advisory Council on Identification for Development, 2017.

[Yi21] Yildiz, Hakan: , Layers of SSI Interoperability, January 2021.

[Yo21] Young, Kaliya: Verifiable Credentials Flavors Explained. p. 21, 2021.

[Zu21] Zundel, Brent: , Why the Verifiable Credentials Community Should Converge on BBS+, March 2021. Section: Thought Leadership.

A Appendix