BLOCKCHAIN-ENABLED SELF-SOVEREIGN IDENTITY An exploratory study into the concept Self-Sovereign Identity and how blockchain technology can serve the fundamental basis

Thesis · December 2017

DOI: 10.13140/RG.2.2.17693.82406

1 author:

Marvin Van Wingerde
Tilburg University
1 PUBLICATION **0** CITATIONS

SEE PROFILE

MASTER THESIS

Master of Science in Information Management

BLOCKCHAIN-ENABLED SELF-SOVEREIGN IDENTITY

An exploratory study into the concept Self-Sovereign Identity and how blockchain technology can serve the fundamental basis.

M.E.M. VAN WINGERDE

Tilburg University, School of Economics and Management

Iquality Business Solutions

December 13, 2017

ABSTRACT

Self-sovereign identity management is a new paradigm, ignited by blockchain technology. The field of identity management currently faces issues in multiple areas. Identity theft and data breaches are not uncommon, and are often the result of insecure identity management practices. Identity management is largely centralized, harming the privacy of subjects. The general public has to trust large corporations and governments to correctly handle their personal data. In contrast to centralised identity management, the self-sovereign identity paradigm places the subject central to their own administration. To facilitate a self-sovereign identity system, a decentralized information system is needed. Blockchain technology might fulfil this need, as it makes use of a distributed ledger, on which consensus about its state is reached by decentralised cryptographic protocols. However, as with most emergent technologies, there is a lack of scientific research into the concept of self-sovereign identity. Therefore, this research answered the following research question: “what are the system requirements for a regulatory compliant Self-Sovereign Identity information system, and how can blockchain technology solutions serve as a solid foundation?"

To be able to formulate an answer to this question, a set of requirements and constraints have been developed by using design science research. Two design iterations have been conducted. In the first iteration, an initial set of requirements have been developed through introspection and semi-structured interviews with industry experts from multiple industries. In the second iteration, the set of requirements and constraints have been analysed and optimised, and sent to the same industry experts for peer-review. After the set of requirements and constraints were developed, blockchain technology is evaluated using partial requirement satisfaction. Lastly, two practical and current state-of-the-art implementations, namely Sovrin and uPort, have been evaluated using binary requirement satisfaction. At this moment, Sovrin better satisfies the requirements.

Based on the findings, we can conclude that indeed, blockchain can serve as a solid foundation for a self-sovereign identity system. However, it became evident that blockchain is not an absolute solution, additional technology is needed. Blockchain does aid in maintaining integrity of personal data and providing subjects the freedom to privately exchange verifiable claims about their identity, with a lower need for trust in large institutions.

Keywords: identity management, self-sovereign identity, blockchain, distributed ledger technology, verifiable claims, system requirements, GDPR, eIDAS, PSD2, Sovrin, uPort.

TABLE OF CONTENTS

ABSTRACT ..... **I**

TABLE OF FIGURES ..... **IV**

LIST OF ABBREVIATIONS ..... **V**

1 INTRODUCTION **1**

1.1 RESEARCH MOTIVATION 1

1.2 PROBLEM INDICATION 2

1.3 PROBLEM STATEMENT 3

1.4 RESEARCH APPROACH 3

1.5 THEORETICAL AND SOCIAL RELEVANCE 4

1.6 STRUCTURE OF THE REPORT 5

2 LITERATURE REVIEW **6**

2.1 IDENTITY MANAGEMENT 7

2.2 BLOCKCHAIN TECHNOLOGY 17

2.3 REGULATORY ENVIRONMENT 27

2.4 PROPOSITIONS 30

3 METHODOLOGY **31**

3.1 REQUIREMENTS ENGINEERING 31

3.2 DESIGN ACTIVITIES 32

3.3 EVALUATING BLOCKCHAIN TECHNOLOGY FOR SELF-SOVEREIGN IDENTITY 34

4 OPERATIONALIZING SELF-SOVEREIGN IDENTITY **35**

4.1 ONGOING DEVELOPMENT OF SELF-SOVEREIGN IDENTITY FRAMEWORKS 35

4.2 SYSTEM REQUIREMENTS FOR SELF-SOVEREIGN IDENTITY 38

5 BLOCKCHAIN TECHNOLOGY FOR SELF-SOVEREIGN IDENTITY **52**

5.1 SATISFACTION OF REQUIREMENTS BY USING BLOCKCHAIN TECHNOLOGY 52

5.2 ANALYSIS OF EVALUATION RESULTS 60

6 COMPARATIVE ANALYSIS OF SSI IMPLEMENATATIONS **63**

6.1 SOLUTION DESCRIPTION: SOVRIN 63

6.2 SOLUTION DESCRIPTION: UPORT 69

6.3 REQUIREMENT SATISFACTION OF SOVRIN AND UPORT 73

7 CONCLUSION **77**

8 DISCUSSION AND RECOMMENDATIONS.....79

9 LIMITATIONS AND FURTHER RESEARCH.....80

REFERENCES 82

APPENDIX A. GENERAL DATA PROTECTION REGULATION (GDPR) 89

APPENDIX B. EXTENDED TEN PRINCIPLES OF SELF-SOVEREIGN IDENTITY 92

APPENDIX C. EXAMPLE OF A VERIFIABLE CLAIM.....94

TABLE OF FIGURES

FIGURE 1: DESIGN SCIENCE RESEARCH, ADAPTED FROM HEVNER ET AL. (2004, P. 80) 6
FIGURE 2: GLOBAL INTERNET USAGE AND GROWTH (INTERNATIONAL TELECOMMUNICATIONS UNION, 2017) 7
FIGURE 3: IDENTITY MANAGEMENT PROCESSES, BY HARRIS & STONEBRAKER (2015, P. 725) 10
FIGURE 4: IDENTITY MANAGEMENT STAKEHOLDERS, BY BERTINO ET AL. (2011, P. 29) 11
FIGURE 5: CENTRALIZATION, DECENTRALIZATION, AND DISTRIBUTION. BY BARAN (1964, P. 4) 12
FIGURE 6: DATA STRUCTURE OF A BITCOIN BLOCK, AS DESCRIBED BY NAKAMOTO (2008, P. 4) 21
FIGURE 7: TRANSACTIONS ON A BLOCKCHAIN, BY NAKAMOTO (2008, P. 2) 21
FIGURE 8: SELF-SOVEREIGN IDENTITY ARCHITECTURE 37
FIGURE 9: SOVRIN ENTITY ARCHITECTURE (BEST ET AL., 2017, P. 7) 65
FIGURE 10: SOVRIN IDENTIFIER TYPOLOGY (BEST ET AL., 2017, P. 8) 65

TABLE 1: SYSTEM REQUIREMENTS ENGINEERING ACTIVITIES 31
TABLE 2: STAKEHOLDERS TO BE INTERVIEWED 33
TABLE 3: GDPR COMPLIANCE VALIDATION CRITERIA 38
TABLE 4: PSD2 COMPLIANCE VALIDATION CRITERIA 39
TABLE 5: EIDAS COMPLIANCE VALIDATION CRITERIA 39
TABLE 6: USE-CASE DESCRIPTION - ESTABLISH A DIGITAL IDENTITY 40
TABLE 7: USE-CASE 1 - FUNCTIONAL REQUIREMENTS 40
TABLE 8: USE-CASE 1 - NON-FUNCTIONAL REQUIREMENTS 41
TABLE 9: USE-CASE DESCRIPTION - ISSUE VERIFIABLE CLAIMS 42
TABLE 10: USE-CASE 2 - FUNCTIONAL REQUIREMENTS 42
TABLE 11: USE-CASE 2 - NON-FUNCTIONAL REQUIREMENTS 43
TABLE 12: USE-CASE DESCRIPTION - ASSERTS VERIFIABLE CLAIMS 45
TABLE 13: USE-CASE 3 - FUNCTIONAL REQUIREMENTS 45
TABLE 14: USE-CASE 4 - NON-FUNCTIONAL REQUIREMENTS 46
TABLE 15: USE-CASE DESCRIPTION - REVOKE A CLAIM 47
TABLE 16: USE-CASE 4 - FUNCTIONAL REQUIREMENTS 47
TABLE 17: USE-CASE DESCRIPTION - ENTITY AUTHENTICATION 48
TABLE 18: USE-CASE 5 - FUNCTIONAL REQUIREMENTS 48
TABLE 19: USE-CASE 5 - NON-FUNCTIONAL REQUIREMENTS 49
TABLE 20: USE-CASE DESCRIPTION - AUTHORIZE DATA ACCESS 50
TABLE 21: USE-CASE 6 - FUNCTIONAL REQUIREMENTS 50
TABLE 22: USE-CASE 6 - NON-FUNCTIONAL REQUIREMENTS 51
TABLE 23: SATISFACTION OF REQUIREMENTS BY USING BLOCKCHAIN TECHNOLOGY 53
TABLE 24: REQUIREMENTS SATISFACTION RESULTS 60
TABLE 25: ADDED VALUE OF BLOCKCHAIN TECHNOLOGY PER SSI PRINCIPLE 61
TABLE 26: REQUIREMENT SATISFACTION OF SOVRIN AND UPORT 73

LIST OF ABBREVIATIONS

DID	Decentralised Identifier
DLT	Distributed Ledger Technology
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
EIDAS	Electronic Identification, Authentication and Trust Services
FR	Functional Requirement
GDPR	General Data Protection Regulation
IDM	Identity Management
IDMS	Identity Management System
IPFS	Inter-Planetary File System
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
NFR	Non-Functional Requirement
NIST	National Institute for Standards and Technology
PKI	Public Key Infrastructure
POS	Proof of Stake
POW	Proof of Work
PSD2	Revised Payment Services Directive
RSA	Rivest, Shamir and Adleman (Algorithm)
SHA	Secure Hash Algorithm
SK	Secret Key
SSI	Self-Sovereign Identity
TX	Transaction
VC	Validation Criterion
VK	Verification Key

1 INTRODUCTION

1.1 Research motivation

Blockchain technology has the potential to be the foundation of a new technological paradigm, much like the internet was in early 2000. Where the internet enabled a global exchange of information, a blockchain enables global peer-to-peer exchange of value. The technology is now at the peak of its expectations, promising radical change in mainly established industries. One of the prominent use-cases of blockchain technology is self-sovereign identity management.

The field of identity management has rapidly evolved over the last decade. Due to the ever-increasing online presence of the public, and the global transactions which are made, it has become vital to identify and authenticate digital identities. Moreover, there is a trend of personal data accumulation by large corporations, leaving their databases vulnerable to malicious attackers and users not feeling in control of their own data. During the six-month period that this research has been conducted, multiple large-scale attacks have been conducted against companies processing large amounts of personal data. Perhaps the most devastating example is that of Equifax, who lost over 140 million records of their clients, including social security numbers. These recurring mishaps stress the need for radical advances in the field of identity management.

Identity management faces serious challenges in security, privacy, usability and interoperability. Blockchain technology might be the silver bullet to tackle many of these issues, however, inherent to a potentially disruptive technology, there is a need for scientific research (Pilkington, 2015; Swanson, 2015). By conducting this research, I aim make a contribution to the theoretical and social knowledge base of blockchain-enabled self-sovereign identity management.

This research has been conducted at Iquality Business Solutions. Iquality, established in 1994, is an IT-solutions provider. Iquality designs, develops, and supports software, applications, and websites – based on the needs of their customers. The company conducts projects, offers multidisciplinary teams as a service, organizes knowledge-sharing labs, and offers outsourcing and training programs. Iquality operates with about forty people, collectively generating four million euro in revenue annually. Iquality's core competences lie in the fields of data management, content management, IT-integration, front- and back-end development, business intelligence, and artificial intelligence.

1.2 Problem indication

Proving who you claim to be is of vital importance in today's digital world. Being able to authenticate yourself, and trusting that counterparties are who they claim to be, is a day-to-day concern for people and business. People make use of multiple digital identity service providers, from both public and private institutions. In the public sector, the government issues your proof-of-existence, in the form of a social security number and a passport. The Driver and Vehicle Licensing Agency issues your driver's license, and municipalities issue your proof-of-residence. In the private sector, large corporations like Google and Facebook have established themselves as a central point of personal data collection and storage. Much of the public's personal data is stored on their servers, leading to security and privacy issues. Moreover, private institutions like banks and insurers have access to your financial history, and have the authority to prove your financial creditability. This current system of centralized and federative identity management brings multiple issues in the domains of security, privacy, trust, regulation, and usability.

Blockchain technology is posited to decentralize identity management, putting users in control of their own data. Large organizations, both public and private, are researching how blockchain technology might be used to solve the issues of identity management. Multiple proof-of-concepts, minimum-viable products, and working groups have been established, but there is still a lack of rigorous theoretical validation. Therefore, the theoretical relevance of this research lies in formulating what the requirements and constraints are for a self-sovereign identity management solution. This enables us to test to what extent the properties and functionalities of blockchain technology meet the discovered requirements and constraints.

Moreover, pressing regulatory changes in the European Union are due for 2018. Over the last three years, the General Data Protection Regulation, the Revised Payment Services Directive, and the electronic Identification, Authentication and Trust Services have been accepted. These will all become effective over the course of 2018. All three regulations significantly affect the way personal data must be handled, and how identities interact with each other. Combined with the need for a change in identity management, compliance with these regulatory advances should be taken into account.

Iquality wants to sustain a competitive position in the market. Inherent to the rapidly changing environment of information technology and systems, keeping up with the latest and most promising technologies is of key importance. Management has recognized blockchain technology as potentially disruptive, and wants to deeper explore what the fundamentals of the technology are, and how they might be able to utilize it. The results of this research might be

immediately applicable to their in-house start-up, a project concentrated on a shared mobility platform.

1.3 Problem statement

Identity management is currently non-sovereign. Data is fragmented, users do not feel in control of their own data, and multiple security and privacy issues exist. Sharing personally identifiable information is a great concern in managing privacy, protecting data, and complying with regulations (Maler & Reed, 2008). Blockchain technology is rapidly gaining traction (Panetta, 2017), and could provide a solution to the identity management problem.

Multiple companies and research groups have developed proof-of-concepts and early-stage products for blockchain-enabled identity management, but they have not been thoroughly analysed and have not yet been adopted by the public. One of the problems is that the theoretical base for blockchain technology is sparse, and little scientific work has been published on how blockchain technology might enable a self-sovereign identity information system. A deeper insight in regulatory compliant requirements and constraints of such a system is needed.

To acquire this insight, the following research question has been formulated: **"What are the system requirements for a regulatory compliant Self-Sovereign Identity information system, and how can blockchain technology solutions serve as a solid foundation?"**

1.4 Research approach

In the field of Information Systems Research, there are two common research approaches; behavioural science and design science. Behavioural science aims to justify and prove hypotheses or theories, while design science focuses on building and evaluating artefacts. Hevner et al. (2004) argued that the two paradigms are complementary, and should be jointly utilized in what they call Design Science Research. Nunamaker et al. (1991) argued that the process of constructing and exercising innovative IT artefacts enable design-science researchers to understand the problem addressed by the artefact and the feasibility of their approach to its solution. The nature of the described problem for this research lies in not fully understanding the problem addressed by existing artefacts, partly due to unstable and ill-defined requirements and constraints. Due to this problem, existing artefacts cannot be rigorously evaluated. To solve this problem, a Design Science Research approach is taken. On the one hand, this will enable us to evaluate existent artefacts, on the other hand, it will serve as a foundation to build new artefacts.

Derived from the three-cycle view of Design Science Research as defined by Hevner et al. (2007), the research approach to design the artefact will consist of the following three cycles:

1. Relevance cycle: define the problem relevance and describe the environmental context, through an extensive literature review.
2. Rigor cycle: select the appropriate techniques for the build and evaluate activities. Elicit new insights through interviews with industry experts.
3. Design cycle: design a viable artefact in the form of system requirements; a set of constraints and functional and non-functional requirements.

After designing the artefact, it will immediately be applied to determine how blockchain technology might serve as a solid foundation for self-sovereign identity. Satisfaction levels for the identified requirements will be allocated to each requirement and constraint. By applying the artefact, we can both refine the set of requirements and use it to determine the added value of blockchain technology. Lastly, the requirement satisfaction level of two operational self-sovereign identity solutions will be determined, to determine to what extent current implementations of blockchain-enabled self-sovereign identity satisfy the requirements and constraints.

1.5 Theoretical and social relevance

Due to the early stage of blockchain technology research, the number of academic articles published in scientific journals is scarce. To illustrate this statement, querying "blockchain" on Google Scholar returns seventeen thousand results, while querying "digital identity management" returns two million results. A substantial amount of research done on blockchain by developers and research groups is published in online communities, such as GitHub, Reddit, and various blogs like Medium. By conducting this research, I aim to add to the relatively undersized theoretical foundation of blockchain and identity management. An understanding of self-sovereign identity management, enabled by blockchain technology, would be of valuable theoretical relevance.

As indicated in §1.2, identity management is a domain which people and business interact with on a near daily basis. Changing the identity management model from a centralized or federated approach to a decentralized and self-sovereign approach, would have great implications for society. Putting people in control of their own data, letting them exchange personal data and attributes on their own terms, is an aspiring goal. Although this research aims to solve and provide

insight in a small piece of the total solution, it is socially relevant to understand where issues in security, privacy and trust lie. The Netherlands has initiated the National Blockchain Coalition – a joint operation between Dutch public and private institutions – to research and develop a solid identity framework. The results of this study could be immediately applicable to self-sovereign identity frameworks. A self-sovereign identity solution that adheres to the identified requirements and constraints will change how people and businesses are identified, authenticate each other, and authorize the sharing of their data.

1.6 Structure of the report

The report will continue with the literature review in chapter 2. It is divided into four parts, a review of the current state of identity management, an overview of blockchain technology, an overview of the upcoming regulatory environment, and a set of propositions that follow from the literature review. Chapter 3 will describe the methodology, based on which the findings have been gathered and analysed. Next, in chapter 4, a set of system requirements and constraints are described, based on interviews with and validation by multiple external stakeholders. Based on this, in chapter 5 the satisfaction levels of the requirements and constraints by using blockchain technology is determined. In chapter 6, two practical implementations – Sovrin and uPort – of blockchain-enabled self-sovereign identity are described and evaluated against the requirements and constraints. Next, in chapter 7, the conclusions and an answer to the research question are presented. In chapter 8, the results and conclusions are discussed, and recommendations presented to the company. Lastly, in chapter 9, research limitations and future research directions are discussed.

2 LITERATURE REVIEW

The goal of the literature review is to describe the current environment, according to Hevner's research framework, as displayed in figure 1 below:

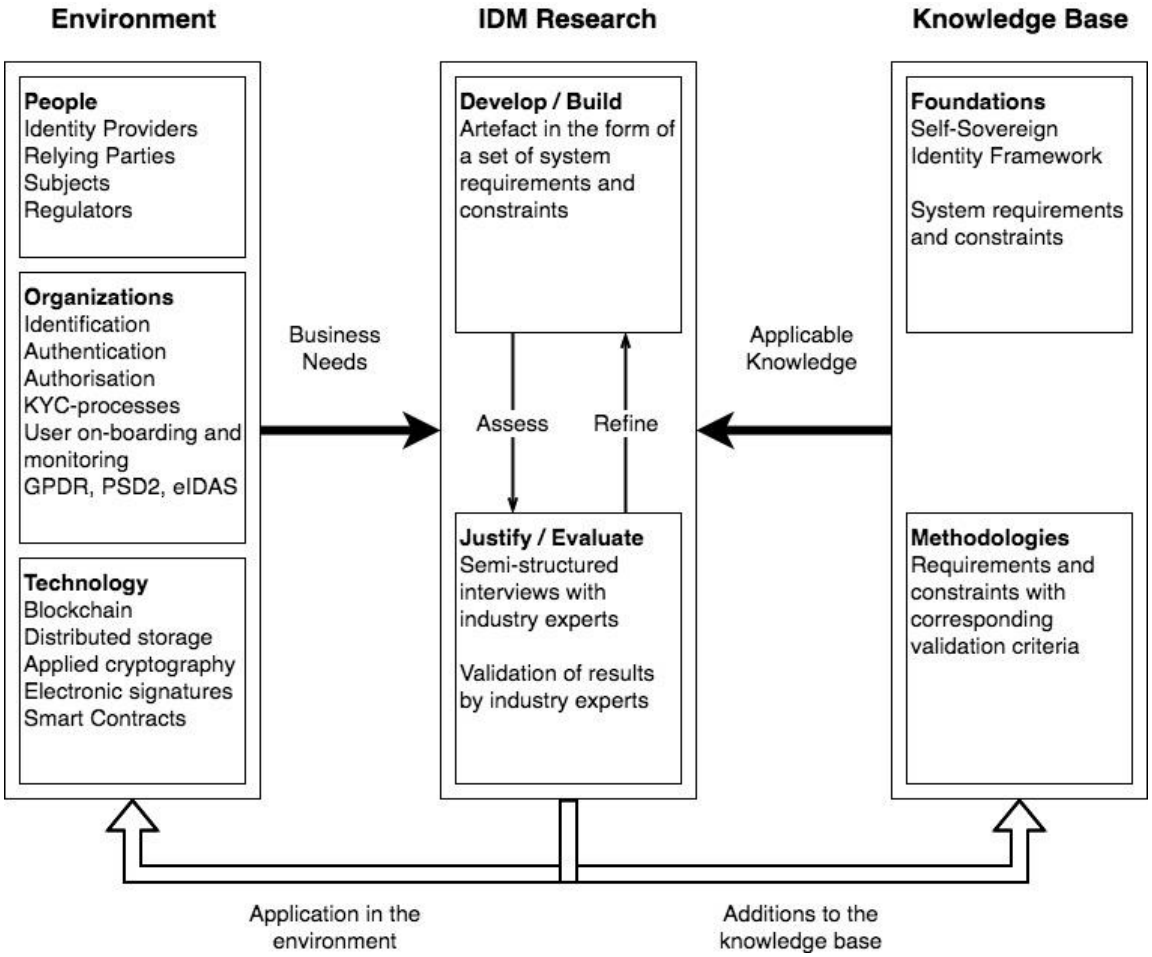

Figure 1: Design Science Research, adapted from Hevner et al. (2004, p. 80)

We will start by describing the current state of affairs within the field of Identity Management. Following that, blockchain technology, its properties, and its challenges will be described. Next, the European regulatory environment will be investigated. Hereafter, we will investigate which challenges within the Identity Management domain could be solved with blockchain technology.

2.1 Identity Management

Since the the year 2000, the global online presence of the public has risen considerably, as shown in figure 2 below:

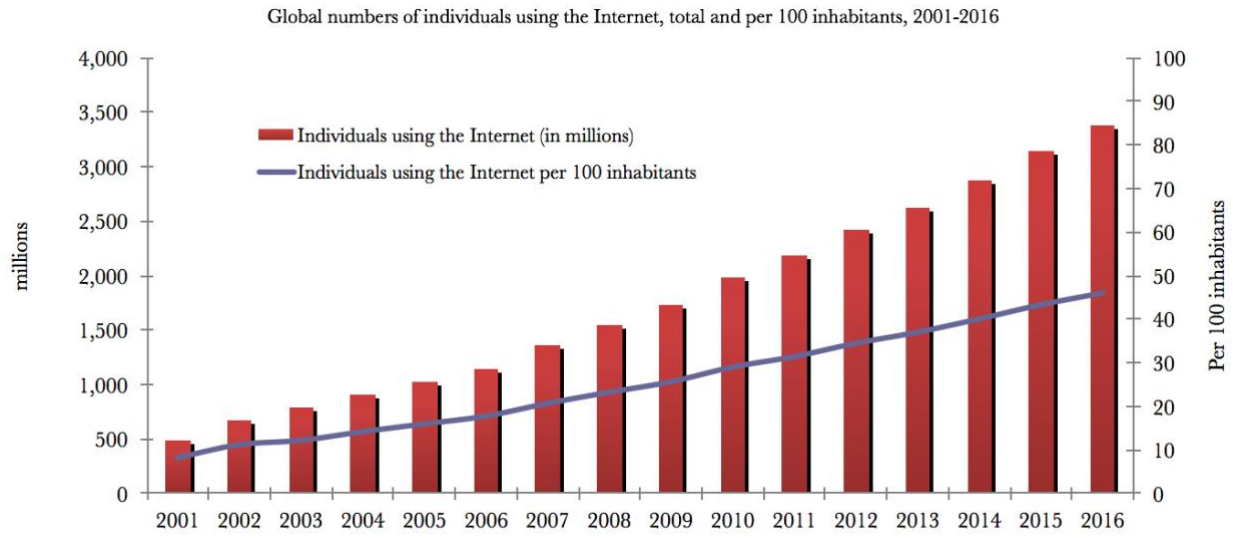

Figure 2: Global internet usage and growth (International Telecommunications Union, 2017)

Following this trend, the need for digital identity management became highly important. When communicating or transacting via Web 2.0, knowing the identity of the counterparty is often the first requirement (Jøsang & Pope, 2005). People want user-friendly and personalized services, which requires organizations to store personal data. This trend has led to organizations realizing the financial potential of this personal data, and finding ways to monetize it (Narayanan & Shmatikov, 2010). A prime example of how they accomplish this, is by providing internet users personalized advertisements and services.

The ever-increasing online presence of people and organizations has made digital identity management a fundamental domain (Whitley, Gal, & Kjaergaard, 2014). Many challenges in the fields of security, privacy, usability and trust are still present.

2.1.1 Definitions

To be able to properly define self-sovereign identity management, we start by defining what a digital identity is. In their most recent guidelines for Digital Identity, published in June 2017, the National Institute of Standards and Technology (NIST) defines a digital identity as follows (Fenton, Danker, Greene, & Theofanos, 2017, p. 15): "a digital identity is the unique representation of a subject engaged in an online transaction". The NIST further states that a

digital identity is always unique in the context of a digital service, but does not necessarily need to uniquely identify the subject in all contexts. In this definition, a subject is a mundane persona, organization, or asset, which is thus digitally represented. The relationship between the mundane subject and their digital identity is a 1:m relationship.

The field of Identity Management (IdM) is receiving increasingly more interest in literature (Halperin, 2006). Multiple definitions of identity management exist, but the one which best suits this study is found in the book "Identity Management: Concepts, Technologies, and Systems", where identity management is defined as "to maintain the integrity of identities through their life cycles in order to make the identities and their related data available to services in a secure and privacy-protected manner" (Bertino & Takahashi, 2011, p. 23).

When discussing self-sovereign identity, the core lies in users having the sole authority to manage their own identity. There appears to be no consensus about the definition of a self-sovereign identity in literature, however, multiple articles describe their view of its main characteristics. Christopher Allen, a researcher and speaker in the field of identity management and blockchain technology, published a whitepaper (Allen, 2016) to describe his vision of a self-sovereign identity. He states that "for identity management to be self-sovereign, the user must be central to the administration of their identity."

2.1.2 Data properties of a digital identity

A digital identity consists of different types of data. The International Telecommunication Union (ITU) – an agency within the United Nations – groups data associated with an entity into three categories (ITU-T, 2009); identifiers, credentials, and attributes. In the following paragraphs, each of these will be discussed separately.

2.1.2.1 Identifiers

An identifier is a random or non-random combination of bits (Pfitzmann & Kiel, 2008) which uniquely identifies an entity in a specific context (Camp, 2004). It is important to note, that the identifier does not have to be universally unique, but only within its context. For example, a URL is unique within the Domain Name System, but not within every database. An identifier is used in conjunction with one or more credentials to claim authenticity of the identity.

Creating or issuing identifiers should include three important facets (Harris & Stonebraker, 2015); uniqueness, non-descriptiveness, and issuance methods. That is, the identifier must be unique in its context, should not describe the role or purpose of the identity,

and the issuance method of the identity can be physical or digital. Eriksson & Ågerfalk (2010) provide guidelines on the design of identifiers, claiming that the design of identifiers should be based on technical, usage, institutional, and information infrastructure aspects.

2.1.2.2 Credentials

Credentials prove the authenticity of an identifier. ITU (2009, p. 3) defines a credential as "an identifiable object that can be used to authenticate the claimant is what it claims to be and to authorize the claimant's access rights". Since credentials can and should be used in different scenarios, NIST categorized credentials into three levels (Grassi et al., 2017):

1. Primary identity credentials: a by-product of significant life events, such as birth and marriage, which are typically only issued once.
2. Secondary identity credentials: are issued in response to a request for authorization to perform an action or demonstrate proof of affiliation.
3. Tertiary identity credentials: are issued by an authority or organization for a limited purpose, with identity verification and proofing requirements varying enormously.

2.1.2.3 Attributes and claims

Attributes are the set of data that collectively make up the digital identity. A subset of attributes describes part of the characteristics of the identity, for example a date of birth, an address, or gender. Attributes can de declarative or certified (Laurent & Bouzefrane, 2015). Declarative attributes are declared by the owner of the identity, without proof by other parties. In contrast to declarative attributes, certified attributes are proven to be valid by a third party.

Derived from attributes, statements can be made. For example, derived from an attribute 'Date of Birth', the statement 'Is older than 21' could be generated. One or multiple statements combined together can form a claim. In turn, if a certain claim is verified and signed by an identity provider, it becomes a verifiable claim. Verifiable claims provide a strong method of separating authentication and authorization (Baier et al., 2010).

2.1.3 Identity Management processes

Identity Management generally deals with four different processes, as displayed in figure 3 below (Harris & Stonebraker, 2015); identification, authentication, authorization, and accountability:

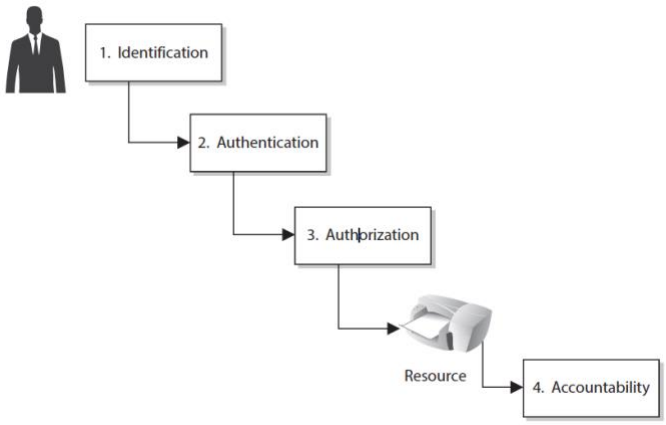

Figure 3: identity management processes, by Harris & Stonebraker (2015, p. 725)

- Identification is the process of stating who you claim to be. This is done by providing your identifier to the counterparty. The counterparty can look up the provided identifier, and may have an internal identifier coupled to it.
- Once an entity has identified themselves, the counterparty cannot trust their claim is valid. For the entity to prove who they claim to be, they must authenticate themselves, by providing credentials. NIST defines authentication as (Grassi et al., 2017, p. 2): "the process of determining the validity of one or more authenticators used to claim a digital identity".
- Once a subject has identified itself, and authenticity is verified, the subject might have certain authorizations within the current context. The degree of authorization defines what a subject is allowed to do or access within a certain context.
- Accountability concerns logging identifier activities, to be able to later claim accountability for its actions.

2.1.4 Stakeholders

Within the domain of Identity Management, there are multiple parties involved. Bertino et al. (2011) categorize these stakeholders into four groups; subjects, identity providers, relying parties, and – in some cases – control parties.

The ecosystem of stakeholders is visualized in the figure below:

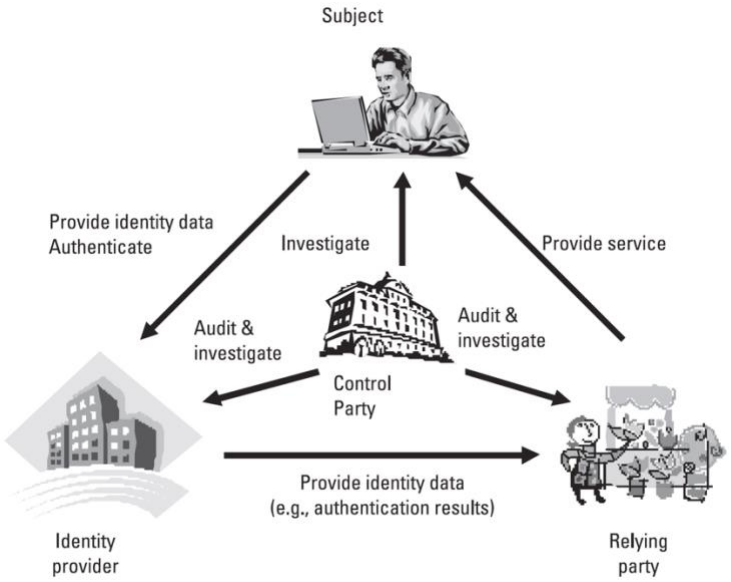

Figure 4: identity management stakeholders, by Bertino et al. (2011, p. 29)

- A subject is the entity who wishes to use his digital identity to transact or interact with a counterparty.
- An identity provider has the authority to issue identifiers to entities, assign attributes to them, and manage validity of the authentication process by managing credentials.
- A relying party provides the subject with a service, and relies on the identity provider to verify the identity of the subject.
- A controlling party is often a governmental or regulatory party who needs to access identity information. They must be able to audit and investigate logs of all other stakeholders.

2.1.5 Identity Management Systems

In §2.1.1, we defined identity management as the maintaining of the integrity of identities through their life cycles to make data available to third parties. To accomplish this, an information system (IS) is needed. An Identity Management System (IDMS) is an IS which

integrates various technologies and processes, in which subjects can manage their identity attributes, authentication factors, and authorization privileges across multiple relying parties (Alkhalifah & D'Ambra, 2015).

2.1.5.1 Three architectural approaches

Since the widespread adoption of the web, IDMS have been subject to multiple architectural approaches. Generally, there are three approaches to IDMS; centralization, decentralization, and distribution. The foundations lie in work on distributed communication networks (Baran, 1964). Figure 5 below visually displays the structure of these approaches:

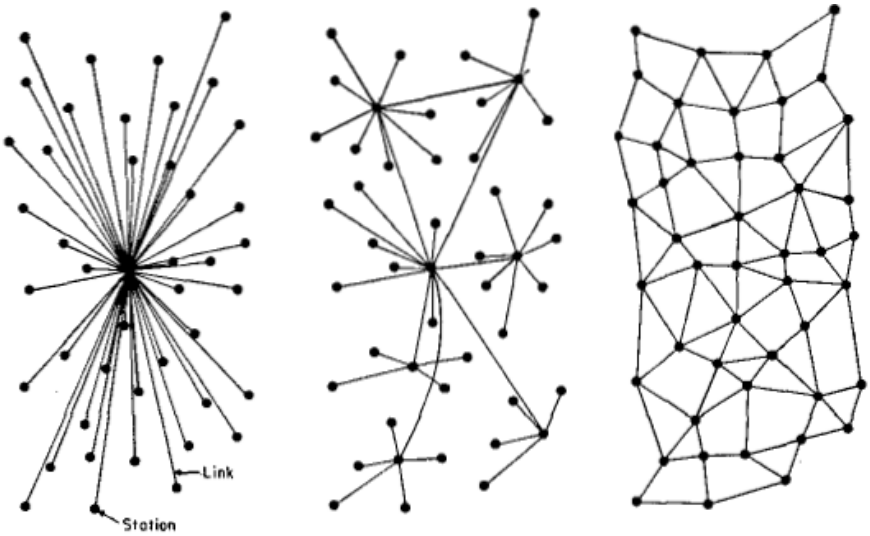

Figure 5: centralization, decentralization, and distribution. By Baran (1964, p. 4)

Traditionally, IDMS have been centrally managed, giving one entity sole power over authentication and authorization functions. In the late 80's and 90's, separate entities had full control over proving the validity of IP addresses and the registration of internet domain names (Allen, 2016). Even today, many identity management systems are centralised – if Facebook ceases to exist, their user's accounts will likely be erased.

Since the year 2000, a shift to decentralized approaches was taken, in the form of federated identity management. In a federation, an identity provider authenticates a subject's credentials, while a service requestor manages the subject's authorizations through an interface with the identity provider. Microsoft was one of the first to introduce a federated identity management solution, by introducing Microsoft Passport in 1999. It enabled users to log-in to multiple sites, by only using their passport. However, this solution was not successful, as it still

put Microsoft at the centre of the federation. Nowadays, Google is an example entity who provides subjects with an account, which they can use to authenticate themselves at multiple relying parties. However, the problem of centralized data management persists in this situation. Personal data associated with a Google account is centrally stored on Google's servers, leading to multiple privacy and security issues.

The last approach, distributed identity management, puts the subject at the centre. In this approach, subjects manage their own data, have possession over their credentials, and choose who they share their attributes with. It removes the need for subjects to trust a central authority, restoring the sovereignty of the subject. Multiple initiatives stimulate development in this model (European Parliament, 2016; Siriwardena, 2017a; Sporney & Longley, 2016). For example, the European GDPR regulation requires subjects to be in control of their personal data. An interesting effort by C. Allen (2016) conceptualizes principles for a self-sovereign identity. These principles are abstract, and arguably need further development and operationalization:

1. **Existence**: users must have an independent existence
2. **Control**: the user must be the ultimate authority of his identity
3. **Access**: a user must always be able to easily retrieve all data within his identity
4. **Transparency**: systems and algorithms must be transparent
5. **Persistence**: identities should last for as long as the user wishes
6. **Portability**: information and services about identity must be transportable
7. **Interoperability**: identities should be as widely usable as possible
8. **Consent**: sharing of data must only occur with the consent of the user
9. **Minimization**: disclosure should involve the minimum amount of data necessary to accomplish the task at hand
10. **Protection**: identity authentication must occur through independent algorithms that are censorship-resistant and force-resilient

2.1.5.2 Standards and technologies

As the field of identity management matured, a call for standardization has been made. Over the past two decades, multiple technologies have established themselves as foundations within IDMS. The list on the following page will cover Security Assertion Mark-up Language (SAML), Single Sign-On (SSO), OpenID, and Open Authorization (OAuth).

- Security Assertion Mark-up Language: the first specification of SAML, developed by the OASIS Security Services Technical Committee (SSTC), dates from 2002. SAML consists of a set of technical specifications to manage identities based on identity federation (Bertino & Takahashi, 2011). By using XML formats, SAML standardizes the way data between relying parties and identity providers are exchanged.
- Single Sign-On: a protocol which enables federated identity management by letting users authenticate themselves once, and are then able to access multiple relying parties (Armando, Carbone, Compagna, Cuellar, & Tobarra, 2008). It combats the need for subjects to manage multiple username/password combinations. SSO is often based on SAML 2.0, for example by Google Apps.
- OpenID: created in 2005 by the OpenID foundation, OpenID lets subjects use an existing account to sign in to multiple websites. Subjects can choose an identity provider, who will authenticate your credentials for relying parties. The focus of OpenID is therefore on authentication.
- Open Authorization: the original OAuth protocol was developed in 2006 by a group of industry professionals, and formalized by the IETF in 2007. The OAuth focuses on authorization by letting an application access a resource on behalf of a subject, and is widely used by social networks (Laurent & Bouzefrane, 2015).

2.1.6 Identity Management challenges

The field of Identity Management faces multiple challenges. These challenges will be divided and addressed in five categories; security, usability, privacy, trust, and implementability.

2.1.6.1 Security

Identity theft is one of the most high-level security issues within current IDMS practices. Malicious attackers gain control of the digital identity of subjects, often for financial reasons. In 2016, a record number of 15.4 million people in the United States have been subject to identity theft (Pascual, Marchini, & Miller, 2017), resulting in losses of USD 16 billion.

Data leaks are another form of common security issues, with the reported number of breaches increasing yearly. In 2016, data breaches increased by 40% over 2015 (Daitch, 2016). Personal information stored on central databases are an interesting target for attackers.

2.1.6.2 Usability

The most standard form of authentication on the web, is using a username and password combination. As the public conducts transaction on multiple websites, it becomes hard for subjects to manage their accounts (Osmanoglu, 2013). A dominant portion of the public re-uses password across multiple websites, resulting in security risks.

A study by Dhamija and Dusseault (2008) points out that the focus of authentication is often on the side of the subject. However, to reduce phishing and hacking attempts, relying parties should also authenticate themselves to the subject. This is not yet a common practice, and should be addressed to guard subjects.

2.1.6.3 Privacy

Correct processing of personally identifiable information (PII) is an important facet of privacy preservation. Companies tend to state that a subject's data will only be shared in a non-identifiable form. In practice, classifying data as PII is ubiquitous, as there is no clear consensus on identifying and non-identifying data (Narayanan & Shmatikov, 2010).

As pointed out earlier, in many situations subject data is centrally stored. This is a direct privacy issue, as the subject has no control over who accesses this data. Employees of the server park might – willingly or forced – get access to PII.

2.1.6.4 Trust

Due to the nature of centralized or federated IDMS, subjects must trust identity providers and relying parties to correctly handle their personal data. Subjects are not in control, and are thus not able to prevent misuse of their sensitive data. Another trust problem in IDMS is how they handle reputation. Ensuring validity of reputation on social profiles is hard, leading to a need for verifiable attestations (Bertino & Takahashi, 2011).

2.1.6.5 Implementability

Implementing Identity Management solutions has been proven to be a complex undertaking. Identity management crosses multiple domains and stakeholders, as has been discussed in §2.1.4. Moreover, the data used by identity management systems is inherently private, thus needs the highest level of security measures. When implementing an identity management system, multiple challenges may arise (Jacobs, 2011; Wiese, 2013):

- Reaching no consensus among stakeholders on standards and technologies
- Not being able to generate a solid business case to convince upper management
- Not obtaining support from top management
- Not taking into account the need for change management
- Ineffective transition planning or phased implementation

2.2 Blockchain Technology

Blockchain technology is posited to drastically change the way we digitally exchange value. Where the internet has enabled us to globally exchange information at a rapid rate, it has not solved the double-spend problem. Currently, each time a peer sends a piece of information to another peer, he sends a copy of that information, instead of the information itself. For this fact, trusted third parties have established themselves as middlemen to guarantee counterparties secure methods of transacting. Common middlemen institutions are banks, insurers, clearing houses and settlement agencies, among others. These trusted third parties have grown to be a centralized power, leaving little room for the public.

Following the global financial crisis of 2007, a pseudonym called Satoshi Nakamoto published a whitepaper (2008) which describes a trustless peer-to-peer cash system. It is called Bitcoin, and is now the dominant cryptocurrency with a total market valuation of more than USD 160 billion. The fundamental breakthrough of Bitcoin, is blockchain technology.

2.2.1 Definition

Being an emergent technology, there is no clear consensus on a formal definition of blockchain technology among academics. Books and academic articles do describe the principles of a blockchain, and its main application areas. Mougayar (2016) describes the fundamental principles of a blockchain from three perspectives. First, from a business view; a blockchain is a network on which value can be digitally exchanged peer-to-peer. Second, from a legal view, a blockchain is a network protocol which can validate transaction without the need for trust in a third party. The third and last view, a technical view, describes blockchain as a back-end database which maintains a publicly visible, immutable distributed ledger.

2.2.2 Distributed Ledger Technology

Distributed ledger technology (DLT) is arguably at the core of a block chain. Essentially, a distributed ledger is a shared database, which a set of nodes – devices running specific software – collectively maintain (Mills et al., 2016). In a blockchain environment, the distributed ledger records transactions, grouped into blocks. These blocks are timestamped, enabling a chronological state of the ledger.

The entities owning the nodes may have different roles. For this reason, distributed ledgers are classified into two categories; permissioned and permissionless ledgers. In the field of data management, a common acronym for operations is “CRUD” – create, read, update, and

delete. In a permissioned environment, nodes may only have the authority to perform a subset of these operations. In a permissionless environment, all nodes have full authority.

A distributed ledger in the context of a blockchain can take on three different forms; public, private, or that of a consortium (Buterin, 2015). Each of these distributed ledger forms have different characteristics (De Kruijff & Weigand, 2017):

- Public ledger: a permissionless ledger, with anonymous nodes, and no requirement for trust by network members.
- Private ledger: a permissioned ledger, operated by a single entity, with a high need for trust by network members.
- Consortium / Hybrid ledger: a permissioned ledger, operated by a pre-selected set of nodes, with a degree of trust needed by network members.

2.2.3 Application of cryptography

Like distributed ledger technology, cryptography is a domain which is fundamental to blockchain technology. Cryptography enables a blockchain's integrity, its transactions to be authentic and private, and provides pseudonymous identity to its network members.

The most used cryptographic algorithm by blockchains is the Secure Hash Algorithm 3 (SHA3), developed by the NIST. By their definition, "a hash function is a function on binary data for which the length of the output is fixed." (Dworkin, 2015, p. 1). The SHA3-256 algorithm can take any form of digital input – a text file, picture, document, video – and output it to a 256-bit string. What makes this algorithm powerful, is that it is relatively easy to go from input to output, but extremely difficult to reproduce the input from the output – referred to as collision resistance (Rogaway & Shrimpton, 2004). Two only slightly different inputs provide greatly different hash outputs.

For example, producing a SHA3-256 hash of the sentence "Hello World" produces the following string:

e167f68d6563d75bb25f3aa49c29ef612d41352dc00606de7cbd630bb2665f51

In comparison, the SHA3-256 hash of the sentence "Hello, World" – which only appends a comma – produces this string:

844af7bf6a5d414359dcd8845cb52d515397410e1668e00c8469ea8728c4ffe8

In a blockchain setting, both transactions and block headers (which will be discussed later) are represented by their hash strings. The set of content which makes up the transaction or block header serves as input for the hash function, to ensure the integrity of said content.

2.2.4 Blockchain-based Public-Key Infrastructure

Identity on a blockchain is managed via a Public-Key Infrastructure (PKI). Using cryptographic techniques, a PKI binds identities with public keys. Upon creation of the public key, the user also gains a 'secret' private key. In a PKI, users use their public key to share with other subjects, and their private keys to sign transactions. PKI's might be managed centrally, using a database owned by one or several Certificate Authorities (CA), or decentralized, using DLT. Inherent to the decentralized nature of blockchain technology, a blockchain-based PKI is managed on a peer-to-peer basis. The ledger now acts as a decentralized and tamper-proof CA.

The main operations for signing and verifying messages in a PKI-environment, are part of the Digital Signature Algorithm (DSA). The DSA has three basic operations (Fromknecht & Velicanu, 2014):

1. Generate a private and public key pair
2. Produce a digital signature on a message using the private key
3. Determine whether the signature is a valid signature on the message under the private key corresponding to the public key

We illustrate the concept of blockchain-based PKI with an example from the Linux Foundation (Ryabitsev, 2014). Anyone in a blockchain network can transact with a subject, provided he or she knows the subject's public key. When the transaction is initiated, it is encrypted with the public key of the recipient. The sender now signs this transaction with their private key. The contents of this transaction can now only be decrypted with the private key of the recipient, who is able to verify that indeed the transaction comes from the sender, by verifying the digital signature.

DSA is not the only signature algorithm compatible with a blockchain environment. Other algorithms such as the Rivest, Shamir and Adleman (RSA) algorithm (1978), and the Elliptic Curve Digital Signature Algorithm (ECDSA) (Johnson, Menezes, & Vanstone, 2001), might be used. Where DSA only provides means to sign and verify messages, RSA and ECDSA also enables users to encrypt messages. Compared to RSA, ECDSA produces smaller keys and signatures, but it takes longer to verify ECDSA signatures.

Currently, in a blockchain environment, subjects manage their key pairs via so-called 'wallet' software. Wallets are applications which interact with a blockchain. Upon initial loading, wallets prompt the user to generate a key pair, which is then associated with their account. A limitation of wallet software is that it is not user-friendly enough for the digitally less experienced users.

2.2.5 Technology layers of a blockchain

A blockchain has multiple technology layers. Looking at the (basic) Bitcoin blockchain, three layers can be distinguished (Swan, 2015):

- The bottom layer; a distributed ledger which records and verifies transactions
- The middle layer; the bitcoin protocol software to conduct transactions
- The top layer; the cryptocurrency layer, in this case Bitcoin (BTC)

2.2.5.1 The distributed ledger layer

Using one of the three forms of DLT as described in §2.2.2, the DLT layer records transactions in what are called 'blocks'. Each block consists of two main data fields; the block header and the contents (Grewal-Carr & Marshall, 2016). The hash value of a block header is made by hashing multiple data fields, most importantly; the hash of the previous block, the root hash of all contents of the current block, the timestamp, and a nonce. The data structure of a block is visually represented in figure 6 on the next page.

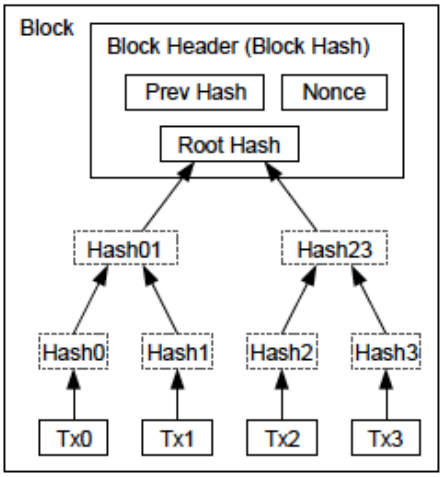

Figure 6: Data structure of a bitcoin block, as described by Nakamoto (2008, p. 4)

An important note is that each block references the previous block and contains a timestamp. These two data fields ensure that a blockchain is – and remains – chronological, as a new block will not be added if the reference to the previous block is incorrect, or the timestamp of the current block is earlier than that of the previous block.

The contents of a block are made up by transactions – commonly referred to as tx's. A transaction contains the public key of the receiving party, the hash of the previous transaction, and a signature by the sending party, verified by the public key and signed with his/her private key. A visual representation is displayed in figure 7 below.

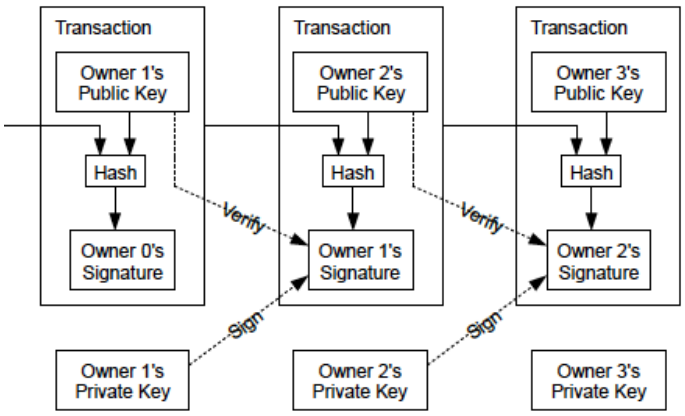

Figure 7: transactions on a blockchain, by Nakamoto (2008, p. 2)

2.2.5.2 The protocol layer

We illustrate how a transaction is recorded with the Bitcoin protocol by providing a simple example; Bob wants to transact with Alice. To initiate the transaction, Bob needs the public key of Alice, and his own private key. Bob can now use specific software to transact with Alice, by using Alice's public key as the receiving address, and signing the transaction with his private key. This transaction is then broadcast to the global network of peers. Once the peers verify that Bob can make the transaction (e.g. does he have enough currency), the transaction is grouped into a block together with other transactions. The block is now ready to be 'mined', to prove its validity and reach consensus on the new state of the blockchain. The process of validating blocks and how consensus might be reached – the consensus mechanism – will be explained in more detail in §2.2.5. The consensus mechanism is arguably at the core of a blockchain protocol.

2.2.5.3 The cryptocurrency layer

The top layer consists of a cryptocurrency, of which there are more than 1200 active[1] as of November 2017. A cryptocurrency is generally created by one of two approaches; starting with zero coins in circulation, or having coins "pre-minted".

Bitcoin was initiated by having no coins in circulation. New coins were generated – or minted – by validating blocks. The person validating the block received 25 bitcoins as a reward for his efforts. The amount of, and rate at which, new cryptocurrency which can be minted is defined in the protocol. In the case of Bitcoin, there is a capped maximum supply of 21 million Bitcoin. Ether, on the other hand, was pre-minted by the Ethereum developers. When a cryptocurrency is pre-minted, a part of the total coins in circulation is allocated to addresses before the actual creation of the blockchain. A common practice is when peers pay the developers of coin A a certain amount of cryptocurrency B, and in exchange are allocated an amount of coin A[2].

Multiple cryptocurrency layers can make use of the same protocol and distributed ledger layer (Swan, 2015). For example, an imaginary "Alphacoin" could use the same protocol to transact with, and record them on the same blockchain as that of Bitcoin. On the other hand, a

[1] See www.coinmarketcap.com for an actual overview of cryptocurrencies
[2] This model of cryptocurrency allocation is called an Initial Coin Offering (ICO)

combination of cryptocurrency and protocol layers might record their blocks on the same distributed ledger as another combination of cryptocurrency and protocol.

2.2.6 Consensus mechanisms

For every node to have and agree on the same state of a blockchain, consensus must be reached on the state of said blockchain. The algorithm which defines how each node processes this state transition is called a consensus mechanism. According to Baliga (2017), a consensus mechanism has three key properties:

1. Security: do all nodes produce valid outputs, ensuring a consistent state of the blockchain.
2. Liveness: do all non-faulty nodes participating in consensus eventually produce an output.
3. Fault tolerance: can the protocol recover from failure of a node participating in consensus.

Many consensus mechanisms exist. Generally, there are two types. One is based on proofs, one on validation. Proof mechanisms are suited for public permissionless blockchains. In contrast, validation mechanisms are mostly used in permissioned settings. For the scope of this study, we will now describe two of the most widely used 'proof' consensus mechanisms for blockchains, Proof-of-Work (POW) and Proof-of-Stake (POS).

2.2.6.1 Proof-of-Work

As its name implies, the POW algorithm is based on nodes using CPU/GPU resources to prove the work they have done. In the creation of a block, all data fields but the nonce data field are static. To prove the validity of the block, the nonce must be incrementally updated, calculating the hash of the block for each increment. A block is valid once the hash meets a difficulty condition as defined in the protocol, e.g. the hash must start with four zeroes. Due to the fact that blocks reference their previous blocks, a block is practically immutable once it is proved valid. An adversary wanting to mutate the contents of a block x, would have to re-do all the proof-of-work of blocks x to the most recent block. This would require a significant amount of resources, and has proven not to be economically feasible.

In return for their effort, miners are rewarded a set amount of cryptocurrency upon validating a block. A major drawback of POW is that the electricity required by the CPU/GPU is purely wasted. At the time of writing this report, the bitcoin blockchain uses 23% of the total energy consumption of the Netherlands[3].

2.2.6.2 Proof-of-Stake

Contrary to the POW algorithm, a POS algorithm lets users act as validators by staking an amount of cryptocurrency to determine validity of blocks. Arguably the most advanced version of Proof-of-Stake algorithm, is the Casper protocol of Ethereum (Buterin & Griffith, 2017). There are generally two types of POS algorithms (Gubik & Buterin, 2017): chain-based POS and byzantine fault tolerant (BFT) style POS. In a chain-based POS algorithm, validators lock up their funds in an address, and the algorithm periodically (e.g. every ten seconds) determines which validator earns the right to create a new block. In a BFT-style POS algorithm, *"validators are randomly assigned the right to propose blocks, but agreeing on which block is canonical is done through a multi-round process where every validator sends a vote for some specific block during each round, and at the end of the process all (honest and online) validators permanently agree on whether or not any given block is part of the chain"*.

A major advantage of POS over POW is the fact that mining with POS is virtual instead of physical. Instead of having hardware consuming electricity to prove their work, POS lets users stake their virtual currency. Since 'stakers' use significantly less electricity with POS, they also need less incentive to participate in the form of block rewards, which reduces the inflation of the network.

2.2.7 Smart Contracts

Smart contracts add an extra layer to the blockchain technology stack. Instead of only being able to perform 'basic' transactions – sending cryptocurrency from A to B – smart contracts provide the network with the ability to program transactions. The idea of smart contracts was first proposed in 1997 to formalize electronic relationships (Szabo, 1997), but never had a solid use-case until blockchain technology was introduced. Smart contracts are essentially a piece of software code, of which the consensus mechanism on a blockchain ensures the correct execution.

[3] See https://digiconomist.net/bitcoin-energy-consumption

The contract has its own public key address, and fully resides on the blockchain. Contrary to a 'normal' account, a smart contract is enforced not by any form of private key, but only by its immutable code.

The most popular smart contract platform is the Ethereum network. Once a contract is deployed on Ethereum, its address holds an amount of Ether, has a private storage, and is associated with its code (Luu, Chu, Olickel, Saxena, & Hobor, n.d.). Users who want to interact with a contract, usually send two types of information. First, 'gas' is provided to the contract address. On Ethereum, gas specifies the amount of Ether that the user is willing to use, so that the network executes the contract. Second, a data-field is provided in the transaction to the contract, which serves as an input for the contract's code.

An urgent security issue with smart contracts, is that there is no widespread method of formal verification (Atzei, Bartoletti, & Cimoli, 2017). Due to the immutability of smart contracts, they must be resistant to all forms of attacks before deploying them. In practice, there have been multiple cases of insecure smart contracts – deployed without formal verification – which have been exploited. This had led to significant financial losses, such as is in the Parity multi-signature contract[4] and the DAO exploit[5].

2.2.8 On- and off-chain data handling

Deciding which data to store in blocks, and which should reside off the blockchain, is highly important. Although DLT provides a high level of security and privacy, it is also relatively costly and slow. It is however possible for transactions to point and have access to data stored on off-chain databases. For example, a contract may point to the location of a file with a certain hash value.

By definition, a blockchain is unable to access external data sources. However, smart contracts may very well need external data feeds. Oracles – an agent that finds and verifies real world occurrences and submits this information to a blockchain[6] – are off-chain solutions which can feed data to smart contracts. The benefit of oracles is that their data is external, thus handled more efficient. However, no consensus mechanism ensures the integrity of oracle data.

---

[4] USD 30 million stolen: https://blog.ethcore.io/the-multi-sig-hack-a-postmortem/
[5] USD 150 million stolen: http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/
[6] https://blockchainhub.net/blockchain-oracles/

Another development to handle data in a blockchain environment, are decentralized storage solutions. Two advanced projects in this field are IPFS and Swarm. Swarm is part of the Ethereum technology stack, while IPFS is an independent solution. Inspired by BitTorrent, both Swarm and IPFS are distributed storage solutions which make use of content-addressable storage (CAS). With CAS, smart contracts can access data based on their hashname, instead of a traditional IP-location.

2.3 Regulatory environment

2.3.1 General Data Protection Regulation (GDPR)

Starting in May 2018, the GDPR will become active. This regulation aims to protect the privacy and security rights of natural persons. According to Siriwardena (2017a), the main implications for identity management are as follows:

1. User onboarding should be an automated process
2. Only minimal data should be captured and processed
3. User consent must explicitly be recorded
4. Users should be able to gain access to a personal portal, to change any current practices regarding their data
5. Where possible, data should be anonymous, or else pseudonymous
6. Subject data should be processed transparently
7. Protect the integrity and privacy of subject data by ensuring security

The GPDR drastically changes the way personal data is controlled and processed. Personal data is defined in Article 4 as 'any information relating to an identified or identifiable natural person' (European Parliament, 2016, p. 33). Personal data is not limited to PII, but also concerns data such as location and IP-addresses. Before the GDPR, only data controllers – the entities who determine the purpose and means for processing data – were held accountable for their actions. When the GDPR comes in effect, data processors – entities processing data on behalf of a data controller – have increased accountability and responsibility (Salmon, 2016). For example, data processors may only process data on behalf of a controller if a written contract is in place, according to instructions of the controller. Data processors may not engage with sub-processors without written authorization of the controller. Processors must take additional security measures and report any data breach to the respective controllers.

2.3.2 Revised Payment Services Directive (PSD2)

In November 2015, the Revised Payment Services Directive (PSD2) was accepted by the European Union (European Parliament, 2015a). The directive aims to harmonize and standardize rules for payment services across the EU, for both financial and non-financial institutions. The regulation will become effective from January 2018. The directive has significant

implications for identity management, as the key rules concern the following (European Parliament, 2017):

- Strict security requirements for electronic payments and the protection of consumers' financial data, guaranteeing safe authentication and reducing the risk of fraud
- Transparency of conditions and information requirements for payment services
- Rights and obligations of users and providers of payment services

PSD2 differentiates between Payment Service Providers (PSP's), Account Information Service Providers (AISP's), and Payment Initiation Service Providers (PISP's). PSP's can be either banks or non-banks who provide their customers with payment services, such as holding a bank account with funds. Under PSD2, service providers can get insight into multiple bank accounts of a customer. AISP's can then, with explicit consent of their customers, aggregate financial account information into one portal. Lastly, PISP's can – again with explicit consent of their customers – initiate transactions on behalf of their customers, which can streamline payment processes for consumers.

To facilitate the payment services directive, multiple important process must be implemented. The European Banking Authority (EBA) is responsible for defining technical standards and implementation guidelines, to facilitate integration. As of 2017, they have drafted the Regulatory Technical Standards (RTS) on Strong Customer Authentication (SCA) and secure communication (European Banking Authority, 2017). For a PISP or AISP to gain access to a consumers PSP, explicit consent of the consumer is required.

In regard with consent requirements as described by PSD2, synergy is found with the GDPR, where the following requirements for a valid consent are described (Privacy Valley & Van Hasselt, 2017):

- Consent must be actively given, the subject has to perform a deliberate action
- Consent must be freely given and revocable
- Consent must be granular, giving the subject options
- Consent must be specific and informed
- Data must only be used for the original purpose to which the subject consented

- A proof of consent must be demonstrable
- The consent of a parent is needed when the subject is a child
- A subject must be able to withdraw consent at any given time

2.3.3 Electronic Identification, Authentication and Trust Services (eIDAS)

In July 2014, the member states of the EU have agreed to standardize and harmonize signatures and transactions in the EU internal market. eIDAS will come into full effect by September 2018. The regulation is aimed at three key areas (European Parliament, 2015b, p. 83):

- "the conditions under which Member States recognize electronic identification means of natural and legal persons falling under a notified electronic identification scheme of another Member State"
- "rules for trust services, in particular for electronic transactions"
- "a legal framework for electronic signatures, electronic seals, electronic time stamps, electronic documents, electronic registered delivery services and certificate services for website authentication"

Over the last decade, multiple countries have developed their own governmental e-ID systems, such as the German nPA, the Austrian Citizen Card, the Belgian eID and the Dutch DigiD (Cuijpers & Schroers, 2015). However, these have all been separately developed, thus lack interoperability. eIDAS aims to create common ground for eID schemes, therefore an important aspect of eIDAS are digital signatures, and the authentication and encryption that accompany them. eIDAS defines three Authentication Assurance Levels (AAL) – low, substantial, and high – for enrolment of identities, electronic identification means management, authentication, and management and organisation (Commission, 2015).

Electronic signatures are divided into three categories; standard, advanced, and qualified. A standard electronic signature is logically associated with other data in electronic form and which is used by the signatory to sign (Farah, 2017). An advanced signature has the features of a standard signature, but also requires that is backed by a digital signature in sole control of the signer (Davies, 2016). Last, a qualified electronic signature is the same as an advanced signature, but requires the signer to be on a centrally managed list of qualified providers (Farah, 2017).

2.4 Propositions

We draw the following conclusions from the literature review. First, (digital) identity management is becoming an ever-increasingly important facet of our society. However, the domain of identity management faces multiple issues in the areas of security, privacy, usability, and implementability. These issues are pushing for a self-sovereign identity management model, instead of a centralized or federated model.

Second, blockchain technology can provide the means of a distributed and decentralized identity management system, which is an important driver for self-sovereignty.

Third, the European Union has made significant pushes to address the issues of identity management, by creating a new regulatory environment. As of 2018, three major regulations and directives will become effective; the General Data Protection Regulation, the Revised Payment Services Directive, and the electronic Identification, Authentication and Trust Services Regulation.

We propose to further develop the construct of self-sovereign identity, by operationalizing Allen's ten principles – stated in §2.1.5.1 – for a regulatory compliant self-sovereign identity system. This operationalization will be done in the form of system requirements, which will then need practical validation. This will provide a solid foundation for the concept of Self-Sovereign Identity, which can then be used to evaluate the added value of blockchain technology.

3 METHODOLOGY

3.1 Requirements engineering

The building and evaluating of this design – the IS research in Hevner's model – will be an iterative process. The high-level phases of a requirements engineering process are as follows (Parviainen, Tihinen, Lormans, & Solingen, 2005):

1. System requirements development
2. Requirements allocation and flow-down
3. Software requirements development
4. Continuous activities

This study will – due to time and scope constraints – focus on the first part of the requirements engineering process, system requirements development. In the context of this study, system requirements are not to be mistaken with hardware requirements. Instead, system requirements consist out of a set of functional and non-functional requirements, which together describe the desired functioning of the system, according to stakeholder objectives. System requirements development can be divided into three separate activities: (1) requirements gathering, (2) analysis and documentation, and (3) validation and verification. Each activity is described in more detail in table 1 below:

Table 1: system requirements engineering activities

System requirements development activity	Description
Requirements gathering	Identify stakeholders and elicit raw requirements.
Analysis and documentation	Analyse dependencies, conflicts, overlaps, omissions, and inconsistencies.
Validation and verification	Review with stakeholders.

3.2 Design activities

In accordance with the design activities of our research framework, two iterations of build and evaluate will be undertaken. The steps taken and methodologies employed in the iterations are as follows:

1. **Build**: deduct high-level requirements by introspection, using existing literature, self-sovereign identity principles and regulatory documents.
2. **Evaluate**: conduct semi-structured interviews with stakeholders to elicit new requirements and validate requirements from step 1. Notes from the conducted interviews can be found in appendices D up to and including H.
3. **Build**: analyse and improve the complete set of requirements, based on the results from the step 2.
4. **Evaluate**: send results of step 3 to the stakeholders, so that they may validate and verify the identified requirements.

The first build iteration starts with an introspection. Introspection is often the first step in a requirements engineering process (Sharmila & Umarani, 2011). To retain a holistic view of the Self-Sovereign Identity concept, and develop the most exhaustive system requirements as possible, the results of the introspection should be validated by a diverse set of stakeholders. At the very least, input from Government, Financial Services, and R&D/Academia is needed.

In the first evaluation iteration, semi-structured qualitative interviews will be conducted. In the field of information systems research, qualitative interviews are a powerful tool to gather data and elicit new insights (Myers & Newman, 2007). Since the nature of this research is exploratory, semi-structured interviews fit well to elicit new viewpoints from open questions, while at the same time providing consistency from pre-defined questions (Yousuf & Asger, 2015).

The second build iteration takes the results from the first evaluation iteration as input, to analyse and improve the set of requirements. Dependencies, inconsistencies, and double requirements will be analysed, so that a set of requirements can be built which can be sent to the stakeholders for feedback.

In the second evaluation iteration, the non-validated results of the second build iteration will be sent to the stakeholders for review. They have been asked to review the identified requirements based on correctness, accuracy and completeness. To support the above activities, the list of professionals in table 2 have been consulted in the two evaluation iterations. A

disadvantage of conducting semi-structured interviews for requirements engineering, is that difficulties have been encountered in finding appropriate interviewees, and finding the time to set meetings with them, and have them validate the results. Moreover, the findings may not be generalizable.

Table 2: stakeholders to be interviewed

Name	Organisation	Industry	Input role
A. de Kok	RVIG	Government	Elicitation and validation
M. Everts	TNO	Private Research	Elicitation and validation
D. Baars	Rabobank	Financial Services	Elicitation and validation
K. Roodnat	VitalHealth	IT solutions - Healthcare	Elicitation
J. van Beek	Iquality	IT solutions - Generic	Elicitation and validation
J. de Kruijff	Tilburg University	Academia	Validation

3.3 Evaluating blockchain technology for self-sovereign identity

After the artefact has been built and evaluated, blockchain technology will be evaluated to determine to what extent using the technology satisfies each requirement. As described in previous work (Lapouchnian, 2005; Letier & Lamsweerde, 2004), in some situations it is desirable to introduce additional qualitative levels of requirement or goal satisfaction. Geared towards non-functional requirements, the NFR framework (Chung, Nixon, Yu, & Mylopoulos, 2000) has developed a four-step scale; denied, satisfied, partially denied, and partially satisfied. For the scope of our study, which includes constraints, functional and non-functional requirements, we introduce a three-step scale:

1. Does not satisfy the requirement
2. Partially satisfies the requirement
3. Fully satisfies the requirement

Normally, satisfaction of a requirement is binary, either the requirement is met, or it is not. However, for the purpose of this research and the scope of blockchain as a technology, the three-step scale has been chosen. The requirements and validation criterions might not be fully satisfied by just using blockchain technology. Therefore, satisfaction level 2 provides the opportunity to nuance the evaluation. Other technology or processes might be needed to fully satisfy a given requirement or constraint. It is important to note that each requirement will have the same weighting. Arguably, this is a limitation of the study. Determining weighting of each requirement would improve the analysis, however, due to time constraints, this will not be done. When the satisfaction levels for all requirements have been identified, each constraint and use-case will have a total satisfaction score. This is simply computed by summing the satisfaction levels for each validation criterion within a given constraint or use-case, and dividing that by the maximum possible satisfaction level.

After the self-sovereign identity construct has been operationalized, and blockchain technology evaluated against the system requirements and constraints, a comparative analysis will be performed to illustrate how two specific self-sovereign identity system implementations work, and differ in their approaches. This analysis will provide the reader with a complete view of the subject matter, by illustrating a practical example of self-sovereign identity usage. Two system implementations will be compared; Sovrin and uPort.

4 OPERATIONALIZING SELF-SOVEREIGN IDENTITY

Self-sovereign identity management (SSIM) has long been imagined as a primary use case for blockchain technology. One of the first self-sovereign identity initiatives was the Open Mustard Seed framework. It proposed an extra layer on top of the internet, enabling users to become their own central authority. Over the last two years, many more parties worked on a self-sovereign identity management framework. What follows is a description of four different SSI frameworks, an analysis of their general architecture, and the requirements to which a self-sovereign identity system must adhere.

4.1 Ongoing development of Self-Sovereign Identity frameworks

Multiple SSIM frameworks are in development by industry and government professionals. We will discuss the frameworks developed by the World Economic Forum, the ID2020 joint program, a Dutch public-private partnership called Techruption, and the 3DID framework of Consult Hyperion.

The World Economic Forum (WEF) published a report on "the blueprint for digital identity" (2016). The WEF argues that financial institutions should be put at the forefront of new identity management solutions, due to their existing role as both an identity provider, control party, and relying party. Further, the report states that financial institutions could take three different approaches to introducing SSIM solutions; (1) a single-institution approach, (2) a consortium approach, and (3) a utility approach. They advise to use centralized or distributed identity archetypes, depending on the context. For individuals, a distributed model is advised, while for legal entities and assets, a centralized model is advised.

A joint program, pioneered by Microsoft and Accenture, called ID2020 aims to unify governments, NGO's and private institutions in creating a global SSIM solution. ID2020 aims to comply with the ten principles outlined by C. Allen. In contrast to the framework of the WEF, ID2020 envisions an ecosystem in which all stakeholders as defined by Swan operate together. A draft working paper (Sporney & Longley, 2016) outlines the SSIM architecture of ID2020.

In the Netherlands, seven participants of the Techruption – initiated by the Dutch Blockchain Coalition – project are developing a Self-Sovereign Identity framework, inspired by

the work of Baars (2016), based on blockchain technology (Joosten, 2017). They classify digital identity as the cornerstone for further development of blockchain applications. The global design of the framework is aimed at decentralizing the creation, validation and revocation of digital identities. The framework makes use of attestations – signed statements about an identifier – to enable subjects to freely exchange verifiable claims.

Lastly, Consult Hyperion – a consulting firm based in the UK – developed a three-layered DID model. Birch (2016) states that we should "think about the digital identity as a private-public key pair and think about the virtual identities as public key certificates that take the public key from the digital identity and link it to attributes to form credentials". In the identification domain, a subject is bound to a digital identity. In the authentication domain, the digital identity is bound to one or more subjects who are entitled to use the digital identity. Lastly, in the authorization domain, a digital identity is linked to one or more virtual identities, which form credentials from attributes associated with the public key of the digital identity.

Although the above frameworks all differ in their approach, the basic framework architecture are alike. The similarities between them are as follows:

- Subjects have an independent existence which rely on decentralized identifiers
- Identity providers issue verified claims coupled to a subject's identifier
- Subjects are able to store self-asserted or verified claims on a personal repository
- Subjects can provide informed consent in regard to with whom they exchange certain parts of a claim
- Relying parties are able to verify attestations of a claim
- Consent and attestations can be revoked

A visual representation of the basic architecture is displayed in figure 8. Note that this architecture is agnostic of the technology stack to be used by the 'public ledger'. Moreover, the proprietary databases of each stakeholder are out of scope for this research.

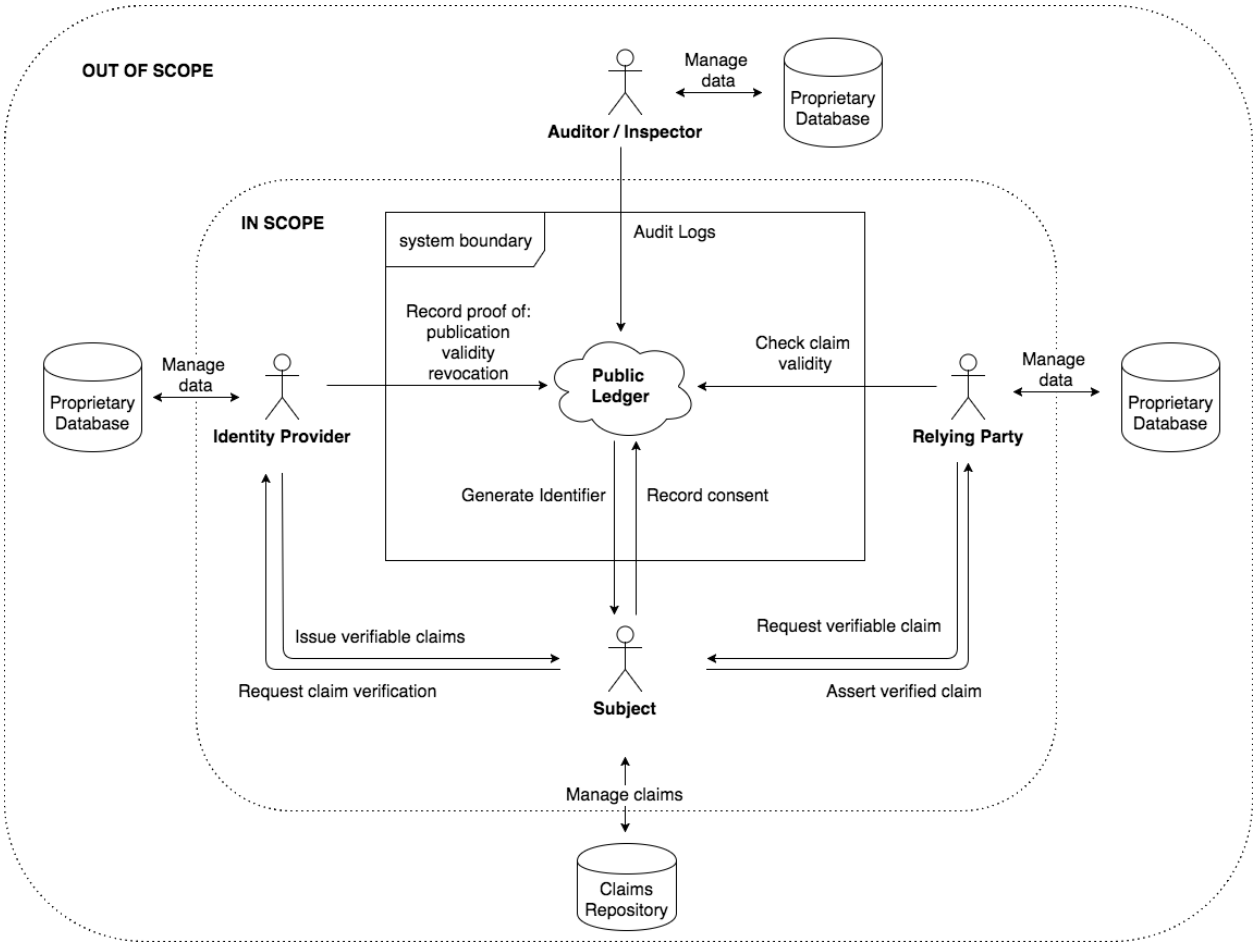

Figure 8: self-sovereign identity architecture

4.2 System requirements for Self-Sovereign Identity

To gain a deeper understanding of Self-Sovereign Identity concept, a set of system requirements have been developed. The stakeholders are the same as in figure 8; identity providers, relying parties, subjects, and possibly auditors. The regulatory environment as discussed in §2.3 act as constraints for the system requirements. The requirements will be split up per use-case. For each use-case, relevant stakeholders, SSI principles, event triggers, process steps, and use-case results will be described. Following, we describe functional and non-functional requirements for each use-case, each with rationale and validation.

4.2.1 Regulatory constraints

As described in §2.3, a self-sovereign identity system must be compliant with the GDPR and eIDAS, and be compatible with PSD2. Therefore, these regulations act as constraints from which we derive global validation criteria, applicable to each use-case. Each constraint with accompanying validation criteria are displayed in tables 3, 4 and 5.

Table 3: GDPR compliance validation criteria

Constraint: The system MUST be compliant with the General Data Protection Regulation (GDPR)	
Validation Criterion 1	Is any personal data of a subject only accessible by other entities with informed consent of the subject?
Validation Criterion 2	Can any records of personal data owned by other entities than the subject be erased upon the request of the subject, given the conditions laid out in GPDR Art. 17 are met?
Validation Criterion 3	Can personal data be ported to other systems in an automated manner?
Validation Criterion 4	Is personal data securely and privately managed by design, through the use of the system's technology?
Validation Criterion 5	Does the system provide means for subject to control consent to who they share personal data with? Can consent be easily provided and revoked?

Table 4: PSD2 compliance validation criteria

Constraint: The system MUST be compatible with the Revised Payment Services Directive (PSD2)	
Validation Criterion 1	Does the system provide the means to provide or facilitate Strong Customer Authentication?
Validation Criterion 2	Does the system facilitate the means to only allow PISP's and AISP's access to a subject's PSP with informed and explicit consent of the subject, which at any time can be revoked?

Table 5: eIDAS compliance validation criteria

Constraint: The system MUST be compliant with the electronic Identification, Authentication and Trust Services regulation (eIDAS)	
Validation Criterion 1	Can public institutions from European Member States verify a subject's data, originate from another Member State, on the system?
Validation Criterion 2	Does the system provide the technological means to sign data using advanced or qualified electronic signatures?

4.2.2 System requirements per use-case

In a self-sovereign identity ecosystem, multiple use-cases have been identified for which requirements must be developed. These use-cases are as follows:

UC1. Establish a digital identity
UC2. Issue verified claims
UC3. Assert a verified claim
UC4. Revoke a claim
UC5. Entity authentication
UC6. Provide access to personal data to another entity

For each use-case, a general description will be provided. This includes the relevant stakeholders as described in §2.1.4, relevant SSI principles as described in §2.1.5.1, what event triggers the use-case, the process steps, and the result of the use-case. Subsequent to the use-case

description, the functional and non-functional requirements are provided. Each requirement has a description, rationale, and validation criterion.

4.2.2.1 Use-case 1: Establish a digital identity

Table 6: use-case description - establish a digital identity

Use-Case	UC1: subject establishes a digital identity
Stakeholders	Subject
Relevant SSI principles	Existence, Control, Persistence
Event triggers	The subject wishes to establish a digital identifier, possibly related to a specific entity
Process steps	1. Generate identifier 2. (optional) Associate the identifier with another entity
Result	The subject has control over an identifier, possibly associated with another entity

Table 7: use-case 1 - functional requirements

UC1 - Functional Requirement 1: The system SHALL let any natural person have the sole right and ability to create a digital representation of themselves, through the generation of a digital identifier	
Rationale	To be independent from any entity, subjects must have the capability to independently create and control identifiers.
Validation	Does the subject need no other entity to generate a key pair?
UC1 - Functional Requirement 2: The system SHALL enable a subject to create as many separate digital representations of themselves as they see fit.	
Rationale	Subjects must be able to create as many identifiers as they see fit. It is undesirable to re-use identifiers across multiple relationships, as this would lead to correlativity.
Validation	Can the subject generate multiple identifiers without technological limitations?

UC1 - Functional Requirement 3: The system SHALL let the subject choose which data to associate with an identifier.	
Rationale	Subjects must be free to choose which data they wish to associate with a specific identifier. Other entities might request a certain set of data, but the subject must be free to make this decision.
Validation	Does the subject have control over which data is associated with their identifiers?

Table 8: use-case 1 - non-functional requirements

UC1 - Non-Functional Requirement 1: The system SHALL NOT let the usage of a subject's identity be irrevocably controlled by a single or group of entities other than the subject. Only the subject MUST be able to control when to use data associated with their identifiers.	
Rationale	To be independent from any entity, subjects must have control over the private keys to exercise control over their identifiers
Validation	Has the subject sole control over the private key?
UC1 - Non-Functional Requirement 2: The system SHALL operate on a decentralized and non-proprietary information system.	
Rationale	For subjects to be independent from any entity and have control over their identifiers, the information system must be of decentralized nature.
Validation	Are identifiers generated on a network of open-source nature, not owned by any one entity?
UC1 - Non-Functional Requirement 3: The system SHALL NOT directly link a subject's identifier to the controlling identifier, which has the private key	
Rationale	If a subject loses control of their private key, for example in case of device theft, it is not desirable to have to change all of a subject's identifiers. The control logic may be separated from the operation logic.
Validation	Upon loss of a private key, can the corresponding identifiers stay the same?

4.2.2.2 Use-case 2: issue a verifiable claim

Table 9: use-case description - issue verifiable claims

Use-Case	UC2: identity provider issues a verifiable claim
Stakeholders	Subject, Identity Provider
Relevant SSI principles	Control, Access, Portability, Interoperability
Event triggers	The subject requests credentials from an identity provider
Process steps	1. Subject requests verifiable claim 2. Subject identifies and authenticates with the identity provider 3. Identity provider checks validity of statements, according to a required assurance level 4. Identity provider issues the verifiable claim, and binds it to the subject's identifier by publishing a proof of existence on a publicly accessible ledger
Result	The subject has received a verifiable claim in his repository, bound to his identifier

Table 10: use-case 2 - functional requirements

UC2 - Functional Requirement 1: The system SHALL let any entity act as an identity provider by issuing a verifiable claim to a subject	
Rationale	To enable peer-to-peer attestations, any entity must be able to attest to certain statements about a subject
Validation	Can any entity attest to a statement and generate a claim?
UC2 - Functional Requirement 2: The system SHALL let the subject self-attest to claims about its identity attributes	
Rationale	Subjects must be able to act as their own identity provider. They must be able to self-attest to statements made about themselves.
Validation	Can subjects self-attest to statements about themselves?

UC2 – Functional Requirement 3: Upon issuing a claim or attesting to an existent claim, proof of this event SHALL be published on a publicly accessible ledger

Rationale	To enable Relying Parties to inspect the validity of a claim, they must be able to check a provided claim against a proof of its issuance.
Validation	Does the proof of issuance or attestation of a claim require publication on a publicly accessible ledger?
UC2 – Functional Requirement 4: Upon issuing or attesting a verifiable claim, the system SHALL let the issuer or attester choose whether this is done publicly or pseudonymously	
Rationale	Identity providers must be able to choose whether they want their identity to be known to the public, or use a pseudonym.
Validation	Can identity providers choose whether they re-use a (well-known) public address or a one-time pseudonymous address?
UC2 – Functional Requirement 5: The system SHALL ensure claims can be limited to what is necessary in relation to the purposes for which they are processed, and thus consist of granular statements	
Rationale	Articles 5 and 6 of the GDPR advance the principle of data minimization. Personal data must be relevant to the purpose for which they are collected and/or further processed. For example, a subject must be able to receive a claim that states he or she is older than 18 years old, instead of his or her birthdate.
Validation	Upon creating a claim, can the statements within it be minimized for its purpose, and are they individually signed?

Table 11: use-case 2 - non-functional requirements

UC2 – Non-Functional Requirement 1: Entities SHOULD be able to bind their identifiers to human-readable names, which MAY be done via a key-based naming service	
Rationale	UC2-FR4 requires that entities shall have the possibility to issue or attest to claims publicly or pseudonymously. To publicly reveal the identity of the identity provider, their public key must be associated with a human-readable name
Validation	Can entities associate an identifier with a human-readable name?
UC2 – Non-Functional Requirement 2: The system SHALL treat claims and/or attestations as objects linked to an identifier	
Rationale	To prevent multiple claims to be correlated with each other by entities other than the subject, they must be separate data objects, linked to an identifier
Validation	Are claims and/or attestation separate objects, linked to an identifier?

UC2 – Non-Functional Requirement 3: Claims MUST be issued in machine readable format, with standardized semantics	
Rationale	To ensure maximum portability and interoperability, claims should be issued in a standardized format. One of the most common formats for exchange of attributes is JSON-LD. In the data object, references can be made to semantic schemas.
Validation	Are claims issued via a standardized, machine-readable format, with clearly defined semantics?

4.2.2.3 Use-case 3: assert a verifiable claim

Table 12: use-case description - asserts verifiable claims

Use-Case	UC3: subject asserts a verified claim
Stakeholders	Subject, Relying Party
Relevant SSI principles	Control, Access, Minimization, Consent, Protection, Portability
Event triggers	Subject interacts with a relying party
Process steps	1. The relying party requests proof of a statement 2. The subject (selectively) presents his verified claim 3. The relying party inspects the claim and verifies its validity
Result	The subject proved validity of a set of statements to a relying party

Table 13: use-case 3 - functional requirements

UC3 – Functional Requirement 1: Relying parties SHALL be able to request a verifiable proof of statements about a subject, and MAY request that it be attested for by a specific entity or group of entities	
Rationale	Relying Parties often need proof of certain statements about a subject, before they engage with the subject. Relying Parties must be free to choose what kind of proof they can request.
Validation	Are relying parties able to ask for specific verifiable claims, possibly requiring specific statement types?
UC3 – Functional Requirement 2: The subject SHALL be able to selectively disclose parts of a claim to his or her choosing, to minimally satisfy the requirement of the relying party	
Rationale	Often, claims contain more information than necessary in a given scenario. Subjects must only disclose that information which is relevant to the relying party. They must be free to choose any claim or combination of statements across claim, which satisfies the relying party.
Validation	Upon asserting a claim, can the subject freely choose from which source to prove the minimal needed statements?
UC3 – Functional Requirement 3: A personal repository SHALL aggregate all claims in a subject's control, inform with which third parties they are shared, and the validity of the subject's consent to share them	

Rationale	To ensure maximum portability and control, claims should be held in control of the subject via a personal repository. To inform the subject to whom and under what conditions he or she has asserted a claim, the repository should display with which entities the claim is shared with, and the validity of the consent to share
Validation	Are claims asserted to a relying party via a personal repository of the subject, over which only the subject has control? Does the repository inform the subject with whom claims are shared, and the validity of their consent to share the claim?
UC3 – Functional Requirement 4: If a relying party wishes to share personal data from a shared claim with another entity, the subject MUST again provide informed consent. The system SHALL register a proof of consent	
Rationale	Upon sharing a claim with an entity, the subject provides consent for this action. However, driven by GDPR Art. 7, if said entity wishes to share personal data captured in the claim, informed consent of the subject is again needed.
Validation	Provided the relying parties handle according the rules, is the subject made aware to who they have given consent to use their personal data? Is explicit consent of the subject required for a third party to share personal data with other entities?

Table 14: use-case 4 - non-functional requirements

UC3 – Non-Functional Requirement 1: Data in the personal repository of subjects SHALL be encrypted, according to the latest industry standards	
Rationale	To preserve the security and privacy of subjects, and adhere to GDPR, personal data stored in a subject's personal repository should be encrypted
Validation	Is all personal data in personal claim repositories automatically encrypted, according to the highest industry standards?

4.2.2.4 Use-case 4: revocation of a verifiable claim

Table 15: use-case description - revoke a claim

Use-Case	UC4: identity provider revokes a claim
Stakeholders	Identity Provider, Subject
Relevant SSI principles	Transparency, Control
Event triggers	The identity provider wishes to revoke a verified claim
Process steps	1. The identity provider determines a claim invalid 2. The identity provider publishes a proof of revocation 3. The claim of the subject is no longer verified by the identity provider
Result	The identity provider revoked a claim

Table 16: use-case 4 - functional requirements

UC4 – Functional Requirement 1: An identity provider SHALL be able to revoke a claim, or an attestation to a claim, which they have issued	
Rationale	If an identity provider no longer deems a claim it has issued valid, they must be able to revoke it
Validation	Does the system provide a mechanism for identity providers to revoke claims or attestations?
UC4 – Functional Requirement 2: A relying party SHALL be able to verify, in an automated fashion, whether a claim has been revoked.	
Rationale	Relying parties should be able to verify whether a claim, or an attestation to a claim, presented by a subject has been revoked
Validation	Can a relying party verify whether any given claim or attestation to a claim has been revoked?

4.2.2.5 Use-case 5: entity authentication

Table 17: use-case description - entity authentication

Use-Case	UC5: entity authentication
Stakeholders	Subject, Relying Party, Identity Provider
Relevant SSI principles	Protection, Interoperability
Event triggers	An entity requests authentication of a subject
Process steps	1. The subject presents the identifier which is used for the relationship with the entity 2. The subject is asked to prove possession and control over the identifier 3. The subject inputs necessary authentication methods
Result	The subject has authenticated itself

Table 18: use-case 5 - functional requirements

UC5 – Functional Requirement 1: The system SHALL provide means to electronically authenticate entities with varying authentication assurance levels	
Rationale	When two entities engage in a transaction with each other, they must prove they have control over their corresponding identifiers. Therefore, the system must provide means to electronically authenticate the entities. This might be by proving control of the private key through one or multiple factors. For example, PSD2 mandates Strong Customer Authentication in payment transaction scenarios, by authentication through hardware modules and biometrics.
Validation	Does the system provide means for electronic authentication with varying authentication assurance levels?
UC5 – Functional Requirement 2: The system SHALL authenticate entities through proof of possession of a private key through a cryptographic protocol. The system SHALL require the entity to re-authenticate every 12 hours, or after 15 minutes of inactivity.	
Rationale	Cryptographic protocols are arguably the most secure method of electronic authentication. Not only subjects, but also relying parties and/or identity providers must authenticate themselves to prevent verifier-impersonation and verifier-compromise.

Validation	Are entities required to authenticate themselves by proving possession and control over a private key?

Table 19: use-case 5 - non-functional requirements

UC5 – Non-Functional Requirement 1: The system SHOULD NOT enable entities to store authentication records about other entities. The system SHALL authenticate entities locally, and provide cryptographic proof of authentication to the other entity.	
Rationale	If the system was to record authentication records, this may pose both security and privacy risks, even if the records are encrypted. Therefore, authentication must happen locally on the device of the subject.
Validation	Does the system store no records of authentication?

4.2.2.6 Use-case 6: authorize another entity access to (personal) data

Table 20: use-case description - authorize data access

Use-Case	UC6: subject authorizes another entity to access a selection of data
Stakeholders	Subject, Relying Party, Identity Provider
Relevant SSI principles	Consent, Control, Protection
Event triggers	An entity requests access to data of a subject
Process steps	1. An entity details to which data of the subject it requests access to 2. The subject determines if, and under what conditions, he or she wants to provide the other entity access to his or her data. 3. If the subject wishes to grant the entity access, he records a proof of consent on the public ledger 4. The entity gains access according to predetermined conditions, which are recorded in the proof of consent
Result	The entity has access to a subject's data

Table 21: use-case 6 - functional requirements

UC6 – Functional Requirement 1: The system SHALL require consent actions by the subject to be deliberately provided, and record a proof of it on a publicly accessible ledger	
Rationale	Consent must be deliberately expressed, after which proof must be recorded on a public ledger. The requesting entity can now prove that the subject has provided him or her consent to share data, providing an auditable log.
Validation	Does the system require consent before sharing data, and is proof of consent recorded on a public ledger?

UC6 – Functional Requirement 2: A subject SHALL be able to revoke his consent to share data with another entity. Revocation of consent SHALL be registered on the publicly accessible ledger.

Rationale	If a subject no longer wishes to grant another entity access to his or her data, the subject must be able to revoke the consent made in an earlier stage. The revocation should be recorded on a public ledger, so that the subject can now prove that consent was revoked, providing an auditable log.
Validation	Can a subject revoke consent to sharing data, and is a revocation recorded on a public ledger?

Table 22: use-case 6 - non-functional requirements

UC6 – Non-Functional Requirement 1: The requesting entity SHALL clearly state the purpose of collection, and the duration of needed access, for the purpose of informing the subject before providing consent	
Rationale	The subject must know what he or she will agree or disagree to. The requesting entity should clearly state which data is needed, why it is needed, and for how long access is needed
Validation	Upon request of data access, is the subject unmistakably made clear of the purpose and duration of needed access?

5 BLOCKCHAIN TECHNOLOGY FOR SELF-SOVEREIGN IDENTITY

A noteworthy talk by Vitalik Buterin (2017) – the founder of Ethereum – at a developer conference summarized, in its current development state, the primary advantages and disadvantages of blockchain technology. He argued that blockchains are good at (1) ensuring that the state of a blockchain is valid, by ensuring that the process of getting to the current state followed a set of rules, (2) informing how a ledger reached its current state, to provide transparency, and (3) assuring the integrity of data, by guaranteeing immutability. On the contrary, blockchains are less well designed to achieve high scalability, and currently do not preserve full privacy of their users.

This judgment stresses that blockchain technology should not be seen as a solution to all challenges. Instead, we must investigate in which domains it can provide added value.

We will now investigate to what extent using blockchain technology as the public ledger displayed in figure 8, meets the requirements as set out in chapter 4. To accomplish this, each requirement is validated by using the three-step scale as described in §3.3.

5.1 Satisfaction of requirements by using blockchain technology

Table 23, starting on the next page, displays the results of evaluating to what extent blockchain technology satisfies the system requirements and constraints. The first column identifies each requirement as defined in the previous chapter. The second column states how satisfaction of the requirement can be validated, and the last column states the authors opinion on to what extent blockchain satisfies the requirement. For each requirement validation, a rationale is provided.

Table 23: satisfaction of requirements by using blockchain technology

Requirement	Validation	Satisfaction (1-3) + Rationale
GDPR – VC1	Is any personal data of a subject only accessible by other entities with informed consent of the subject?	2 – data recorded on a blockchain is pseudonymous. However, on some blockchains the identifiers might be traced back to an IP-address, which classifies as personal data. Data recorded off the blockchain – such as verified claims – are shared with informed consent of the subject.
GDPR – VC2	Can any records of personal data owned by other entities than the subject be erased upon the request of the subject, given the conditions laid out in GPDR Art. 17 are met?	2 – no private claims or proofs should be recorded on a blockchain. Only a proof of its issuance or revocation is published on the blockchain. The entity only has to erase copies of personal data on its own servers, which is out of the SSI scope. However, if an entity publishes a public statement on the blockchain, it cannot be erased.
GDPR – VC3	Can personal data be ported to other systems in an automated manner?	3 – personal data is stored in a machine-readable format in the subject's claims repository. This is agnostic of the blockchain, and can thus be ported to other systems.
GDPR – VC4	Is personal data securely and privately managed by design, through the use of the system's technology?	2 – data put on a blockchain is highly secure. However, due to the transparent nature of blockchains, in current protocol implementations the information on the ledger lacks privacy. Therefore, only non-private data should be recorded on a distributed ledger.
GDPR – VC5	Does the system provide means for subject to control consent to who they share personal data with? Can	3 – blockchains provide highly suitable methods to record and revoke consent. An immutable and granular proof of consent can be recorded on a public ledger. In similar

Requirement	Validation	Satisfaction (1-3) + Rationale
	consent be easily provided and revoked?	fashion, subjects can publish a consent revocation on the ledger.
PSD2 – VC1	Does the system provide the means to provide or facilitate Strong Customer Authentication?	3 – authentication of decentralized identifiers works with state-of-the-art cryptography. SCA mandates authentication through possession of a cryptographic key, activated through a combination of hardware modules and biometrics. Activation of the private key can be customized through extra software.
PSD2 – VC2	Does the system facilitate the means to only allow PISP's and AISP's access to a subject's PSP with informed and explicit consent of the subject, which at any time can be revoked?	2 – blockchain systems do not regulate the actual access to PSP's. However, they are very well suited to record and revoke a subject's consent for PISP's and AISP's to be granted access to their PSP's, similar to GDPR – VC5.
eIDAS – VC1	Can public institutions from European Member States verify a subject's data, originate from another Member State, on the system?	3 – due to the publicly accessible ledger a blockchain should operate on, data can be verified by any entity.
eIDAS – VC2	Does the system provide the technological means to sign data using advanced or qualified electronic signatures?	3 – blockchains support extensive public key cryptography, such as RSA, DSA, and ECDSA. A proof of a valid registrar of qualified public keys can be published on the ledger, to determine whether a signature comes from a qualified entity.
UC1 – FR1	Does the subject need no other entity to generate a key pair?	2 – the blockchain must either be permissionless or permissioned with rights to generate new key pairs, to satisfy this requirement.

Requirement	Validation	Satisfaction (1-3) + Rationale
UC1 – FR2	Can the subject generate identifiers without constraints?	3 – given UC-FR1 is satisfied, a subject can create as many key pairs as desirable.
UC1 – FR3	Does the subject have control over which data is associated with their identifiers?	2 – the subject can control which data is associated with independent identifiers, which are not related to an IDP. However, an IDP can still control which data must be supplied before associating a claim with an identifier.
UC1 – NFR1	Has the subject sole control over the private key?	2 – upon creation of a key pair, only the creator has knowledge and control over the private key. However, key management is a challenge for most users. Loss of private keys is therefore a distinct risk.
UC1 – NFR2	Are identifiers generated on a network of open-source nature, not owned by any one entity?	2 – this requirement is met on a public permissionless network. However, a permissioned blockchain is owned by one or more entities.
UC1 – NFR3	Upon loss of a private key, can the corresponding identifiers stay the same?	2 – only blockchain environments which support the use of smart contracts are able to satisfy this requirement. In the case smart contracts are deployed, control over an identifier and using an identifier can be separated.
UC2 – FR1	Can any entity attest to a claim?	3 – using the PKI of a blockchain network, entities can sign claims. See appendix C for an example of an attestation to a claim.
UC2 – FR2	Can subjects self-attest to statements about themselves?	3 – using the PKI of a blockchain network, subjects can sign their own claims.
UC2 – FR3	Does the proof of issuance or attestation of a claim require publication on a public ledger?	1 – blockchain as a technology does not require or enforce entities to publish proof of

Requirement	Validation	Satisfaction (1-3) + Rationale
		issuance or attestation on a ledger. It can however facilitate it.
UC2 – FR4	Can identity providers choose whether they re-use a (well-known) public address or a one-time pseudonymous address?	3 – multiple addresses can be generated on a blockchain network. Parties can choose whether they re-use an existing address, or create a new one for every interaction.
UC2 – FR5	Upon creating a claim, can the statements within it be minimized for its purpose?	2 – blockchain technology does not enforce the structure of statements within a claim. How these are created, is entirely up to the issuer and subject. However, standards will likely be developed for this matter.
UC2 – NFR1	Can entities associate an identifier with a human-readable name?	3 – naming services such as the Ethereum Naming Service (ENS) enable the option for human-readable names to resolve to a public key derived address.
UC2 – NFR2	Are claims and/or attestation separate objects, linked to an identifier?	3 – in the model described in figure 8, the public ledger manages identifiers, not personal data claims. Each claim can then be cryptographically linked to an identifier.
UC2 – NFR3	Are claims issued via a standardized, machine-readable format, with clearly defined semantics?	1 – every claim can be issued in the format the issuer wishes. The blockchain system does not enforce any format.
UC3 – FR1	Are relying parties able to ask for specific verifiable claims, possibly requiring specific statement types?	2 – relying parties can ask for any combination of statements. However, to verify them, each statement within a subject's claim must be separately signed by the issuer.
UC3 – FR2	Upon asserting a claim, can the subject freely choose from	3 – the subject can assert the credential he or she sees fit for purpose. However, the relying

Requirement	Validation	Satisfaction (1-3) + Rationale
	which source to prove the minimal needed statements?	party may request that statements are signed by a specific identity provider.
UC3 – FR3	Are claims asserted to a relying party via a personal repository of the subject, over which only the subject has control? Does the repository inform the subject with whom claims are shared, and the validity of their consent to share the claim?	3 – blockchain environments support the use of a personal repository, fully under control of the subject. These repositories can be made as simple or complex as the subject wishes. The repository can track with which entities claims are shared, and the validity of consent, by aggregating data which has been published on the blockchain.
UC3 – FR4	Provided the relying parties handle according the rules, is the subject made aware to who they have given consent to use their personal data? Is explicit consent of the subject required for a third party to share personal data with other entities?	1 – using a blockchain does not prevent entities to share data without consent of the subject. Additional software is needed for user-friendly consent management.
UC3 – NFR1	Is all personal data in personal claim repositories automatically encrypted, according to the highest industry standards?	1 – data is not automatically encrypted on every form of repository. A blockchain cannot enforce encryption of off-ledger data.
UC4 – FR1	Does the system provide a mechanism for identity providers to revoke claims or attestations?	3 – identity providers could publish a list of signature revocations on the ledger (Tobin, 2017, p. 7), or via smart contracts revoke individual signatures (Al-bassam, 2017).
UC4 – FR2	Can a relying party verify whether any given claim or	3 – relying parties are able to verify whether a given claim is signed, and verify if the signature is not on a revocation registrar.

Requirement	Validation	Satisfaction (1-3) + Rationale
	attestation to a claim has been revoked?	
UC5 – FR1	Does the system provide means for electronic authentication with varying authentication assurance levels?	3 – blockchains make use of decentralized PKI. Additional software may provide mechanisms with varying assurance levels to authenticate a user, by proving control of the private key of the blockchain key pair.
UC5 – FR2	Are entities required to authenticate themselves by proving possession and control over a private key?	3 – in current practice, subjects prove possession of a private key by inputting a passphrase. However, this method may be customized.
UC5 – NFR1	Does the system store no records of authentication?	3 – in a blockchain environment, authentication is done with a cryptographic request, to which the subject responds with a signed message. No record of these messages is stored on the blockchain.
UC6 – FR1	Does the system require consent before sharing data, and is proof of consent recorded on a public ledger?	2 – a blockchain does not enforce the subject to provide consent. However, it can – and is very well suited – to record a receipt of provided consent on a public ledger.
UC6 – FR2	Can a subject revoke consent to sharing data, and is a revocation recorded on a public ledger?	3 – subjects are at any time able to revoke consent to share data with another entity. Recording a proof of consent revocation is possible on the blockchain, so that any party can verify the event.
UC6 – NFR1	Upon request of data access, is the subject unmistakably made clear of the purpose and duration of needed access?	1 – this is generally a requirement which must be met through a procedure of data access requests. Blockchain technology does not aid

Requirement	Validation	Satisfaction (1-3) + Rationale
		in making sure that the subject is well informed.

5.2 Analysis of evaluation results

In total, the satisfaction level of 36 requirements were determined. Based on the scale of 1 to 3, a maximum satisfaction level of 108 could have been achieved. The evaluation resulted in a satisfaction level of 85 out of 108, translating into a percentage of 79%. As indicated in §3.2, it is important to note that each requirement has the same weighting. Table 24 below displays the satisfaction level per constraint and use-case.

Table 24: requirements satisfaction results

Constraint / Use-Case	Score out of Maximum	Satisfaction Percentage
Constraint: GDPR compliance	12 out of 15	80%
Constraint: PSD2 compliance	5 out of 6	83%
Constraint: eIDAS compliance	6 out of 6	100%
UC1: subject establishes a digital identity	13 out of 18	72%
UC2: identity provider issues a verified claim	19 out of 24	79%
UC3: subject asserts a verified claim	9 out of 15	60%
UC4: identity provider revokes a claim	6 out of 6	100%
UC5: entity authentication	9 out of 9	100%
UC6: subject authorizes another entity to access a selection of data	6 out of 9	67%
Total	**85 Out of 108**	**79%**

The following validation criteria have been evaluated with a score of 1:

- Does the proof of issuance or attestation of a claim require publication on a public ledger?
- Are claims issued via a standardized, machine-readable format, with clearly defined semantics?

- Provided the relying parties handle according the rules, is the subject made aware to who they have given consent to use their personal data? Is explicit consent of the subject required for a third party to share personal data with other entities?
- Is all personal data in personal claim repositories automatically encrypted, according to the highest industry standards?
- Upon request of data access, is the subject unmistakably made clear of the purpose and duration of needed access?

Based on these validation criteria, and the rationales which support their evaluation, there appears to be a common reason why blockchain – as a conceptual technology – does not seem to satisfy these requirements: unenforceable standard practices. This stresses that while blockchain technology provides a valuable technological foundation, additional processes and software must be built around it.

If we go back to Allen's ten principles for self-sovereign identity, as described in §2.1.5.1, blockchain provides a valuable technical foundation. An overview is presented in table 25.

Table 25: added value of blockchain technology per SSI principle

SSI principle	Added value of blockchain technology
Existence	Blockchain provides a decentralized PKI to enable subjects to create as many identifiers as they see fit
Control	Initiation of a transaction on a blockchain requires control over a private key, which can stay under sole control of the subject
Access	A blockchain can record proofs of existence of data, which are accessible by the data owner. However, a blockchain does not aid in providing access to the original data source, as putting personal data on a public ledger would harm privacy
Transparency	Transactions and algorithms on public blockchains are transparent by definition
Persistence	Only subjects with control over their private keys can revoke their identifiers in a blockchain network

Portability	A blockchain does not store any personal data, only proofs of off-ledger data objects are stored. Since the data objects are stored locally at entities, this makes the use of blockchain highly portable
Interoperability	Public blockchains ensure that any entity is able to interact with another entity's identifiers. Interoperability among multiple blockchains is a challenge, a decentralised identifier infrastructure may be of value for this challenge
Consent	Blockchain does not enforce registration of explicit consent, it does however provide excellent means to record proofs of consent, and verify validity of existent consent
Minimization	Blockchain does not enforce data minimization. However, it facilitates multiple privacy enhancing technologies, such as zero-knowledge disclosures, and pseudonymous identifiers
Protection	Secure identification and authentication are well-supported by blockchain environments through decentralized PKI

6 COMPARATIVE ANALYSIS OF SSI IMPLEMENATATIONS

In this chapter, we analyse and compare two practical implementations of blockchain enabled self-sovereign identity; Sovrin and uPort.

6.1 Solution description: Sovrin

The Sovrin Foundation aims to serve as a public utility by developing an identity layer for the internet to provide people, organisations and things with an identity they own and control. Sovrin makes use of a public permissioned distributed ledger, validated by a selected group of so-called 'Stewards', eligible and responsible to operate Sovrin nodes. The Sovrin foundation appoints these stewards, who have to come to a contractual, legally binding, agreement with the foundation.

The Sovrin technology stack consists of three layers (Reed, Law, & Hardman, 2016):

- Distributed ledger: Sovrin makes use of multiple public permissioned ledgers
- Agents: agents serve as the gateway between clients and the distributed ledgers
- Clients: the identity owners

6.1.1 Distributed ledger layer

Sovrin utilizes a public permissioned ledger to store root identity records. It makes use of the custom Plenum consensus mechanism, a byzantine fault tolerant style protocol. The ledger is operated by two types of nodes, validator and observer nodes. Validators run the Plenum algorithm to write new records to the ledger, while observer nodes maintain a read-only copy of the ledger to fulfil read requests from clients. Sovrin uses four different ledgers:

1. The Identity ledger: this ledger is the foundation, and keeps a record off all identity records. This includes a decentralized PKI system, and all records written by the decentralized identifiers (DID's).
2. The Pool ledger: since Sovrin makes use of permissioned ledgers, the pool ledger records the permission levels for each active node. Moreover, the Pool ledger stores the outcome of node votes on the Voting ledger.

3. The Voting ledger: this is where votes among trustees and stewards are held to propose, confirm, or revoke different permissions, e.g., whether a node is permitted to serve as a validator or observer node.
4. The Configuration ledger: this ledger holds network-wide configuration data set by the Sovrin Foundation Technical Governance Board and approved by the Board of Trustees.

Sovrin advises to put the following information on their ledgers: (1) decentralized identifiers, (2) schemas and claim definitions, (3) proof of consent for data sharing, (4) public claims, and (5) revocation registries. It is strongly misadvised to put any private data on the Sovrin ledgers.

6.1.2 Agents

Agents may facilitate interaction between clients and the Sovrin ledger. The core functions of agents are:

1. Peer-to-peer messaging endpoints: like an IP-address to computers, or a phone number to mobile phones, agents can serve as the endpoints to address Sovrin identities. A Sovrin identity can use as many endpoints as suitable.
2. Coordination endpoints for multiple clients: agents may aggregate multiple 'edge' clients – a mobile phone, laptop, car – into one addressable endpoint.
3. Encrypted back-up of Sovrin keyrings: maintain an encrypted back-up of private keys.
4. Encrypted data storage and sharing: manage and automate data storing and sharing, while still under full control by the private key of the identity owner.

6.1.3 Clients

Sovrin's entity architecture is displayed in figure 9 below (Best et al., 2017):

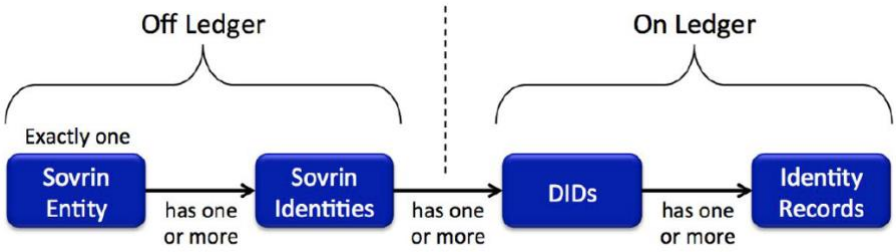

Figure 9: Sovrin entity architecture (Best et al., 2017, p. 7)

To preserve the privacy of their clients, Sovrin separates the DID's, which are on the identity ledger, from the Sovrin identities. Each Sovrin identity is made up by one or more DID's. A DID can be of various types, as shown in figure 10 below. A DID might be an ''anonym'', which are anonymously generated and are the default type of identifier. A ''verinym'' is used by Trust Anchors[7]. The Trust Anchor verinym is the DID of the Trust Anchor itself, the Anchored verinym is the DID of an identity owner whose existence has been authorized by a Trust Anchor.

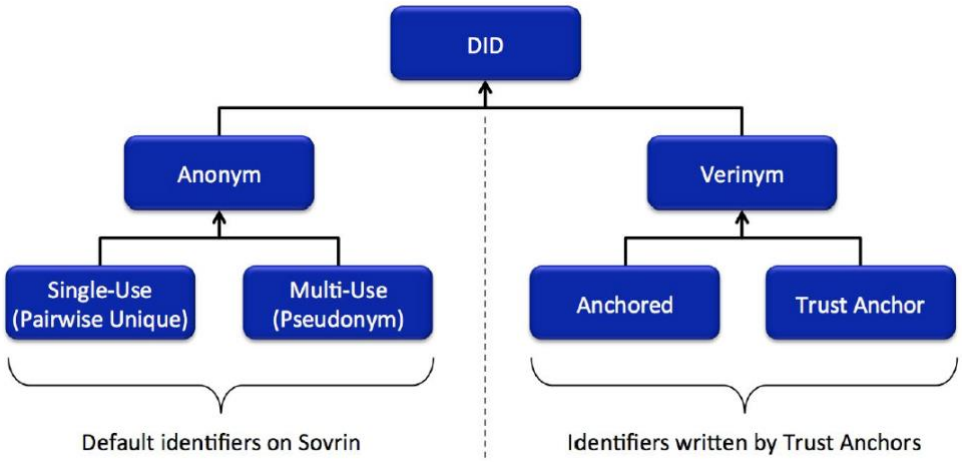

Figure 10: Sovrin identifier typology (Best et al., 2017, p. 8)

[7] Trust Anchor: "Individuals or Organizations for whom there is sufficient public evidence of their trustworthiness and accountability"

6.1.4 Sovrin in practice

The Sovrin architecture is based on identifiers, claims, disclosures, and proofs. We will now illustrate an example of how an identity owner might use Sovrin[8], by describing how each use-case as described in §4.2.2 would work on the Sovrin network. For each use-case, we take Bob as an example, who interacts with his university.

6.1.4.1 Establish a digital identity

Bob wants to establish a digital relationship with his University. Therefore, he wants to create a decentralized identifier on the Sovrin network, so that he may associate verifiable claims and data with it. To generate a decentralized identifier to use with his University, Bob generates a unique verification (VKbob) and signing key (SKbob) on Sovrin. He then presents his VK to his university. In turn, the university has a key pair associated with its verinym, VKuni and SKuni. Bob and the university exchange their verification keys VKbob and VKuni, of which they can verify the validity on the Sovrin identity ledger.

6.1.4.2 Issue verified claims

Bob has created his unique identifier associated with his university, and now wishes to receive a verifiable claim from the university, to state that Bob is indeed an enrolled student. Upon requesting the verifiable claim from his university, Bob first authenticates himself by proving he has control over VKbob, by signing a message with SKbob. The university is now convinced that Bob has control over VKbob, which according to their records is indeed an enrolled student. The university now creates a claim, consisting of multiple statements and a reference to a claim definition. Each statement within the claim – Bob's key VKbob, Bob's name, his student number, the program he is enrolled in, the valid period of the claim – is separately signed by the university's verification key. The claim definition reference points to a description of the semantics for each statement within the claim. Bob now has a verifiable claim, stating that he is a currently enrolled student at the University. The university publishes a proof-of-existence of the claim on the Sovrin ledger, so that any relying party may verify the validity of the Bob's claim.

---

[8] A visual example is also provided by a gif, found on the following URL: http://www.windley.com/archives/2016/10/Sovrin-Animation.gif

6.1.4.3 Assert a verified claim

Bob wishes to prove to the cinema customer service that he is a university student, so that he is eligible for student discount. To accomplish this, he will disclose the verified claim from his university. To do this with maximum privacy, Bob only selects the following statement within his claim: {isStudent: TRUE}. Via a zero-knowledge proof[9], Bob discloses that he is indeed a student at the university, without revealing any other information present in his claim from the university. The cinema can easily verify that the statement "isStudent" was signed by Bob's university, by looking up the proof in the Sovrin ledger.

6.1.4.4 Revoke a claim

Bob has finished his studies at the university. Hence, the university wishes to revoke their claim that Bob is an enrolled student. With Sovrin, the university maintains a revocation registry on the identity ledger. This revocation registry is specific for each claim definition, in this instance the claim definition might be called "enrolledStudent". The university now updates their revocation registry, adding a revocation of Bob's "enrolledStudent" claim. Using advanced cryptography – which will not be explained here, thus treated as a black box – Bob is no longer able to prove that his claim is valid, since it is on the revocation registry. In a similar fashion, upon verifying Bob's claim, relying parties will match it against the university's revocation registry and get returned a message that Bob's claim has been revoked by the issuer.

6.1.4.5 Entity authentication

If Bob has to authenticate himself before interaction with the university, he can use his Sovrin identity. Bob already generated key pair to interact with the university in §6.1.4.1, of which he will now use the corresponding private key to authenticate himself. The university sends a message to Bob, in the form of a one-time challenge string. Bob can now sign this string with his private key, and send the result back to the university. Using public key verification, the university can now verify that indeed the message was signed by Bob, since only Bob's private key would be able to produce a signature which can be verified with Bob's public key.

---

[9] In cryptography, "zero-knowledge proofs are defined as those proofs that convey no additional knowledge other than the correctness of the proposition in question" (Goldwasser, Rackoff, & Micali, 1989).

6.1.4.6 Provide access to personal data to another entity

Bob completed the course 'cybersecurity 101', and received a proof of this event in the form of a pdf file from his university. Bob now wants to share this proof with a company he wants to interview at. The interviewing process will take one month, so Bob feels it is reasonable to share his proof with the company for a month. With Sovrin, Bob will want to receive a consent receipt from the company, stating that they will only use Bob's proof of his course completion for interviewing processes, and that they will only use it for the period of one month.

The consent receipt contains the following information (Tobin, 2017):

- Bob's identifier
- The company's identifier
- A name and/or location reference to the data that was shared
- The conditions on which the data is shared
- A signature by Bob
- A signature by the company

Now, a proof of existence of this consent receipt – in the form of a cryptographic hash – will be recorded on the Sovrin identity ledger. Both Bob and the company can now reference this consent receipt in the case of a dispute.

6.2 Solution description: uPort

uPort is a self-sovereign identity solution built on the Ethereum network. The following description of the uPort solution will rely on their whitepaper (Heck, Torstensson, Mitton, & Sena, 2017) and various GitHub repositories (UPort, 2017), as little other information has been published.

uPort consists of three main technical components; smart contracts, a mobile application, and developer libraries. With regard to the scope of this research, the first two components will be discussed. Developer libraries provide the means for developers to integrate uPort with existing solutions.

6.2.1 Smart contracts

uPort is deployed on the Ethereum blockchain. Ethereum is a platform for smart contract development, as has been discussed in §2.2.7. uPort makes use of four main smart contracts:

- Proxy contracts: normally, blockchain transactions and messages are generated by the key pair of an account. Proxy contracts form a middle layer between the initiation of a message by its owner and the target of the message, acting as a gateway for interactions. The proxy contract has two main functionalities: (1) forwarding Ethereum transactions and (2) replacing the owner of the proxy contract. As of now, each user will only have one proxy contract, and thus one digital identity.
- Controller contracts: this contract embeds the controlling logic, based on which the identity owner can authenticate him- or herself to the proxy contract. An important feature of this contract, is that recovery logic is embedded. In the case of loss of the private key, the user can replace control of the proxy contract with another controller contract. An example of how this might be done is through social recovery, where three out of five other persons must sign a message, before the controller contract is replaced.
- Application contracts: the contract to receive messages from the proxy contracts.
- Registry contract: this is a trusted contract, which keeps track of bindings between a user's proxy contracts and off-chain data objects – such as verifiable claims or proofs – linked to it. A record on the registry contract is a mapping between the public address of the issuer, the public address of the receiver, and the address of a data

object stored on IPFS, OneDrive, local storage, or other data storage systems. uPort formats data objects as either unsigned JSON structures (a self-claimed attribute), or signed JSON web tokens (a verifiable claim).

6.2.2 Mobile application

The mobile application holds the private keys of the user, preferably in the secure enclave. Therefore, the mobile application is used to establish digital identities by sharing identifiers, authenticate users through proving possession of the private key, and to initiate transactions. Using the biometric technologies embedded in the mobile phone – fingerprint, facial recognition, retina scans – advanced methods of private key unlocking can be built in.

6.2.3 uPort in practice

The uPort architecture is based on smart contracts. These contracts give the user enhanced methods of identity control, and enable secure exchange of identity information. We will now illustrate an example of how an identity owner might use the uPort self-sovereign identity system, by describing how each use-case as described in §4.2.2 would work on the uPort network. For each use-case, we take Alice as an example, who interacts with her bank.

6.2.3.1 Establish a digital identity

Alice would like to become a customer at a bank. To associate Alice's existing uPort identity with her bank, she shares her uPort identifier – the address of her proxy contract. The bank can now associate Alice's uPort identifier with their internal records.

6.2.3.2 Issue verified claims

Alice has shared her uPort identifier with her bank, and now wishes to receive a verifiable claim from the bank, to state that Alice has an active bank account. Upon requesting the verifiable claim from her bank, Alice first authenticates herself by proving she has control over the proxy contract, by signing a message through the controller contract. The bank is now convinced that Alice has control over Alice's uPort identifier, which according to their records indeed has an active bank account. The bank now creates a verifiable claim, consisting of multiple statements and a reference to a claim definition. Each statement within the claim – Alice's identifier 'addressProxy', Alice's name, her bank account number, the valid period of the claim – is separately signed by the bank's public key. The claim definition reference points to a description

of the semantics for each statement within the claim. Alice now has a verifiable claim, stating that she has an active bank account at the bank. The bank sends the verifiable claim to Alice, and publishes a proof-of-existence of the claim via the bank's registry contract[10], so that any relying party may verify the validity of the Alice's claim. Alice stores the verifiable claim on a personal repository, and can load the claim into her uPort mobile application.

6.2.3.3 Assert a verified claim

Alice wishes to prove to a credit provider that she has an active bank account, so that she might be eligible to receive a loan. To accomplish this, she will disclose the verified claim from her bank. To do this with maximum privacy, the bank has to send Alice a request which only asks for the specific attribute {hasBankAccount: TRUE}.

6.2.3.4 Revoke a claim

Alice no longer has an active bank account at her bank. Hence, the bank wishes to revoke their claim that Alice has an active bank account. Unfortunately, with uPort this is not possible by revoking the claim itself. Instead, the bank must have put in a validity period upon issuing the claim. When this period ends, the claim is deemed invalid.

6.2.3.5 Entity authentication

If Alice has to authenticate herself before interaction with the bank, the bank generates a random challenge string embedded in a QR code. Alice can now scan this QR code with her uPort mobile application. The mobile application will attempt to sign the challenge string with Alice's private key. Hence, Alice has to authorize this action, through unlocking the private key stored on her mobile phone. Multiple techniques to accomplish this are possible, however, biometric authentication is the most secure. Once the challenge string is signed and returned to the bank, the bank can easily verify if Alice signed the string via public key verification.

---

[10] A recent change has been made as to which data is associated with registry contracts. Now, only the public key of the signer is stored on IPFS. A record of the IPFS hash is then bound to the registry contract, for public key verification purposes. See https://medium.com/uport/private-data-on-public-networks-ab1086a137d8 for more information.

6.2.3.6 Provide access to personal data to another entity

If the bank requests a copy of her passport, Alice can share that via her uPort mobile application. Alice uploads a hash of her passport file onto IPFS, or any other cloud system. Next, she can consent to sharing her passport, by creating a new registry mapping in her registry contract. This will take on three values; the issuing address (Alice), the subject address (the bank), and a value (the hash of her passport file). When this consent action is registered in her registry contract, Alice can send the bank her passport. If the wishes to revoke access, Alice can update her registry contract, to delete the subject address.

6.3 Requirement satisfaction of Sovrin and uPort

Similar to §5.1, the requirement satisfaction of Sovrin and uPort will be determined. Instead of using partial satisfaction levels, we now only have two options; either the requirement is satisfied, or it is not. The reason for this, is that we now do not evaluate a conceptual technology, but full-stack solutions. Sovrin and uPort both make use of blockchain technology, and have additional software and processes built around it. The results of the requirements satisfaction can be found in table 26.

Table 26: Requirement satisfaction of Sovrin and uPort

Requirement / Constraint	Validation Criterion	Satisfaction: 0 - 1	
		Sovrin	uPort
GDPR – VC1	Is any personal data of a subject only accessible by other entities with informed consent of the subject?	1	1
GDPR – VC2	Can any records of personal data owned by other entities than the subject be erased upon the request of the subject, given the conditions laid out in GPDR Art. 17 are met?	1	1
GDPR – VC3	Can personal data be ported to other systems in an automated manner?	1	1
GDPR – VC4	Is personal data securely and privately managed by design, through the use of the system's technology?	1	1
GDPR – VC5	Does the system provide means for subject to control consent to who they share personal data with? Can consent be easily provided and revoked?	1	0
PSD2 – VC1	Does the system provide the means to provide or facilitate Strong Customer Authentication?	1	1
PSD2 – VC2	Does the system facilitate the means to only allow PISP's and AISP's access to a subject's PSP with informed and explicit consent of the subject, which at any time can be revoked?	1	0

Requirement / Constraint	Validation Criterion	Satisfaction: 0 - 1	
		Sovrin	uPort
eIDAS – VC1	Can public institutions from European Member States verify a subject's data, originative from another Member State, on the system?	1	1
eIDAS – VC2	Does the system provide the technological means to sign data using advanced or qualified electronic signatures?	1	1
UC1 – FR1	Does the subject need no other entity to generate a key pair?	0	1
UC1 – FR2	Can the subject generate identifiers without constraints?	1	0
UC1 – FR3	Does the subject have control over which data is associated with their identifiers?	1	1
UC1 – NFR1	Has the subject sole control over the private key?	1	1
UC1 – NFR2	Are identifiers generated on a network of open-source nature, not owned by any one entity?	0	1
UC1 – NFR3	Upon loss of a private key, can the corresponding identifiers stay the same?	0	1
UC2 – FR1	Can any entity attest to a claim?	1	1
UC2 – FR2	Can subjects self-attest to statements about themselves?	1	1
UC2 – FR3	Does the proof of issuance or attestation of a claim require publication on a public ledger?	1	1
UC2 – FR4	Can identity providers choose whether they re-use a (well-known) public address or a one-time pseudonymous address?	1	0
UC2 – FR5	Upon creating a claim, can the statements within it be minimized for its purpose?	1	1
UC2 – NFR1	Can entities associate an identifier with a human-readable name?	0	1

Requirement / Constraint	Validation Criterion	Satisfaction: 0 - 1	
		Sovrin	uPort
UC2 – NFR2	Are claims and/or attestation separate objects, linked to an identifier?	1	1
UC2 – NFR3	Are claims issued via a standardized, machine-readable format, with clearly defined semantics?	1	1
UC3 – FR1	Are relying parties able to ask for specific verifiable claims, possibly requiring specific statement types?	1	1
UC3 – FR2	Upon asserting a claim, can the subject freely choose from which source to prove the minimal needed statements?	1	1
UC3 – FR3	Are claims asserted to a relying party via a personal repository of the subject, over which only the subject has control? Does the repository inform the subject with whom claims are shared, and the validity of their consent to share the claim?	1	1
UC3 – FR4	Provided the relying parties handle according the rules, is the subject made aware to who they have given consent to use their personal data? Is explicit consent of the subject required for a third party to share personal data with other entities?	0	0
UC3 – NFR1	Is all personal data in personal claim repositories automatically encrypted, according to the highest industry standards?	0	0
UC4 – FR1	Does the system provide a mechanism for identity providers to revoke claims or attestations?	1	0
UC4 – FR2	Can a relying party verify whether any given claim or attestation to a claim has been revoked?	1	1
UC5 – FR1	Does the system provide means for electronic authentication with varying authentication assurance levels?	1	1

Requirement / Constraint	Validation Criterion	Satisfaction: 0 - 1	
		Sovrin	uPort
UC5 – FR2	Are entities required to authenticate themselves by proving possession and control over a private key?	1	1
UC5 – NFR1	Does the system store no records of authentication?	1	1
UC6 – FR1	Does the system require consent before sharing data, and is proof of consent recorded on a public ledger?	1	0
UC6 – FR2	Can a subject revoke consent to sharing data, and is a revocation recorded on a public ledger?	1	0
UC6 – NFR1	Upon request of data access, is the subject unmistakably made clear of the purpose and duration of needed access?	1	1
	Total Score out of 36	30	27

Based on the results, we can conclude that Sovrin currently satisfies most of the requirements. Areas where Sovrin could improve so that they may meet all requirements, are:

- Implement a private key revocation scheme, where corresponding identifiers do not have to be replaced.
- Implement a naming system, similar to the Ethereum Naming System, so that identifiers may resolve to human-readable names.
- Provide verifiable claims repository software with automatic encryption, so that subjects can rely on secure storage of their personal data.
- Move towards a public permissionless network model, so that all the subjects are truly self-sovereign, and can operate on the network without constraints.

7 CONCLUSION

This research has contributed to the theoretical and societal knowledge base of the Self-Sovereign Identity concept. Current Self-Sovereign Identity management ecosystems rely on using blockchain technology. Blockchain is the emergent technology behind cryptocurrencies, which provides an immutable and transparent shared ledger. Over the last two years, it has become clear that cryptocurrency is not the only viable use-case for blockchain technology. Many problems of current identity management practices are rooted in centralised data management, lack of privacy, and the need to trust large institutions. Due to its technological characteristics, blockchain technology is well-positioned to be the initiator of an influential change in digital identity.

This study has aimed to answer the following research question: **"What are the system requirements for a regulatory compliant Self-Sovereign Identity information system, and how can blockchain technology solutions serve as a solid foundation?"**

Based on the conducted interviews with stakeholders from various industries – including financial services, private research, and government – a set of system requirements and constraints have been developed in §4.2. Three constraints have been identified; compliance with the pan-European GDPR, eIDAS, and PSD2 regulations. Moreover, six use-cases have been identified, based on which requirements have been constructed; (1) establish a digital identifier, (2) issue verified claims, (3) assert a verified claim, (4) revoke a claim, (5) entity authentication, and (6) provide access to personal data to another entity. The complete set of requirements and constraints have been validated by the interviewees and the two primary supervisors of this research. Using a requirement satisfaction evaluation, including the possibility of partial satisfaction, we have concluded that blockchain technology can serve as a valuable technological basis for self-sovereign identity management. The evaluation results indicated that two out of three constraints and four out of six use-cases need additional technology or processes to fully satisfy the requirements.

Thus, blockchain technology *can* serve as the solid foundation for Self-Sovereign Identity. A blockchain provides incomparable security, resulting in significant improvements over current centralised or federal identity management practices. The Self-Sovereign Identity ecosystem is based on the exchange of verifiable claims, which are stored off-ledger. The integrity of these

signed data objects is ensured through storing a hash of the object on a blockchain. When the subject presents his verifiable claim to a relying party, the relying party can easily verify the validity of the claim comparing the hash of the claim with the record on the blockchain, and verifying the included signature. Next to ensuring integrity of data objects, a blockchain provides the means for entities to record and revoke an auditable log of consent actions. Especially for companies preparing to become compliant with GDPR and PSD2, being able to prove explicit and informed consent of their customers is of vital importance. Lastly, since blockchain is based on a decentralised public key infrastructure, strong cryptographic authentication methods can be employed.

To provide a deeper insight in how current self-sovereign identity solutions work, and to what extent they satisfy the requirements, two practical implementations have been analysed and compared. Two solutions have been selected; Sovrin and uPort. They operate on different networks, with different design decisions. Sovrin makes use of a public permissioned blockchain, while uPort operates on the public permissionless Ethereum blockchain. They both have varying architectures, and make use of different technology stacks. Based on the requirements satisfaction evaluation, we can conclude that at this moment Sovrin has an advantage over uPort. Sovrin satisfied 30 out of 36 requirements, while uPort satisfied 27 out of 36 requirements. Based on these findings, we can conclude that indeed, blockchain can serve as a solid foundation for a self-sovereign identity ecosystem. Blockchain aids in maintaining integrity of personal data and providing subjects the freedom to privately exchange verifiable claims about their identity, with a lower need for trust in large institutions. However, it became evident that blockchain is not a meta-solution, additional technology is needed.

8 DISCUSSION AND RECOMMENDATIONS

This research report provided the reader a deeper insight into the concept of self-sovereign identity. It is important to note that the domain of identity management is still very much shaping the ideal architecture. Blockchain technology has spurred innovation, urging identity management professionals to research how the technology may enhance current identity management practices. The idea of what "self-sovereign identity" must be has changed considerably over the last two years. The general consensus has shifted from putting all our personal data on a blockchain, to only recording hashes of data onto the chain. Blockchain in its current form is not a supernatural solution to all problems, rather, it has multiple drawbacks. The technology is not scalable for widespread use, quantum computing might pose a risk to current cryptographic algorithms, privacy is still an issue, and transactions are costly.

However, as has been demonstrated in this research, blockchain-enabled self-sovereign identity is a vast improvement over current identity management practices. Subjects are empowered to share personal data when, with who, and on which conditions they deem appropriate. Although subjects are self-sovereign, that does not imply total independence. Identity providers are still required to issue verifiable claims, relying parties still have to inspect claims and be able to verify who signed them, and regulators still need an auditable log of data.

We have added to the practical base a validated set of 36 system requirements and constraints, each with validation criteria. This set can be used to deduct specific software requirements for self-sovereign identity solutions. Moreover, the ecosystem of self-sovereign identity management has been described and analysed, adding to the theoretical knowledge base.

Based on the findings of this research and the conclusions derived from them, Iquality Business Solutions is recommended to further build upon existing frameworks. Self-sovereign identity is not only about creating blockchain infrastructure, there is a need for middleware and supporting software to create a seamless experience for the general public. In the Netherlands, the members of the Techruption project and the National Blockchain Coalition can be a solid business partner. Based on their ongoing work, Iquality Business Solutions can position itself to provide software solutions tailored to the state-of-the-art self-sovereign identity framework.

9 LIMITATIONS AND FURTHER RESEARCH

This research has been carried out over a six-month period. It is of utmost importance to note that it is an exploratory study, as the concept of self-sovereign identity is new and very much considering what its architecture should look like. The underlying technology, blockchain, has gained traction only a few years back, and is bound to undergo significant design changes. As both self-sovereign identity and blockchain technology are new research domains, this study is limited by a lack of prior academic research. To counter this limitation, academic research on traditional identity management and current – mostly non-academic – institutional blockchain research has been used as a theoretical framework.

The system requirements have been developed by introspection and semi-structured interviews with experts from various industries. Validation has been performed by the same experts. A limitation of this study is that the system requirements could not be validated by a larger sample, across more companies or industries. Furthermore, the set of requirements as presented in this report are not exhaustive. A self-sovereign identity system will likely need to satisfy additional requirements to be well-equipped for operational use. The complete set of requirements have not been weighted. As indicated in §3.3, this is a limitation of the study. Determining weighting of each requirement would improve the analysis, however, due to time constraints, this has not been done.

Moreover, this research studies regulatory compliance of a self-sovereign identity system with three pan-European regulations; the GPDR, PSD2, and eIDAS. However, the legal framework around the use of blockchain technology is yet to be defined. Especially in the case of a public permissionless blockchain, there is uncertainty about accountability and jurisdiction issues. Next to the legal framework, how GDPR, PSD2 and eIDAS will be implemented and enforced is still subject to change.

Because of legal uncertainty, multiple opportunities for future research exist. A framework for the legislative implications of public permissionless blockchains might be developed. Research questions which could arise are:

- Who is responsible in the case of failure of a smart contract?
- Who are the data controllers and processors?
- How do we control publicly visible personal data on a public blockchain?
- Which laws apply and in what jurisdictions?

Next to the legal field, further research may be conducted on the technical foundations of blockchain technology. Much work is already being done in the area of scalability and privacy. The effects of scalability solutions, such as state channel technology and new consensus mechanisms, on the viability of widespread self-sovereign identity usage might be quantitatively determined. Moreover, the application of new cryptographic protocols such as ring signatures, zero-knowledge (range) proofs in identity management practices should be extensively investigated.

REFERENCES

Al-bassam, M. (2017). SCPKI : A Smart Contract-based PKI and Identity System. In *BCC '17 Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* (pp. 35–40). ACM. https://doi.org/10.1145/3055518.3055530

Alkhalifah, A., & D'Ambra, J. (2015). Identity Management Systems Research: Frameworks, Emergemce, and Future Opportunities. In *European Conference on Information Systems* (p. 16). ECIS.

Allen, C. (2016). Self-Sovereign Identity Principles. Retrieved September 3, 2017, from https://github.com/ChristopherA/self-sovereign-identity/blob/master/self-sovereign-identity-principles.md

Armando, A., Carbone, R., Compagna, L., Cuellar, J., & Tobarra, L. (2008). Formal Analysis of SAML 2.0 Web Browser Single Sign-on: Breaking the SAML-based Single Sign-on for Google Apps. In *Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering* (pp. 1–10). ACM. https://doi.org/10.1145/1456396.1456397

Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In *International Conference on Principles of Security and Trust* (pp. 164–186). Berlin: Springer. https://doi.org/10.1007/978-3-662-54455-6_8

Baars, D. (2016). *Towards Self-Sovereign Identity using Blockchain Technology*. University of Twente.

Baier, D., Bertocci, V., Brown, K., Densmore, S., Pace, E., & Woloski, M. (2010). *A Guide to Claims-Based Identity and Access Control: Patterns & Practices* (2nd ed.). Microsoft Press.

Baliga, A. (2017). Understanding Blockchain Consensus Models. Persistant Systems. Retrieved from https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf?pdf=Understanding-Blockchain-Consensus-Models

Baran, P. (1964). On distributed communications networks. *IEEE Transactions on Communications Systems*, *12*(1), 1–9.

Bertino, E., & Takahashi, K. (2011). *Identity Management: Concepts, Technologies, and Systems. Artech House* (1st ed.). Artech House.

Best, J., Boldrin, L., Brown, T., Conway, S., Chango, M., David, S., ... Windley, P. (2017). Sovrin Provisional Trust Framework. Sovrin Foundation.

Birch, D. (2016). Putting "identity" on the "blockchain". Retrieved September 1, 2017, from http://www.chyp.com/putting-identity-on-the-blockchain-part-2-create-an-identity-model/

Buterin, V. (2015). On Public and Private Blockchains - Ethereum Blog. Retrieved September 19, 2017, from https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

Buterin, V. (2017). Blockchains and Privacy through Strong Cryptography. Retrieved October 31, 2017, from https://www.youtube.com/watch?v=9cDFpACnK1U

Buterin, V., & Griffith, V. (2017). Casper the Friendly Finality Gadget. Ethereum Foundation. Retrieved from http://arxiv.org/abs/1710.09437

Camp, J. L. (2004). Digital Identity. *IEEE Technology and Society Magazine*, *23*(3), 34–41. https://doi.org/10.1109/MTAS.2004.1337889

Chung, L., Nixon, B., Yu, E., & Mylopoulos, J. (2000). *Non-functional requirements in software engineering* (1st ed.). Massachusetts, USA: Kluwer academic publishers.

Commission, E. Implementation Act 2015/1502, 235 Offical Journal of the European Union § (2015). European Union.

Cuijpers, C. M. K. C., & Schroers, J. (2015). eIDAS as guideline for the development of a pan European eID framework in FutureID. *GI-Edition Lecture Notes in Informatics*, *2014*, 23–38. Retrieved from https://pure.uvt.nl/portal/en/publications/eidas-as-guideline-for-the-development-of-a-pan-european-eid-framework-in-futureid(2b0a04f5-78a7-479e-bea2-8831b42b176b).html

Daitch, H. (2016). 1 Billion Yahoo Accounts Compromised in Data Breach | IdentityForce®. Retrieved September 19, 2017, from https://www.identityforce.com/blog/one-billion-yahoo-accounts-compromised-new-data-breach

Davies, J. (2016). The eIDAS regulation | What does eIDAS mean for you? - Signable. Retrieved November 1, 2017, from https://www.signable.co.uk/blog/what-does-eidas-mean

De Kruijff, J., & Weigand, H. (2017). Understanding the Blockchain Using Enterprise Ontology. In *International Conference on Advanced Information Systems Engineering* (pp. 29–43). Springer, Cham.

Dhamija, R., & Dusseault, L. (2008). The Seven Flaws of Identity Management: Usability and Security Challenges. *IEEE Security & Privacy*, *6*(2). https://doi.org/10.1109/MSP.2008.49

Dworkin, M. J. (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. *Federal Inf. Process. Stds.(NIST FIPS)-202.* https://doi.org/10.6028/NIST.FIPS.202

Eriksson, O., & Agerfalk, P. J. (2010). Rethinking the Meaning of Identifiers in Information Infrastructures. *Journal of the Association for Information Systems*, *11*(8), 433–454. https://doi.org/Article

European Banking Authority. (2017). *Final report on the draft RTS on SCA and CSC under PSD2*. Retrieved from https://www.eba.europa.eu/documents/10180/1761863/Final+draft+RTS+on+SCA+and+CSC+under+PSD2+%28EBA-RTS-2017-02%29.pdf

European Parliament. (2015a). Directive 2015/2366 (Payment Service Directive 2). *Official Journal of the European Union*, *L 337/35*(260), 35–127.

European Parliament. Electronic identification and trust services for electronic transactions in the internal market (2015). European Union.

European Parliament. Regulation 2016/679 of the European parliament and the Council of the European Union, Official Journal of the European Communities § (2016). https://doi.org/http://eur-lex.europa.eu/pri/en/oj/dat/2003/l_285/l_28520031101en00330037.pdf

European Parliament. (2017). Revised rules for payment services in the EU. Retrieved September 19, 2017, from http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=LEGISSUM:2404020302_1&from=EN&isLegissum=true

Farah, N. (2017). What's the Difference Between Advanced and Qualified Signatures in eIDAS? Retrieved November 1, 2017, from https://www.globalsign.com/en/blog/difference-between-eidas-advanced-and-qualified-electronic-signatures/

Fenton, J., Danker, J. M., Greene, K. K., & Theofanos, M. F. (2017). *DRAFT NIST Special Publication 800-63a Digital Identity Guidelines: enrollment and proofing*. Los Altos, CA.

Fromknecht, C., & Velicanu, D. (2014). A Decentralized Public Key Infrastructure with Identity Retention. *IACR Cryptology ePrint Archive*, *803*, 1–16.

Goldwasser, S., Rackoff, C., & Micali, S. (1989). Interactive Proof Systems Definitions. *Society for Industrial and Applied Mathematics*, *18*(1), 108–128.

Grassi, J., Fenton, J., Newton, E., Perlner, R., Regenscheid, A., Burr, W., & Richer, J. (2017). *DRAFT NIST Special Publication 800-63b Digital Identity Guidelines: Authentication and Lifecycle Management*. Los Altos, CA. https://doi.org/10.6028/NIST.SP.800-63b

Grewal-Carr, V. (Deloitte), & Marshall, S. (Deloitte). (2016). Blockchain Opportunity Contents. Deloitte LLP.

Gubik, M., & Buterin, V. (2017). Proof of Stake FAQ. Retrieved September 19, 2017, from https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ#what-is-proof-of-stake

Halperin, R. (2006). Identity as an emerging field of study. *Datenschutz Und Datensicherheit-DuD*, *30*(9), 533–537. Retrieved from

http://www.springerlink.com/index/T55533325G45W857.pdf

Harris, S., & Stonebraker, A. (2015). *CISSP Exam Guide* (7th ed.). McGraw Hill.

Heck, R., Torstensson, J., Mitton, Z., & Sena, M. (2017). *uPort: a Platform for Self-Sovereign Identity*. Retrieved from https://whitepaper.uport.me/uPort_whitepaper_DRAFT20170221.pdf

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2), 87–92. https://doi.org/http://aisel.aisnet.org/sjis/vol19/iss2/4

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quaterly*, *28*(4), 75–105.

International Telecommunications Union. (2017). Global Internet Usage. United Nations. Retrieved from http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2017/Stat_page_all_charts_2017.xls

ITU-T. (2009). NGN identity management framework, Recommendation Y.2720. *Y.2720*, 1–34.

Jacobs, S. (2011). *Engineering information security: the application of systems engineering concepts to achieve information assurance* (14th ed.). John Wiley & Sons.

Johnson, D., Menezes, A., & Vanstone, S. (2001). The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, *1*(1), 36–63. https://doi.org/10.1007/s102070100002

Joosten, R. (2017). Self-Sovereign Identity Framework and Blockchain. Retrieved October 27, 2017, from https://ercim-news.ercim.eu/en110/special/self-sovereign-identity-framework-and-blockchain

Jøsang, A., & Pope, S. (2005). User centric identity management. In *AusCERT Asia Pacific Information Technology Security Conference* (p. 77). https://doi.org/10.1109/MSP.2007.99

Lapouchnian, A. (2005). *Goal-Oriented Requirements Engineering : An Overview of the Current Research*. Toronto.

Laurent, M., & Bouzefrane, S. (2015). *Digital Identity Management. Architecting User-Centric Privacy-as-a-Set-of-Services*. https://doi.org/10.1007/978-3-319-08231-8_3

Letier, E., & Lamsweerde, A. Van. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. In *ACM SIGSOFT Software Engineering Notes* (pp. 53–62). ACM. https://doi.org/10.1145/1041685.1029905

Luu, L., Chu, D.-H., Olickel, H., Saxena, P., & Hobor, A. (n.d.). Making Smart Contracts

Smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 254–269). ACM. https://doi.org/10.1145/2976749.2978309

Maler, E., & Reed, D. (2008). The venn of identity: Options and issues in federated identity management. *IEEE Security and Privacy*, *6*(2), 16–23. https://doi.org/10.1109/MSP.2008.50

Mills, D., Wang, K., Malone, B., Ravi, A., Marquardt, J., Chen, C., ... Kar-Genian, V. (2016). Distributed ledger technology in payments, clearing, and settlement. *Butterworths Journal of International Banking and Financial Law*, *31 (11)*, 36. https://doi.org/10.17016/FEDS.2016.095

Mougayar, W. (2016). The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology. Wiley & Sons.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, *17*(1), 2–26. https://doi.org/10.1016/j.infoandorg.2006.11.001

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. https://doi.org/10.1007/s10838-008-9062-0

Narayanan, A., & Shmatikov, V. (2010). Myths and fallacies of "personally identifiable information." *Communications of the ACM*, *53*(6), 24. https://doi.org/10.1145/1743546.1743558

Nunamaker, J. F. J., Chen, M., & Purdin, T. D. M. (1991). Systems development in information systems research. *Journal of Management Information Systems*, *7*(3), 89–106.

Osmanoglu, E. (2013). *Identity and Access Management: Business Performance Through Connected Intelligence*. Syngress.

Panetta, K. (2017). Top Trends in the Gartner Hype Cycle for Emerging Technologies. Retrieved November 18, 2017, from https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/

Parviainen, Tihinen, Lormans, & Solingen. (2005). Requirements Engineering: Dealing with the Complexity of Sociotechnical Systems Development. In *Requirements Engineering for Sociotechnical Systems* (pp. 1–20).

Pascual, A., Marchini, K., & Miller, S. (2017). 2017 Identity Fraud: Securing the Connected Life | Javelin. Retrieved September 19, 2017, from https://www.javelinstrategy.com/coverage-area/2017-identity-fraud

Pfitzmann, A., & Kiel, U. L. D. (2008). Pseudonymity , and Identity Management – A

Consolidated Proposal for Terminology. TU Dresden. https://doi.org/10.1017/CBO9781107415324.004

Pilkington, M. (2015). Blockchain Technology: Principles and Applications. *Research Handbook on Digital Transformations*, 1–39. https://doi.org/10.4337/9781784717766.00019

Privacy Valley, & Van Hasselt, E. (2017). Reconciling PSD2 and GDPR. Retrieved October 31, 2017, from https://www.privacyvalley.nl/consent-under-psd2-gdpr/

Reed, D., Law, J., & Hardman, D. (2016). The Technical Foundations of Sovrin. Sovrin Foundation. Retrieved from https://www.sovrin.org/The Technical Foundations of Sovrin.pdf

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public key crypto systems. *Commun. ACM*, *21*(2), 120–126. https://doi.org/10.1145/359340.359342

Rogaway, P., & Shrimpton, T. (2004). Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *International Workshop on Fast Software Encryption* (pp. 371–388). Springer, Berlin. https://doi.org/10.1007/b98177

Ryabitsev, K. (2014). PGP Web of Trust: Core Concepts Behind Trusted Communication. Retrieved August 29, 2017, from https://www.linux.com/learn/pgp-web-trust-core-concepts-behind-trusted-communication

Salmon, B. (2016). GDPR: Data Controller v Data Processor. Retrieved October 31, 2017, from https://www.lexology.com/library/detail.aspx?g=9eabedfb-a61b-48b6-8985-e0728e2ffd8c

Sharmila, P., & Umarani, R. (2011). A walkthrough of Requirement Elicitation Techniques. *International Journal of Engineering Research and Applications (IJERA)*, *1*(4), 1583–1586. Retrieved from https://pdfs.semanticscholar.org/9d89/73d68583fb511f74cdda1bcf38c16e41173e.pdf

Siriwardena, P. (2017a). General Data Protection Regulation (GDPR) for Identity Architects. Retrieved September 8, 2017, from https://medium.facilelogin.com/gdpr-for-identity-architects-1a6423759d30

Siriwardena, P. (2017b). Understanding General Data Protection Regulation (GDPR). Retrieved September 8, 2017, from https://medium.facilelogin.com/understanding-gdpr-9201e1356418

Sporney, M., & Longley, D. (2016). DRAFT: A Self-Sovereign Identity Architecture. ID2020

Design Workshop.

Swan, M. (2015). *Blueprint for a new economy*. *O'Reilly Media, Inc.* https://doi.org/10.1017/CBO9781107415324.004

Swanson, T. (2015). *Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems*.

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, *2*(9).

Tobin, A. (2017). *Sovrin : What Goes on the Ledger ?* Retrieved from https://www.evernym.com/wp-content/uploads/2017/07/What-Goes-On-The-Ledger.pdf

UPort. (2017). GitHub: uPort Project. Retrieved November 16, 2017, from https://github.com/uport-project

W3C. (2017). Verifiable Claims Data Model and Representations. Retrieved November 6, 2017, from https://www.w3.org/TR/verifiable-claims-data-model/#verifiable-claims-model

Whitley, E. A., Gal, U., & Kjaergaard, A. (2014). Who do you think you are? A review of the complex interplay between information systems, identification and identity. *European Journal of Information Systems*, *23*(1), 17–35. https://doi.org/10.1057/ejis.2013.34

Wiese, B. (2013). The 5 Causes of Identity and Access Management Failure. Retrieved November 9, 2017, from https://www.optiv.com/blog/the-5-causes-of-identity-and-access-management-failure

World Economic Forum. (2016). *A Blueprint for Digital Identity*. Retrieved from http://www3.weforum.org/docs/WEF_A_Blueprint_for_Digital_Identity.pdf

Yousuf, M., & Asger, M. (2015). Comparison of Various Requirements Elicitation Techniques. *International Journal of Computer Applications*, *116*(4), 15. https://doi.org/10.6088/ijacit.12.14005

Appendix A. General Data Protection Regulation (GDPR)

Relevant implications of the GDPR – which will be effective from May 2018 – are like following (European Parliament, 2016; Siriwardena, 2017b):

1. The GDPR concerns itself with natural persons. That is, an individual human being.
2. Personal data is any information regarding a natural person, not just personally identifiable data. This includes IP addresses and device identifiers.
3. GDPR applies to any natural person in the EU.
4. A data controller will ensure that their data processors will handle according to GDPR.
5. Processors are restricted from engaging another processor without prior specific or general written authorization from the controller.
6. GDPR defines consent as, any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her.
7. Processing shall be lawful only if and to the extent that at least one of the following applies:
   7.1. The data subject has given consent to the processing of his or her personal data for one or more specific purposes.
   7.2. Processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract.
   7.3. Processing is necessary for compliance with a legal obligation to which the controller is subject.
   7.4. Processing is necessary in order to protect the vital interests of the data subject or of another natural person.
   7.5. Processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller.
   7.6. Processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child.
8. Where the consent has not being gained for the specific purpose in question, the controller must address additional conditions to determine the transparency and fairness of processing.

8.1. Any link between the purposes for which the personal data have been collected and the purposes of the intended further processing.

8.2. The context in which the personal data have been collected, in particular regarding the relationship between data subjects and the controller.

8.3. The nature of the personal data, in particular whether special categories of personal data are processed, or whether personal data related to criminal convictions and offenses are processed.

8.4. The possible consequences of the intended further processing for data subjects.

8.5. The existence of appropriate safeguards, which may include encryption or pseudonymization.

9. GDPR stipulates that the controller must provide data subjects access to their personal data, the purpose of processing their data, the categories of data being processed, the third parties or categories of third parties that will receive their data and the period of time which the data will be stored

10. The controller shall, at the time when personal data are obtained, provide the user with the following further information necessary to ensure fair and transparent processing:

10.1. The period for which the personal data will be stored, or if that is not possible, the criteria used to determine that period.

10.2. The existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object to processing as well as the right to data portability.

10.3. Where the processing is based on, the existence of the right to withdraw consent at any time, without affecting the lawfulness of processing based on consent before its withdrawal.

10.4. The right to lodge a complaint with a supervisory authority.

10.5. Whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data.

10.6. The existence of automated decision-making, including profiling, and, at least in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject.

11. The data subject has the right to receive the personal data concerning him or her, which he or she has provided to a controller, in a structured, commonly used and machine-readable format and have the right to transmit those data to another controller without hindrance from the controller.
12. Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes. The business must explicitly specify the usage and purpose of the personal data collection—and gets the user's consent.
13. GDPR grants full rights to individuals to request deletion or removal of their personal data.
14. It's a good practice to record any additional data that you collect, apart from direct user attributes—against a pseudonym.
15. Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed. Also personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed. No data should be collected for future anticipated usage. For example, if the registration with your business is only for people elder than 21 years—you should not record user's birthdate, but just the claim that your customer is elder than 21.
16. The right to restriction of data processing effectively allows data subjects, under certain specific circumstances, to prevent controllers from conducting specific processing of their data. It means that, although the controller can store the personal data, it cannot process the data unless the individual gives their consent to lift the restriction or the processing is required for the establishment of legal claims
17. Personal data shall be processed in a manner that ensures appropriate security of the personal data. Personal data must be classified as confidential even within the organization.

Appendix B. Extended ten principles of Self-Sovereign Identity

Christopher Allen's ten principles of self-sovereign identity, as he originally described them, are listed below (Allen, 2016):

1. **Existence**: users must have an independent existence. Any self-sovereign identity is ultimately based on the ineffable "I" that's at the heart of identity. It can never exist wholly in digital form. This must be the kernel of self that is upheld and supported. A self-sovereign identity simply makes public and accessible some limited aspects of the "I" that already exists.
2. **Control**: users must control their identities. Subject to well-understood and secure algorithms that ensure the continued validity of an identity and its claims, the user is the ultimate authority on their identity. They should always be able to refer to it, update it, or even hide it. They must be able to choose celebrity or privacy as they prefer. This doesn't mean that a user controls all of the claims on their identity: other users may make claims about a user, but they should not be central to the identity itself.
3. **Access**: users must have access to their own data. A user must always be able to easily retrieve all the claims and other data within his identity. There must be no hidden data and no gatekeepers. This does not mean that a user can necessarily modify all the claims associated with his identity, but it does mean they should be aware of them. It also does not mean that users have equal access to others' data, only to their own.
4. **Transparency**. Systems and algorithms must be transparent. The systems used to administer and operate a network of identities must be open, both in how they function and in how they are managed and updated. The algorithms should be free, open-source, well-known, and as independent as possible of any particular architecture; anyone should be able to examine how they work.
5. **Persistence**. Identities must be long-lived. Preferably, identities should last forever, or at least for as long as the user wishes. Though private keys might need to be rotated and data might need to be changed, the identity remains. In the fast-moving world of the Internet, this goal may not be entirely reasonable, so at the least identities should last until they've been outdated by newer identity systems. This must not contradict a "right to be forgotten"; a user should be able to dispose of an identity if he wishes and claims should be modified or removed as appropriate over time. To do this requires a firm separation between an identity and its claims: they can't be tied forever.

6. **Portability**. Information and services about identity must be transportable. Identities must not be held by a singular third-party entity, even if it's a trusted entity that is expected to work in the best interest of the user. The problem is that entities can disappear — and on the Internet, most eventually do. Regimes may change, users may move to different jurisdictions. Transportable identities ensure that the user remains in control of his identity no matter what, and can also improve an identity's persistence over time.
7. **Interoperability**. Identities should be as widely usable as possible. Identities are of little value if they only work in limited niches. The goal of a 21st-century digital identity system is to make identity information widely available, crossing international boundaries to create global identities, without losing user control. Thanks to persistence and autonomy these widely available identities can then become continually available.
8. **Consent**. Users must agree to the use of their identity. Any identity system is built around sharing that identity and its claims, and an interoperable system increases the amount of sharing that occurs. However, sharing of data must only occur with the consent of the user. Though other users such as an employer, a credit bureau, or a friend might present claims, the user must still offer consent for them to become valid. Note that this consent might not be interactive, but it must still be deliberate and well-understood.
9. **Minimization**. Disclosure of claims must be minimized. When data is disclosed, that ja disclosure should involve the minimum amount of data necessary to accomplish the task at hand. For example, if only a minimum age is called for, then the exact age should not be disclosed, and if only an age is requested, then the more precise date of birth should not be disclosed. This principle can be supported with selective disclosure, range proofs, and other zero-knowledge techniques, but non-correlativity is still a very hard (perhaps impossible) task; the best we can do is to use minimization to support privacy as best as possible.
10. **Protection**. The rights of users must be protected. When there is a conflict between the needs of the identity network and the rights of individual users, then the network should err on the side of preserving the freedoms and rights of the individuals over the needs of the network. To ensure this, identity authentication must occur through independent algorithms that are censorship-resistant and force-resilient and that are run in a decentralized manner.

Appendix C. Example of a verifiable claim

Below is the JSON-LD code of an example verifiable claim (W3C, 2017).

```
{
  "@context": [
    "https://w3id.org/identity/v1",
    "https://w3id.org/security/v1"
  ],
  "id": "http://example.gov/credentials/3732",
  "type": ["Credential", "ProofOfAgeCredential"],
  "issuer": "https://dmv.example.gov",
  "issued": "2010-01-01",
  "claim": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "ageOver": 21
  },
  "signature": {
    "type": "LinkedDataSignature2015",
    "created": "2016-06-18T21:10:38Z",
    "creator": "https://example.com/jdoe/keys/1",
    "domain": "json-ld.org",
    "nonce": "6165d7e8",
    "signatureValue":
"g4j9UrpHM4/uu32NlTw0HDaSaYF2sykskfuByD7UbuqEcJIKa+IoLJLrLjqDnMz0adwpBC
HWaqqpnd47r0NKZbnJarGYrBFcRTwPQSeqGwac8E2SqjylTBbSGwKZkprEXTywyV7gILl
C8a+naA7lBRi4y29FtcUJBTFQq4R5XzI="
  }
}
```