Hasso-Plattner-Institut
Universität Potsdam

Towards Practical and Trust-Enhancing Attribute Aggregation for Self-Sovereign Identity

Dissertation

zur Erlangung des akademischen Grades des Doktors
der Naturwissenschaften (Dr. rer. nat.) in der Wissenschaftsdisziplin
IT-Systems Engineering am Fachgebiet Internet-Technologien und Systeme

eingereicht an der
Digital Engineering Fakultät
der Universität Potsdam

von
Andreas Grüner
Potsdam, October 2022

To my parents

First Supervisor
Prof. Dr. Christoph Meinel, Hasso Plattner Institute

Second Supervisor
Prof. Dr. Andreas Polze, Hasso Plattner Institute

Dissertation reviewers
Prof. Dr. Christoph Meinel, Hasso Plattner Institute
Prof. Dr. Wolfgang Hommel, Universität der Bundeswehr München
Prof. Dr. Axel Küpper, Technische Universität Berlin

Examination committee
Prof. Dr. Anja Lehmann (Chair), Hasso Plattner Institute
Prof. Dr. Christian Dörr, Hasso Plattner Institute
Prof. Dr. Andreas Polze, Hasso Plattner Institute

Disputation
24.10.2022

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-43571
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-435713

Abstract

Identity management is at the forefront of applications' security posture. It separates the unauthorised user from the legitimate individual. Identity management models have evolved from the isolated to the centralised paradigm and identity federations. Within this advancement, the identity provider emerged as a trusted third party that holds a powerful position. Allen postulated the novel self-sovereign identity paradigm to establish a new balance. Thus, extensive research is required to comprehend its virtues and limitations. Analysing the new paradigm, initially, we investigate the blockchain-based self-sovereign identity concept structurally. Moreover, we examine trust requirements in this context by reference to patterns. These shapes comprise major entities linked by a decentralised identity provider. By comparison to the traditional models, we conclude that trust in credential management and authentication is removed. Trust-enhancing attribute aggregation based on multiple attribute providers provokes a further trust shift. Subsequently, we formalise attribute assurance trust modelling by a meta-framework. It encompasses the attestation and trust network as well as the trust decision process, including the trust function, as central components. A secure attribute assurance trust model depends on the security of the trust function. The trust function should consider high trust values and several attribute authorities. Furthermore, we evaluate classification, conceptual study, practical analysis and simulation as assessment strategies of trust models. For realising trust-enhancing attribute aggregation, we propose a probabilistic approach. The method exerts the principle characteristics of correctness and validity. These values are combined for one provider and subsequently for multiple issuers. We embed this trust function in a model within the self-sovereign identity ecosystem. To practically apply the trust function and solve several challenges for the service provider that arise from adopting self-sovereign identity solutions, we conceptualise and implement an identity broker. The mediator applies a component-based architecture to abstract from a single solution. Standard identity and access management protocols build the interface for applications. We can conclude that the broker's usage at the side of the service provider does not undermine self-sovereign principles, but fosters the advancement of the ecosystem. The identity broker is applied to sample web applications with distinct attribute requirements to showcase usefulness for authentication and attribute-based access control within a case study.

Zusammenfassung

Das Identitätsmanagement ist Kernbestandteil der Sicherheitsfunktionen von Applikationen. Es unterscheidet berechtigte Benutzung von illegitimer Verwendung. Die Modelle des Identitätsmanagements haben sich vom isolierten zum zentralisierten Paradigma und darüber hinaus zu Identitätsverbünden weiterentwickelt. Im Rahmen dieser Evolution ist der Identitätsanbieter zu einer mächtigen vertrauenswürdigen dritten Partei aufgestiegen. Zur Etablierung eines bis jetzt noch unvorstellbaren Machtgleichgewichts wurde der Grundgedanke der selbstbestimmten Identität proklamiert. Eine tiefgehende Analyse des neuen Konzepts unterstützt auf essentielle Weise das generelle Verständnis der Vorzüge und Defizite. Bei der Analyse des Modells untersuchen wir zu Beginn strukturelle Komponenten des selbstbestimmten Identitätsmanagements basierend auf der Blockchain Technologie. Anschließend erforschen wir Vertrauensanforderungen in diesem Kontext anhand von Mustern. Diese schematischen Darstellungen illustrieren das Verhältnis der Hauptakteure im Verbund mit einem dezentralisierten Identitätsanbieter. Im Vergleich zu den traditionellen Paradigmen, können wir festellen, dass kein Vertrauen mehr in das Verwalten von Anmeldeinformationen und der korrekten Authentifizierung benötigt wird. Zusätzlich bewirkt die Verwendung von vertrauensfördernder Attributaggregation eine weitere Transformation der Vertrauenssituation. Darauffolgend formalisieren wir die Darstellung von Vertrauensmodellen in Attribute Assurance mit Hilfe eines Meta-Frameworks. Als zentrale Komponenten sind das Attestierungs- und Vertrauensnetzwerk sowie der Vertrauensentscheidungsprozess, einschließlich der Vertrauensfunktion, enthalten. Ein sicheres Vertrauensmodell beruht auf der Sicherheit der Vertrauensfunktion. Hohe Vertrauenswerte sowie mehrere Attributaussteller sollten dafür berücksichtigt werden. Des Weiteren evaluieren wir Klassifikation, die konzeptionelle und praktische Analyse sowie die Simulation als Untersuchungsansätze für Vertrauensmodelle. Für die Umsetzung der vertrauensfördernden Attributaggregation schlagen wir einen wahrscheinlichkeitstheoretischen Ansatz vor. Die entwickelte Methode basiert auf den primären Charakteristiken der Korrektheit und Gültigkeit von Attributen. Diese Indikatoren werden für einen und anschließend für mehrere Merkmalsanbieter kombiniert. Zusätzlich betten wir die daraus entstehende Vertrauensfunktion in ein vollständiges Modell auf Basis des Ökosystem von selbstbestimmten Identitäten ein. Für die praktische Anwendung der Vertrauensfunktion

und die Überwindung mehrerer Herausforderungen für den Dienstanbieter, bei der Einführung selbstbestimmter Identitätslösungen, konzipieren und implementieren wir einen Identitätsbroker. Dieser Vermittler besteht aus einer komponentenbasierten Architektur, um von einer dedizierten selbstbestimmten Identitätslösung zu abstrahieren. Zusätzlich bilden etablierte Identitäts- und Zugriffsverwaltungsprotokolle die Schnittstelle zu herkömmlichen Anwendungen. Der Einsatz des Brokers auf der Seite des Dienstanbieters unterminiert nicht die Grundsätze der selbstbestimmten Identität. Im Gegensatz wird die Weiterentwicklung des entsprechenden Ökosystems gefördert. Innerhalb einer Fallstudie wird die Verwendung des Identitätsbrokers bei Anwendungen mit unterschiedlichen Anforderungen an Benutzerattribute betrachtet, um die Nützlichkeit bei der Authentifizierung und Attributbasierten Zugriffskontrolle zu demonstrieren.

Acknowledgement

Pursuing a doctoral degree in computer science is an extensive journey. Along with this journey, the student needs to meet various challenges. In the beginning, one must identify a research area that meets their passion. Afterwards, determining open problems and their solutions require thorough investigation. Condensed research results are written as a paper and submitted for publishing to a conference or journal. Between the submittal and the notification time, there is the hope for acceptance. An accepted paper lead to a feeling of joy, whereas rejection and the respective comments demand further analysis. As a final part, the student writes and defends the thesis.

None of the challenges usually come with an easy short-cut. However, these hurdles can be mastered with an undeniable dedication. During this journey, remarkable support from and vivid discussions with colleagues, friends and family are an essential cornerstone for progress. Therefore, I would like to thank everybody for constant backing and encouragement. In particular, I would like to thank Professor Dr. Christoph Meinel for my thesis' supervision and his permanent support.

Furthermore, the productive discussions within the Secure Identity Lab team are highly appreciated. Last but not least, I would like to thank my family and friends, especially for the tolerance that I could only devote scarce time to them.

Thank you, all!

Contents

List of Figures

List of Tables

List of Definitions

List of Equations

1 Introduction

1.1 Current Situation

Social networks, instant messaging, and web stores are only a few examples of online services that permeate everyday life. Nonetheless, the digitisation of businesses and the private living space is still in its infancy. The COVID19 pandemia [1] suddenly pushed the boundaries to significantly increase work at home and video conferencing, and further online services. Despite this singular event, digitisation is seen as a panacea and is continuously driven in all domains [2]. In the same breath, the security of these online services demands immense attention to defend against adversaries. Common security objectives comprise confidentiality, integrity, and availability [3]. Confidentiality [4] refers to information disclosure to legitimate parties. The objective of integrity [4] describes that data can not be tampered by unauthorised persons. Availability [4] guarantees that resources are serviceable for authorised users.

In general, these security objectives differentiate between legitimate and unauthorised persons. Legitimate users are allowed to access an application, execute functions, read data and communicate. Unauthorised parties are not permitted to start an application, use a function, read or write data. Legitimacy might be applied on the service level or used in a very fine-grained manner on function or data dimension. Thus, identity and access management techniques are a foundational cornerstone and at the forefront of every service's security to enable this differentiation.

Notwithstanding the importance of identity and access management, the Internet was created without an identity layer [5], leading to numerous challenges. Already in 1993, P. Steiner published a caricature to state that *"On the Internet, nobody knows you're a dog"* [6] and therefore highlighted the difficulty to identify an individual over a network. Steiner's caricature depicts the situation in a very humorous way. However, using another person's identity, referred to as identity fraud or theft, implies serious consequences for an individual and is up to date as never before. Thereby, imposters misuse captured personal information, e.g. credit card numbers, to achieve illegitimate benefits and might take over accounts due to weak or leaked passwords. An incredible number of 550 million passwords were leaked on the dark net between 2017 and 2019 [7].

The multi-faceted vulnerability of identity traces back to its origin and can be best illustrated by the development of their models. In the beginning, each service encompassed a dedicated identity provider. Therefore, users are required to register at each service independently. During the enrolment, a service-specific account and a corresponding credential are issued to the user. With an increasing number of websites, the users got overwhelmed by the number of registration processes. Aside from the repeated effort, a myriad of credentials must be securely stored and remembered. According to a study [8], an average user has about 100 passwords. Further identity management patterns have been developed to alleviate this situation. The identity provider became a dedicated component for several applications within a company gradually. Furthermore, it transformed into an independent party beyond the boundaries of a single organisation. As a result, one registration at a specific identity provider and one credential enable access to several services. Social networks, e.g. Facebook, became an identity provider due to their large user base where customers own an account [9]. This capability is leveraged to provide identity service, in the form of a simple login, for other online services. With this progress, the identity provider became a trusted third party towards users and service providers.

The trust in the identity provider relates to manifold themes. First, users expect secure storage of credential verification information to support the authentication process. Furthermore, service providers require thorough verification of the user's attributes for service provisioning. Moreover, the adherence to privacy principles is also underpinned by the General Data Protection Regulation [10]. Last but not least, the identity provider is expected to adhere to contractual obligations and does not exert unwanted control over its identity data. Besides the trust of the service provider and user, a centralised identity provider is also a single point of failure and a lucrative target for attackers.

To turn over a new leaf in identity management, the recently proposed self-sovereign identity management paradigm focuses explicitly on the user and strives to bring it back into control of its identity and data. The sovereignty about the identity should lie with the user that is represented by it. As a principle, they should be self-sovereign. The control should not reside with the identity provider that holds a powerful position within the previous non-self-sovereign paradigms. The term self-sovereign identity management was shaped by Allen [11]. He did not determine this term by definition but postulated ten foundational principles. These principles encompass, for instance, control, data minimalisation, access, and user consent for data disclosure. The complete concept targets the user and its protection against more powerful entities. However, a technological implementation of the concept is a challenging endeavour due to the centralised nature of applications that are controlled by their hosting entities.

A comparable dilemma prevails in other domains than identity management. A central authority exists that is a trusted third party for other entities. Additionally, there is a disparity of power between the central authority and further participants. In the financial domain, banks are trusted third parties for customers. On one side, federal banks regulate money issuance and distribution. Besides that, commercial banks control access to bank accounts, bank transfers, and handling of cash. Therefore, courts and the government target with regulations financial institutions to control money flows or to freeze bank accounts [12]. Research activities do not only concentrate on digital cash schemes but also on decentralised electronic cash concepts that do not rely on a central trusted third party. To realise an entirely decentralised approach, solutions for the double-spending problem [13] is a research area. The double-spending problem describes that a digital coin can be copied and spent several times. The traditional governance of a central authority is a simple approach to solve this challenge.

The proposal of Bitcoin [14] realised a decentralised digital cash scheme that does not require a trusted third party for solving the double-spend problem. A successive sequence of blocks agreed and maintained by a network of equitable peers reflects the core of the blockchain system. Within the blocks, messages that represent cash transfers are persisted. The advancement of Bitcoin led to the development of general decentralised execution platforms. Programs are distributed to and executed by every peer. Afterwards, the result of the computation is broadcasted. A consensus algorithm achieves agreement between the peers. Therefore, the blockchain became also a viable implementation option for the self-sovereign identity paradigm. A decentralised identity provider implemented on a blockchain does not represent a trusted third party anymore [15]. Thus, there is no external control of the digital identity. The implementation of the decentralised identity provider is open and transparent to all peers.

1.2 Motivation

Having outlined the significance of identity management for any online service's security, the development of a new paradigm requires particular research to fully understand implications, options and chances for all participating actors. The novel self-sovereign identity paradigm breaks with principles of traditional identity management concepts. The schema focuses in particular on the user and its rights. Additionally, the usage of blockchain technology as a suitable implementation option has not been applied before and extends the existing set of approaches.

Traditional identity management models have been well studied from different perspectives. Trust is a principal component in relationships [16]. Therefore, researchers analysed trust requirements between the user, the service provider and

the identity provider in the isolated, centralised and federated identity management scheme [17] [18]. By remediating the identity provider as a trusted third party with the support of blockchain technology, trust requirements significantly change. This transformation raises open questions to weather the omission of the identity provider grants benefits or implies drawbacks compared to the existing models in terms of trust. Moreover, a substantial matter is the potential replacement of the identity provider by another central authority. This circumstance may let seem the reduction of the identity provider to be questionable. Analysis and insights are demanded to understand the consequences of this development fully.

Furthermore, the self-sovereign identity paradigm and the applied blockchain-based implementation effectively decouples the attributes from a digital identity's identifier [15]. In the traditional models, the identity provider issues both elements of an identity jointly. In identity federations, the identifiers of different providers are linked for translation together [19]. The identifier of a blockchain-based self-sovereign identity is registered on a decentralised application. Various attribute providers may issue characteristics for this identifier of the user. A property provider and consumer ecosystem is constituted. Therefore, attributes can be easily aggregated from distinct providers to be used jointly at a service provider. Users and service provider might trust each attribute provider differently. Reasons range from distinct political opinions, negative reputation based on adverse press articles, errors in attribute verification processes or already happened attacks. Users may prefer varying attribute providers, and service providers might have other favourites. In traditional identity management models, the flexibility to choose a specific identity provider is minimal. Suppose that a user intends to consume a particular service and the respective provider offers only one identity provider. In that case, the user might register at this identity provider or does not consume the service at all. With the ability to select a specific attribute provider or sets of attribute providers in the self-sovereign identity paradigm, new trust models can be applied. Each service provider nominates its trustworthy providers or combinations of them. The user also decides on its attribute providers. If the combinations matches, a successful interaction can commence. This flexibility enables a novel way of trust-enhancing attribute aggregation to establish a trust model between the classical web of trust and chain of trust. Research on the theoretical foundation and the practical application is required to make this approach usable.

Moreover, the user and the service provider are equally needed to let the identity ecosystem flourish. If the user strives for a particular identity provider while no service provider accepts it, the identity provider is of no value. The same applies to the self-sovereign identity paradigm that strongly focuses on the user. Nonetheless, the demands of the service provider are neglected to a certain ex-

tent. A plentitude of blockchain-based self-sovereign identity solutions exists [20]. These solutions can be integrated into services via dedicated application libraries. Established protocols are not holistically applied, respectively, new standards for self-sovereign identity requires development. Additionally, a newly created self-sovereign identity does not comprise any attributes. The user can self-issue characteristics. These properties are only trusted in a limited manner by the service provider because no external verification was conducted. Attribute providers are one source for properties, though service providers may also require the capability to issue attributes themselves. For instance, a service provider owns authoritative data about memberships that can be issued as attributes to an identity. To sustainably evolve the self-sovereign identity ecosystem, flexible integration solutions for service providers require research and development.

1.3 Contributions

The content of this thesis intends to drive the understanding of the self-sovereign identity pattern and to enable service providers to practically use it for their customers. We focus particularly on trust requirements and trust models regarding attribute assurance under the light of the new characteristics of self-sovereign identity. Especially, we elaborate on trust-enhancing attribute aggregating to reduce the dependency towards a specific identity respectively attribute provider. Moreover, the practical implementation leads to the creation of an identity broker to directly apply the results of the present research. The following subsections particularise the addressed research questions and the contributions made by this thesis.

1.3.1 Systematisation of Self-Sovereign Identity

The concept of self-sovereign identity is proposed by reference to ten principles. These principles reflect the objectives or advantages of a solution for a user. However, this paradigm might be implemented in different ways by using various patterns or technologies. We provide an overview of essential components for self-sovereign identity and their realisation. Moreover, we outline the interaction between the concept and blockchain technology. We study in particular the decentralisation of components to remediate trusted third parties and to avoid the reintroduction. Generally, central authorities undermine the objective of self-sovereignty. The outline brings transparency to the composition of self-sovereign identity. Furthermore, we depict the arrangement of the new model in alignment with the existing patterns. Chapter 2.6 delineates this contribution.

1.3.2 Trust Requirements for Self-Sovereign Identity

Trust is an important component of relationships between persons and also organisations. In identity management, trustful associations exist between the identity provider, the service provider and the user. For instance, the trust may influence the decision of a user to choose a specific identity provider. The same can apply for a service provider. The various requirements that lead to trust between the entities are in particular important. Within the blockchain-based self-sovereign identity model, the identity provider is not a trusted third party anymore. Therefore, trust requirements change significantly. We define patterns between all actors to analyse trust requirements in the self-sovereign identity concept. Subsequently, we compare the trust situation with the traditional identity management models and the evolvement between each paradigm. Our contribution enables a clear view of trust benefits in the self-sovereign identity paradigm. We scrutinise this research area in Chapter 3.

1.3.3 Formalisation and Assessment Strategies of Trust Models in Attribute Assurance

Originating from changed trust requirements in identity management models, attribute assurance is a significant research area for trust models. Attribute assurance ensures that properties of a digital identity reflect reality. The service provider tremendously relies on correct attributes that are rigorously verified by the identity or attribute provider. The user indirectly depends on the properties for legitimate service consumption. Over time, manifold trust models have been proposed that differently compose trust in attributes. Authors describe these trust models in different modes and manners. A formalised notation is essential to commonly specify a trust model for creating new concepts or advancing existing models.

Moreover, a general depiction enables comparative studies. The formal model comprises important factors for attribute assurance to show structural similarities, respectively differences. Furthermore, the notation builds the foundation for practically implementing trust models in an identity broker to foster a widespread usage. We contribute a general meta-framework to study trust models in attribute assurance. In addition to that, we present an overview of the assessment approaches classification, conceptual and practical analysis, and simulation. By reference to these methodologies, we conduct an evaluation towards the applicability and expressiveness on the different trust model components. This examination shows the benefits and limitations of each approach. We study this research domain in Chapter 4.

1.3.4 Foundations of Trust-Enhancing Attribute Aggregation

Traditional attribute aggregation targets the combination of distinct attributes from different attribute providers. In contrast, trust-enhancing attribute aggregation describes the usage of different attribute providers to increase trust in a certain attribute. Thus, the same feature is issued by various providers. The combination of these attestations to retrieve an increased trust value for the attribute requires a sound theoretical concept. Our contribution embraces a probabilistic aggregation concept based on validity and correctness of an attribute that is issued by a certain provider. Moreover, the trust values of several attribute providers are combined to retrieve an overall trust score. If a specified threshold is exceeded, the service provider trusts the attribute. This research domain is addressed in Chapter 5.

1.3.5 Attribute Trust-Enhancing Identity Broker

The self-sovereign identity paradigm focuses explicitly on the user while neglecting in the first place the requirements of the service provider. In particular, projects and companies develop a myriad of self-sovereign identity solutions that provide a proprietary integration library for applications. Service providers are not able to integrate into all approaches due to the enormous effort. Besides this, organisations are used to apply standard identity and access management protocols for a streamlined integration. Furthermore, service providers demand facilities to issue attributes for self-sovereign identities. Upon creation, a self-sovereign identity has no features. The user might add non-verified properties that the service provider does not trust. However, organisations own authoritative data that expresses entitlements for their services. These privileges can be reflected as attributes of an identity if respective issuance facilities are available. Moreover, the service provider needs a practical approach to use the potential of trust-enhancing attribute aggregation and to offer this option for its users.

Our contribution is an Attribute Trust-Enhancing Identity Broker to address these challenges for the service provider. The identity broker implements an architecture clustered into components to facilitate self-sovereign identity usage for authentication and attribute-based authorisation. Attribute issuance and verification with specific trust models are reflected by further components of the identity broker. Chapter 6 presents this research area. Chapter 7 outlines the results from the practical application of the identity broker as a case study.

1.4 Publications

During the research work for this thesis at the Hasso Plattner Institute, the following articles have been published in international journals and conferences. These publications outline dedicated contributions of the research work.

1.4.1 First Author Publications

- A. Grüner, A. Mühle, C. Meinel: *Analyzing Interoperability and Portability Concepts for Self-Sovereign Identity.* in Proceedings of the 20th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Shenyang, China: IEEE Computer Society, 2021 [21]

- A. Grüner, A. Mühle, C. Meinel: *ATIB: Attribute Trust-Enhancing Identity Broker. Design and Evaluation of an Architecture for Brokered Self-Sovereign Identity Integration and Trust-Enhancing Attribute Aggregation for Service Provider.* IEEE Access, vol. 9, 2021, pp. 138553-138570. [22]

- A. Grüner and C. Meinel: *On the Structure and Assessment of Trust Models in Attribute Assurance.* in Proceedings of the 35th International Conference on Advanced Information Networking and Applications (AINA). Advances in Intelligent Systems and Computing. Cham, Germany: Springer, 2021, pp. 447-458. [23]

- A. Grüner, A. Mühle and C. Meinel, *An Integration Architecture to Enable Service Providers for Self-sovereign Identity.* in Proceedings of the 18th IEEE International Symposium on Network Computing and Applications (NCA). Cambridge, Massachusetts, USA: IEEE Computer Society, 2019, pp. 1-5. [24]

- A. Grüner, A. Mühle, T. Gayvoronskaya and C. Meinel, *A Comparative Analysis of Trust Requirements in Decentralized Identity Management.* in Proceedings of the 33rd International Conference on Advanced Information Networking and Applications (AINA). Advances in Intelligent Systems and Computing, vol. 926. Cham, Germany: Springer, 2019, pp. 200-213. [15]

- A. Grüner, A. Mühle and C. Meinel, *Using Probabilistic Attribute Aggregation for Increasing Trust in Attribute Assurance.* in Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI). Xiamen, China: IEEE Computer Society, 2019, pp. 633-640. [25]

- A. Grüner, A. Mühle, M. Meinig and C. Meinel, *A Taxonomy of Trust Models for Attribute Assurance in Identity Management.* in Proceedings of the Workshops of the International 34th Conference on Advanced Information Networking and Applications (WAINA). Web, Artificial Intelligence and Network Applications, vol. 1150. Cham, Germany: Springer, 2020, pp. 65-76. [26]

- A. Grüner, A. Mühle, T. Gayvoronskaya and C. Meinel, *Towards a Blockchain-based Identity Provider.* in Proceedings of the 12th International Conference on Emerging Security Information, Systems and Technologies (Secureware). Wilmington, Delaware, USA: IARIA, 2018, pp. 73-78. [27]

- A. Grüner, A. Mühle, T. Gayvoronskaya and C. Meinel, *A Quantifiable Trust Model for Blockchain-Based Identity Management.* in Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). Los Alamitos, California, USA: IEEE Computer Society, 2018, pp. 1475-1482. [28]

1.4.2 Co-Author Publications

- A. Mühle, A. Grüner, C. Meinel, *Characterising Proxy Usage in the Bitcoin Peer-to-Peer Network.* in Proceedings of the 2021 International Conference on Distributed Computing and Networking (ICDCN). New York, New York, USA: ACM, 2021, pp. 176-185. [29]

- A. Mühle, A. Grüner, T. Gayvoronskaya and C. Meinel, *A Survey on Essential Components of a Self-Sovereign Identity.* Computer Science Review, vol. 30, pp. 80-86, 2018. [30]

1.5 Thesis Outline

This thesis is organised in eight chapters. Chapter 1 provides the introduction to the field in general, and this thesis in particular. The remainder of the thesis is organised as follows.

Chapter 2 presents the fundamentals of identity management in the context of the self-sovereign identity paradigm. We define identity management and describe types, paradigms and the involved actors. Subsequently, we elaborate on trust in relation to identity management and provide an overview of attribute assurance. Afterwards, the basis of blockchain technology and the ten principles of

self-sovereign identity are introduced to lay the foundation of the corresponding paradigm. In the following sections, we summarise characteristics of blockchain-based self-sovereign identity and explain related components for construction.

After establishing a common understanding of self-sovereign identity and other background topics, trust requirements in the context of self-sovereign identity are analysed thoroughly in Chapter 3. We start with a description of various trust domains and derive single requirements. Subsequently, we elaborate on trust patterns between the user, attribute provider and service provider that a decentralised identity provider connects. Finally, we compare the trust situation of the blockchain-based self-sovereign identity model with the traditional paradigms.

Subsequent to the analysis of trust requirements, we investigate the structure of trust models in attribute assurance and evaluate their assessment strategies in Chapter 4. Hereby, we outline trust model characteristics, security objectives and potential attacks. Furthermore, we formalise the setting as a meta-framework to depict them. In addition to that, we present and evaluate the assessment strategies classification, conceptual and practical analysis, and simulation. We show specifically the components of a trust model that can be evaluated with each strategy.

In the following Chapter 5, we describe a probabilistic attribute aggregation strategy to combine the issued attributes from different attribute providers to increase the overall trust in the attribute value. We represent the authenticity of an attribute as a probability of correctness and validity. Furthermore, we aggregate this information for one provider and illustrate the accumulation of several providers. Using the joint probabilities as a trust function, we describe a holistic trust model and evaluate it as a practical component of an identity broker.

Ensuing, we delineate in Chapter 6, the practical application of the theoretical foundations we have laid in the previous chapters. We describe an Attribute Trust-Enhancing Identity Broker to overcome various challenges that have service providers when adopting self-sovereign identity solutions. Initially, we outline the service provider challenges and derive requirements for the identity broker. Subsequently, we illustrate the architecture into components and their implementation. Finally, we conclude the chapter with a security analysis of the broker and sample authentication flows.

Succeeding the concept of the identity broker, we describe a case study for the use of self-sovereign identity in sample applications. We outline attribute requirements and its realisation when applying the broker to the identity broker's user interface, tele-TASK and OpenHPI.

With Chapter 8, we complete the thesis by summarising the presented contributions and a conclusion on the impact. Furthermore, we outline future research directions in this domain.

2 Fundamentals of Self-Sovereign Identity

This chapter presents background on identity management, trust and attribute assurance to built the foundations for the subsequent topics. Moreover, we outline essential principles and components of self-sovereign identity, blockchain technology and decentralisation characteristics for blockchain-based self-sovereign identity implementations.

2.1 Identity Management

The term identity management and identity have been already used within the introductory chapter without a corresponding definition. We rectify the denotation in this section. Besides that, we use digital identity management and identity management interchangeable. Windley [31] defines identity management as *"creating, managing, using and eventually destroying records"*. The term records relates to identity data. Lewis [32] denotes identity management as *"business processes, and a supporting infrastructure for the creation, maintenance, and use of digital identities"*. Furthermore, a profound characterisation is provided by the Y.2720 identity management framework [33] of the International Telecommunication Union. The framework reflects on identity management as *"functions and capabilities"* [33] to ensure the correctness of identity information and the related entity. In conclusion, identity management involves all activities and implemented processes to handle digital identities.

Thus, at the core of identity management is the digital identity or unpretentiously called identity. A digital identity is a collection of information that characterises a physical entity [34]. The properties of an identity are called attributes. A special attribute is the identifier that is used to reference an identity. Within a domain, all identifiers must be unique to address unambiguously a digital identity. The domain is the scope of consistently applied identity management that is usually under the control of a single authority. The digital identity enables participation in digital processes for the reflected physical object. Thus, only the correct entity should be able to use its specific identity.

There is a binding between the identity and the actual user. This binding

is verified during the authentication process. In case username and password authentication is applied, the password enables the user to demonstrate legitimate access. The password must be kept confidential to avoid any usage by other parties.

The identity management ecosystem differentiates various actors. The main entities are the identity provider, the service provider and the user [35]. Additionally, the attribute provider is differentiated.

- **Service Provider:** The service provider offers an online service that requires identity management. A company is usually a service provider that intends to recognise users.
- **User:** The user is an individual that interacts with a provided online service. Furthermore, it reflects the subject of a dedicated digital identity. The user controls its identity by a credential.
- **Identity Provider:** The identity provider implements identity management functions and offers it for online services. Typical identity management processes encompass user registration, credential management, authentication, authorisation and attribute management including verification procedures.
- **Attribute Provider:** The attribute provider is solely responsible for the verification and issuance of attributes of a digital identity. This actor's functions are a subset of the identity provider.

Identity management models can be two-dimensionally clustered according to the structure or the entity-focus of the model. The structural categories are based on the interaction between the described actors. The entity-focus classes reflect the concentration of the processes towards a specific party. Fig. 2.1 shows an overview of the two viewpoints of the models. Structural identity management patterns are comprised of isolated, centralised, decentralised and federated schemes [34].

In isolated identity management, each online service uses its own internal identity provider component. Therefore, separate identity management processes exist for each application [36]. This pattern was initially applied in the identity management domain. As a disadvantage, every service requires individual registration for a user. With the growing number of online services, the effort and repetition for this process raised significantly. Moreover, the user had an obligation to protect an increasing number of credentials for their identities.

The centralised identity management model is a further development of the isolated pattern. Aligned with this model, a central identity provider offers identity management for several services [36]. These services might exist within one organisation or span multiple organisations. As an advantage, a user solely registers once at the centralised identity provider and can authenticate at all integrated

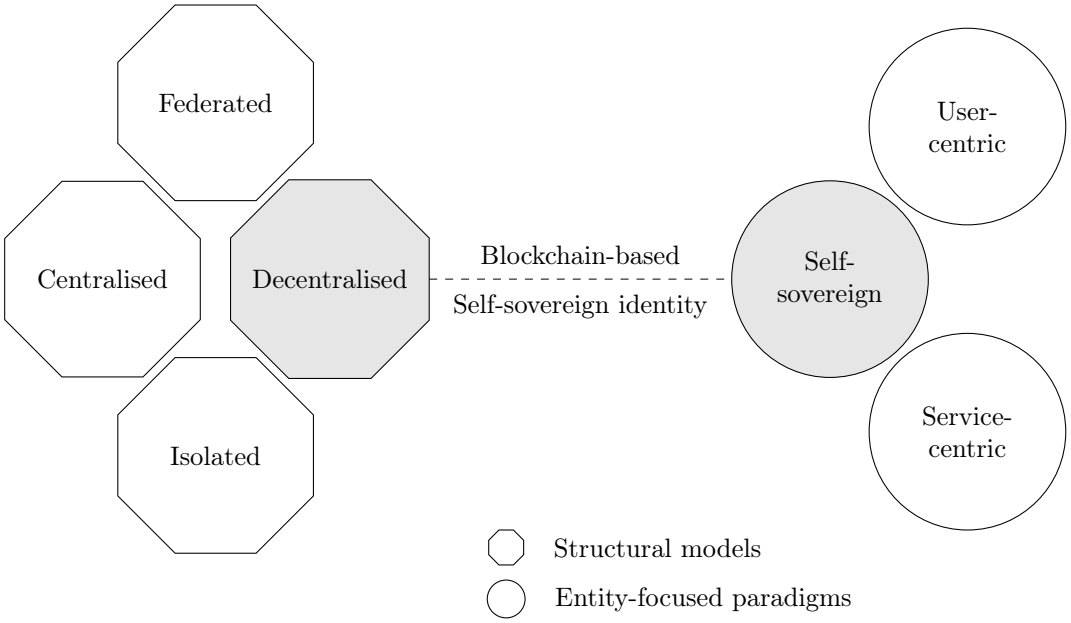
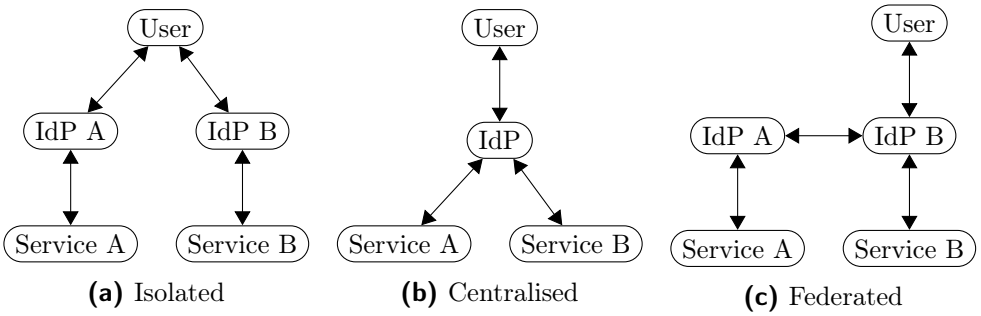


Figure 2.1: Perspectives on identity management schemes

services. This approach leads to a comfort benefit for the user. Nonetheless, the identity provider emerges as a trusted third party with the respective dependencies towards the user and the service provider.

Furthermore, from a classical viewpoint, decentralised identity management refers to the integration of several identity providers at a particular service [34]. This process reduces the dependency towards a specific identity provider and balance identity management to several entities. However, the service provider requires integration to many identity providers. Additionally, the user needs to register several times. More precisely, we describe this setting as vertical decentralisation relating to a simple extension of the number of identity providers. Besides that, horizontal decentralisation reflects a component-based distribution to alleviate a single identity provider as a trusted third party. We elaborate in more detail on a specific horizontal decentralisation type based on blockchain technology in Section 2.6.

Moreover, federated identity management determines the usage of digital identities from one identity provider in the realm of another identity provider [37]. The pattern can be extended to an arbitrary amount of identity providers. All providers are affiliated in a federation and create a circle of trust [38] between the domains where their digital identities are applicable. As an advantage for participating in the federation, the service provider and the user requires only one integration respectively registration. Nonetheless, the entire federation is only as



(a) Isolated **(b)** Centralised **(c)** Federated

Figure 2.2: Traditional identity management models

strong as the weakest affiliate. Fig. 2.2 provides a graphical overview of isolated, centralised and the federated model.

In addition to the structural models, identity management paradigms exist that focus on a certain actor within the identity management ecosystem. The traditional paradigms differentiate the service-centric and user-centric concentration [39]. The service-centric paradigm was not specifically developed, but it implicitly originated from the isolated pattern. Service-specific identity management is aligned with the service. It fulfils best the requirements of a service. Subsequently, the user-centric paradigm was proposed [36]. User-centric identity management concentrates on the *"usability and cost effectiveness"* [36] from the user's perspective.

The self-sovereign paradigm was postulated as further development of the user-centric concept. Identity management should not only focus on the user but transfer the control and ownership to the user. The user is the most important entity that is reflected by the digital identity. A blockchain-based self-sovereign identity solution connects the self-sovereign identity paradigm along with the blockchain capability for decentralisation. We may refer to this concept solely by self-sovereign identity and omit blockchain as a phrase. Additionally, self-sovereign identity also relates to an identity of a specific implementation of this concept. The context differentiates the meaning. Otherwise, we explicitly state by referring to either the concept or identity.

2.2 Trust in Identity Management

Trust is a significant social phenomenon between persons and organisations that is studied in various scientific disciplines, for instance, psychology, economics and computer science [16]. Interpersonal relationships are determined by trust. A person may react differently based on its individual judgement of trust. Subjectivity is a core characteristic of trust and lies in its nature [40]. Therefore, trustworthiness

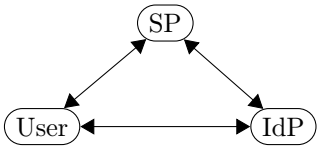


Figure 2.3: Identity management actors dependency triangle

has a different meaning to distinct persons. An individual might be trusted by person A, but not by person B because both individuals apply disparate criteria.

Additionally, trust might not be transitive and inverse in all situations [41]. If a person A trusts person B and person B trusts person C, it is not necessarily the case that person C is also trusted by person A. Additionally, person B might not trust person A only based on the reverse trust relationship. Moreover, a trust relationship is specific to the contextual setting [40]. As a result, an individual is trusted in a certain area and mostly not trusted in general.

The complex nature of trust leads to manifold definitions and characterisations [42] [43] originating in the different scientific domains. In computer science, the definition of decision trust by Josang et al. [42] that is based on previous research of McKnight and Chervany [44] is most relevant to identity management from our perspective [15]. Decision trust is characterised as

> *"the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible".*

The definition underlines three major elements [45] [15]. The first section of the denotation names interacting entities. A party that is dependent and a party that is representing the dependency. In the trust context, these entities are named trustor and trustee as well. The trustor has trust or has no trust in the trustee for a matter. In identity management, both roles are occupied by the identity or service provider and the user depending on a specific circumstance. The situation is the second component of the definition. In a particular situation, the dependency between the entities occur. In identity management, the processes that are implemented by the identity provider are of specific interest. The third part of the denotation refers to an adverse impact that may occur if the dependency is encroached from the perspective of the relying party. The trustee behaves differently than the trustor expects. Therefore, consequences emanate that are harmful for the trustor. These concerns reflect the risk of the relationship. Fig. 2.3 depicts the actors dependency triangle.

2.3 Identity and Attribute Assurance

Identity assurance describes the reliability that a digital identity matches to a real-world person and therefore defines trust in the identity provider [46]. The assurance expresses the certainty that the matching holds true. The higher the certainty, the more confidence a service provider has that the data of the identity is correct. Identity assurance can be dissected in different themes.

Authentication assurance implies that the right person is using the identity [35]. A low assurance is based on one factor authentication. A higher assurance is achieved by using multiple factors. Attribute assurance is another component and refers to the quality of the attributes of a digital identity [46]. In particular, the verification of an attribute to ensure correctness is important. Plentiful identity assurance frameworks exist that contain levels of assurance for expressing different verification grades.

The Kantara Initiative, the successor of the Liberty Alliance project, is a professional association working in the field of identity assurance. Its major deliverable is an identity assurance framework [47] to drive consistently managed identity trust services. The objective of the framework is to improve interoperability and comparability between identity and attribute providers. The identity assurance framework defines four levels of assurance referring to authentication and attribute assurance. Level 1 represents very low confidence in an attribute assertion. Some confidence is provided with an assertion of level 2. Level 3 and level 4 guarantee high, respectively, very high confidence in the claim. With regard to proofing criteria of attributes, level 1 attributes are self-asserted. Level 1 should only be used if no negative impact can occur in case the attributes are false. On level 2, an attribute verification process needs to be executed to determine the correctness of a property. For instance, official identity information, id card or passport can be used for appropriate verification. A wrong characteristic might lead to a moderate impact. The high confidence on level 3 and very high on level 4 demands even stronger attestation and verification of attributes. A direct definition is omitted in the description of the assurance levels. However, the consultation of several legal documents or additional proprietary verification methods might be appropriate.

The E-Authentication Guidance for Federal Agencies by the Office of Management and Budget [48], the e-Government Strategy by the UK Office of the e-Envoy [49] and the Canadian Identity and Attribute Assurance Guidelines by the Canadian Government [50] also apply four levels of assurance in a similar manner. Despite that, the eIDAS regulation [51] by the European Union refers to three assurance levels that are characterised as low, substantial and high. Identity assurance and the elaborated frameworks strive to objectify the assurance of identity data. Nonetheless, the assurance frameworks with three to four levels are

very coarse-grained. Moreover, the service provider is still required to trust the identity provider to adhere to the implemented assurance level.

2.4 Blockchain Technology

In 2008, Satoshi Nakamoto [14] published the foundational paper about Bitcoin describing a completely decentralised digital cash system. Bitcoin combines already invented schemes to form an irreversible chain of blocks that is extended under the governance of a consensus algorithm [52]. The peers of a network hold a copy of the chain of blocks and execute the consensus algorithm. As long as the majority of the nodes are honest and follow the agreed consensus approach, security properties of the system hold true [53]. Various definitions of blockchain exist that give priority to a specific aspect. For instance, the characteristic of providing a distributed database or ledger is emphasised [54]. However, in general, the term blockchain refers to systems that are aligned to a certain extent to the properties of Bitcoin [52]. We follow the latter more generic notation.

A remarkable peculiarity of a blockchain is the non-existence of a central authority to mediate communication or ensure proper operations. Participants in the blockchain network are equitable peers. These nodes can be identified by a public key or a derived identifier as an address. The corresponding private key ensures proper authentication. Nodes can be differentiated based on its level of participation in the blockchain network. A node might only read the chain of blocks or actively contributes to the extension by sending messages and creating new blocks. A new message may contain a transfer of coins from one address to another or further information. The messages are distributed between the nodes by a peer-to-peer communication scheme without a trusted third party. A new block persists the messages and additionally contains a cryptographic hash of the predecessor block. The node which creates the new block issues it to the other nodes. If the block conforms to the rules of the consensus algorithm, the remaining nodes accept it as the newest block and append it to their copy of the chain. At the same time, a decentralised database comprising the blocks including the messages with a copy on each node is established.

Generally, blockchains are two-dimensionally clustered along accessibility and required privileges [55]. Accessibility differentiates the type public, anybody can participate, and private, only selected entities take part in the blockchain network. The category privileges separates the permissioned and the unpermissioned kind. Bitcoin is a public and unpermissioned blockchain. Thereby, nodes can join and leave the network on their own discretion. The majority of honest nodes is reflected by the majority of computational power to solve a costly mathematical puzzle. This Proof of Work [56] mechanism is applied to defeat the Sybil attack [57]. On

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

Table 2.1: Allen's SSI Priniciples categorized by the Sovrin Foundation [55]

the contrary, permissioned blockchains might adopt a simple majority-based voting scheme because the respective nodes are privileged to elect. In both settings, there is no central authority to decide on the next valid block.

Where Bitcoin is strongly focused on digital cash and the transfer of coins within a message, further developed blockchains concentrate on the decentralised execution of arbitrary programs. A general-purpose blockchain for decentralised computation of smart contracts is, for instance, Ethereum [58]. Messages in an Ethereum network can contain new smart contracts or the execution results of existing smart contracts. The program code is also persisted within a block of the chain and available to all participating nodes. Additionally, data can be stored in smart contracts on the blockchain. However, as the data is available to all nodes, adherence to privacy and minimalisation principles is required.

Moreover, decentralised program execution might use feedback about events or information that is external to the blockchain. For instance, exterior effects encompass weather conditions, political events or stock courses. These information providers are generally referred to as oracles [59]. It is essential that decentralised program execution is not bound to an authority by using an external oracle. In this case, centralisation is again introduced.

2.5 Self-Sovereign Identity Principles

Allen [11] proposed the new self-sovereign identity paradigm based on ten principles as the advancement of the user-centric identity management concept. These axioms cover the domains security, controllability and portability [55]. Table 2.1 provides an overview of the categorized principles.

Within the realm of security, Allen states protection, persistence and minimalisation as important objectives. The term protection references the principle to safeguard the user's privileges. The rights of the digital identity's owner have precedence in case of failure of the identity provider. Persistence refers to the long-term existence of the identity. The digital identity is created by its owner and exists until it is intentionally removed by the owner. Moreover, the disclosure of information about the digital identity should be as limited as possible. This

principle is called minimalisation. Privacy-preserving techniques are applied to reduce the amount of exposed information.

The area controllability encompasses existence, control and consent. Existence expresses that a digital identity references an entity, object or person in the real world. The digital identity does not solely exist for itself. The control of the identity is entirely in possession of the user represented by the identity. In the case of objects, organisation or other entities, the respective owner has the control. The axiom of consent implies that the permit of the subject is required for any usage of the identity. Storing and presenting information about the digital identity, particularly claims or attestations or any other data, demands the permit of the digital identity's owner.

The last group of principles targets portability and is comprised of interoperability, transparency and access. The identity and the corresponding identity provider services should be interoperable with customers and service providers, for instance, by applying standard protocols. An identity should be widespread usable at many services. Transparency references a transparent implementation, operation and actioning of the identity provider functions to all involved parties. In particular, all actions should be transparent to the user. Access to information or attributes of the digital identity is easily possible by the owner or any legitimate party.

Overall, the user should have all control about the digital identity and its usage.

2.6 Structure of a Blockchain-based Self-Sovereign Identity

A common feature of traditional identity management models is the implementation of an identity provider. The identity provider might be part of a specific service or is a dedicated entity. Nonetheless, the identity provider is implemented as a common application and reflects a trusted third party.

We determine six characteristics that require decentralisation to eliminate this central authority [15] and provide full control to the user about its identity. We elaborate on the characteristics identification, authentication, attributes, storage, execution and organisation in the following subsections and discuss the capability of blockchain technology for their decentralisation. The first four properties provide at the same time a structure and the essential components for a self-sovereign identity [30]. The examination of these factors is a conjoint work with A. Mühle [30].

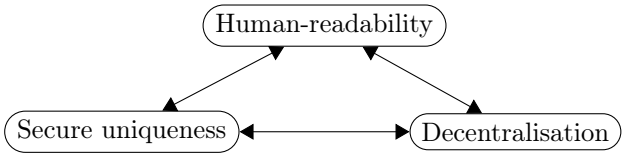


Figure 2.4: Zooko's [60] triangle

2.6.1 Identification

The identifiers of the digital identity are built within a namespace. All identifiers of a certain identity provider must be unique to ensure unambiguous identification. Identifier uniqueness within a namespace is a challenge that is not specific to identity management. Already in 2001, Z. Wilcox-O'Hearn [60] coined the term Zooko's triangle that expresses human-readability, secure uniqueness and decentralisation as contradicting objectives for designator generation (see Fig. 2.4). Human-readability refers to a name that implies a human understandable meaning. Secure uniqueness relates to collision-free generation of new identifiers. And decentralisation describes designator generation without a central authority. Zooko's triangle actually states the assertion that an implemented system may only achieve two out of three of these conflicting objectives. Nonetheless, identifier uniqueness is a foundational requirement that cannot be compromised.

There are two principal solutions without the use of blockchain. These options concentrate on either the human-readability or the decentralisation approach while complying with the uniqueness axiom. At a central authority, the identifier can be registered by applicants. The registered objects can be arbitrarily structured, including a human-readable notation. The central authority accepts new registrations and validates them against existing designators. The registration of web addresses for the domain name system or the issuance of X.509 certificates [61] are examples. In both cases the name is human-readable, and a hierarchically structured authority testifies the ownership of the notation. Certificate authorities that represent trusted third parties issue X.509 certificates.

In contrast, the decentralised approach applies probabilistic generation of random identifiers within a large namespace to avoid collisions. Each entity that requires a new designator creates it randomly and decentralised on its own. There is no alignment with other participants acting in the same namespace. The collision freeness relies on probabilistic assumptions during the random generation of the identifier and the number of possible values within the namespace. An example of this methodology is Universally Unique Identifiers (UUID) version 4 [62]. A UUID in version 4 are random or pseudo-random numbers with potentially 2^{122} different values. Thus, the likelihood is extremely low that two different entities randomly generate the same number independently from each other.

Both the central registration authority approach or the decentralised random generation process achieve only two of the three objectives of Zooko's triangle. However, decentralisation is a crucial requirement for self-sovereign identity. In addition to that, human-readable names are significant for inter-human communication. Deviating from Zooko's triangle, blockchain technology enables a solution to acquire all three targets. Either a decentralised registry based on a smart contract or a dedicated blockchain can be implemented. The enrolment of arbitrary names, including human-readable notions, is possible without a central authority. The peers of the blockchain network agree on newly registered names with support of the inherently used consensus algorithm. Examples for this approach are Namecoin [63] based on the Bitcoin blockchain and the Ethereum Name Service (ENS) [64] that relies on the Ethereum network. The initially applied first come first serve registration logic lead to arbitrary name squatting. Therefore, it was superseded by bidding mechanisms.

The various self-sovereign identity solutions follow their own identifier generation practice. For instance, uPort [65] and Blockstack [66] create a random designator that can be translated with a naming service on the respective blockchain. As a superordinate naming standard, the World Wide Web (W3C) consortium created the Decentralised Identifier (DID) [67] norm. Each self-sovereign identity solution is associated with a short name. The DID of an identity is composed as did:shortname:identifier. The core of identification is a registry for all identifiers. Blockchain technology enables the decentralised implementation of two different registry types.

- **Identifier Registry:** The identifier registry contains all identifiers of the digital identities. It ensures uniqueness and acts as a lookup directory. Additionally, the binding to an authentication mode is maintained. Moreover, invalid or returned identifiers are marked as revoked. The revocation of an obsolete identifier prevents potential misuse.

- **Claims Registry:** The claims registry is an extension of the identifier registry. It holds besides the identifiers, references to the attributes of a digital identity. The removal of an attribute reference indicates a revocation of the specific property. The creation of a reference serves as timestamped proof of existence. Therefore, the issuance and revocation of an attribute are trustfully transparent to the public (cf. Chapter 2.6.3).

2.6.2 Authentication

The authentication process binds the physical entity to the digital identity by using a credential. The binding ensures that only the legitimate user is able to act

with the corresponding digital identity. The authentication credentials are categorised according to the credential type in knowledge, possession and biometry. The cluster knowledge encompasses, for instance, passwords or security questions that need to get remembered by the user. Private keys and devices for authentication, for instance, smart cards, belong to the category possession. Biometry refers to the physical characteristics of the user. During the authentication process, the user presents its credential to the identity provider. The identity provider verifies the credential with the stored information for the identity. Usually the presentation process aligns to the execution of a protocol and the credential is not directly revealed. As a result, a decision is made that indicates a successful or failed authentication. This verification process is usually a centralised activity executed by the identity provider.

Moreover, authentication methods are distinguished in self-authenticating and non-self-authenticating patterns.

- **Self-authenticating:** A self-authenticating method does not require credential verification by a central authority.
- **Non-self-authenticating:** A non-self-authenticating approach needs a trusted third party for credential verification.

Self-authenticating methods comprise, for instance, public/ private key pairs and cryptographic hash functions [68]. A user generates decentralised and randomly a public/ private key pair. The public key can be used in a straightforward case as identifier directly. Besides that, a naming system opens up the possibility to connect it with a human-readable name. During the authentication process, a protocol runs to proof that the user is in possession of the private key which belongs to the presented public key. The execution of this protocol does not require a trusted third party. On the contrary, a simple username and password authentication are non-self-authenticating. At many online services, an email address is applied as a username. An arbitrary password has no mathematical connection to the username. Therefore, a central authority is required to run the verification process during authentication. Self-authenticating schemes are of particular interest for eliminating a central authority in the domain of authentication.

uPort uses the address of a smart contract as identifier of an identity and applies public key cryptography for authentication [65]. A similar authentication method is also used for Blockstack [66]. The usage of public key cryptography as self-authenticating scheme fosters the decentralisation characteristics of blockchain.

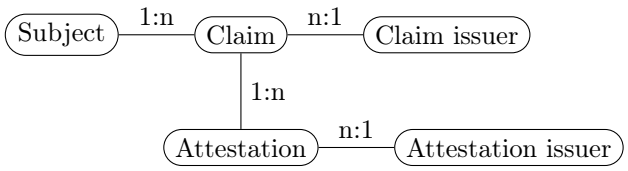


Figure 2.5: Relations between components of verifiable claims

2.6.3 Attributes

Attributes are the important characteristics of a digital identity that enables the service provider to make decisions on service provisioning. In traditional identity management models, attributes are issued by the identity or attribute provider. The use of a single provider indicates a trusted third party. In the self-sovereign identity model attributes are called claims. A claim is a statement about the user and specifies, therefore a property. However, a claim itself might be correct or false. Therefore, attestations assert the correctness of the claim's content. A self-attested claim is issued by the user itself and might not be trustworthy. Despite that, an attestation from another entity provides more credibility.

A verifiable claim [69] is comprised of a claim and a related attestation. The verifiable claim contains an identifier that refers to the user's digital identity. Additionally, the attestation is a cryptographic signature by another identity that can be verified. Moreover, a verifiable credential combines several claims and their attestations that belong to a common context. Fig. 2.5 outlines the associations between the elements and actors of a verifiable claim. A claim issuer publishes a certain claim. The claim can hold one or more attestations that are issued by different attestation issuers. Additionally, a claim is specific to a particular subject.

The notion of a verifiable claim supports a plentitude of attestation issuers and therefore attribute providers. Applying a set of attribute providers reduces the dependency to a dedicated provider and fosters decentralisation. Moreover, according to the claim registry paradigm, using the blockchain enables a public verifiable timestamp as proof of claim creation and a decentralised single point of revocation. Thus, there is a central registry for claim validity that is implemented in a decentralised manner to avoid the engagement of a trusted third party.

2.6.4 Storage

Information about a digital identity and attribute data requires storage that is accessible for all entities. The identity provider usually implements a centralised storage that is under its full control for creation, modification and deletion of objects. Blockchain-based self-sovereign identity solutions relocate the identity provider-owned storage to a solution-specific or user-defined position. Information

that can be or must be publicly available is stored on-chain. The on-chain storage refers to persisting data on the blockchain. This data is available to all nodes of the blockchain network and cannot be deleted anymore. Before using on-chain storage, data privacy considerations demand an in-depth analysis. The identifier of an identity and authentication information, for instance, the public key, are typically stored on-chain.

On the contrary, information that should be kept private to a certain extent must be stored off-chain. By preserving any information on the blockchain about a verifiable claim, the advantages of a claim registry cannot be applied. Storing on-chain a reference to a verifiable claim preserves privacy and gains benefits from the claim registry model. Within both approaches, storage for the verifiable claims that is under the control of the user is needed. Storing claims on the user's device in the identity wallet is an option. For instance, uPort and Jolocom [70] follows this proposition. Furthermore, centralised cloud storage providers, e.g. Amazon S3, Dropbox, Google Drive, can also be chosen by the user. For a specific claim, the storage would be under the control of the respective organisation. However, the user might choose several providers for different claims or locate a specific claim at several hosts. Besides this, the InterPlanetary File System (IPFS) [71] is a decentralised storage approach. IPFS implements a decentralised peer-to-peer file system. Overall, the selection of a suitable off-chain storage solution lies in the control of the user.

2.6.5 Execution

An identity provider is realised as software and hosted in a server environment that is under control of the hosting entity. The used program might be of proprietary nature and is not disclosed to the public. Even in the case that open-source software is used, it is challenging to verify that the published code is also executed on the server. Trust is required into the central entity to execute the identity provider software in accordance with published properties and in adherence to contractual agreements.

An identity provider implementation based on blockchain supports the decentralisation of the execution and its environment. Program code is publicly available to all nodes of the blockchain network. Therefore, verification by other entities is possible. In addition to that, the nodes need to agree on execution results by applying the consensus algorithm of the blockchain network. Thus, the execution is holistically verifiable, and the execution of the identity provider is decentralised.

Component	Decentralised Option (Blockchain)	Centralised Option
Identification	Identifier or claim registry	Central registry
Authentication	Self-authentication scheme	Non-self-authentication scheme
Attributes	Verifiable claim	Regular attribute
Storage	User-defined Storage	Identity provider-owned Storage
Execution	Peer-to-peer network	Centrally hosted
Organisation	None or diverse committee	Organisation

Table 2.2: Decentralised and centralised variants of identity components

2.6.6 Organisation

The identity provider is operated by an organisation, consortium or any other entity. This entity takes care of the required financial support and decides on the offered service, including the actual operations. The operating party represents a central authority and may influence the service of the identity provider.

The decentralised implementation of an identity provider supported by blockchain can remediate the organisation as a trusted third party. In case an unpermissioned blockchain is used, nodes of the network can join at their own preference. Nodes join the blockchain network based on an inherent incentive to receive tokens. Bitcoin [14] and Ethereum [58] belong to the unpermissioned category. There is no authority that may influence the operation of the identity provider. Besides that, the use of a permissioned blockchain requires an entity to grant privileges for participating in the blockchain network. Hyperledger (HL) Indy [72] is a representative of this class. The structure of the permission granting body is essential to determine if it is a central authority. In the case of HL Indy, the Sovrin foundation [73] runs a public network that is governed by a diverse committee. The board applies a voting scheme to admit new stewards into the network. Such a diverse committee might not be seen as a trusted third party.

2.6.7 Synopsis

Table 2.2 presents an overview of the presented decentralised and centralised options for the identity components. Blockchain technology allows the implementation of a decentralised identifier or claim registry in contrast to a centralised catalogue. By using public/ private key cryptography, a self-authentication scheme is used within a blockchain network. Non-self-authentication solutions comprise, for instance, login with a username and password. These solutions require a central authority for verification. Concerning attributes, the decentralised approach uses verifiable claims from various issuers, whereas regular user attributes are delivered by a single identity provider. Moreover, the user can decide on a storage location of its properties in the decentralised pattern. The centralised option persists

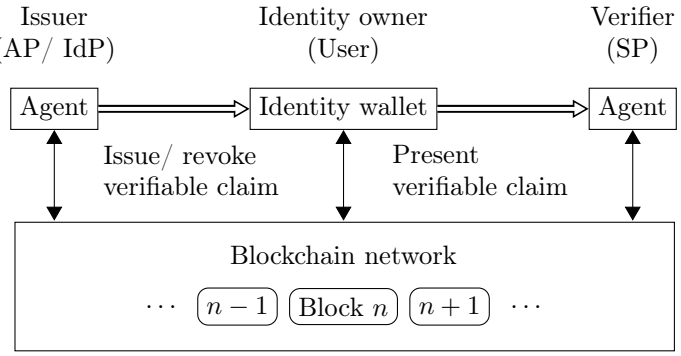


Figure 2.6: Self-sovereign identity actors and interaction

user attributes on an identity provider-owned location. In addition to that, the blockchain enables a decentralised execution pattern within a peer-to-peer network that is operated by a diverse committee in the permissioned case or without priorly specified entities in the unpermissioned setting. The centralised variant regularly hosts an identity provider that is supervised by an organisation.

2.7 A Self-Sovereign Identity Management System

The traditional actors in identity management change their roles and interaction paths in the self-sovereign identity setting [21]. The identity owner represents the user that is embodied by the identity. The identity owner applies an identity wallet to manage its identity. In particular, the identity wallet provides means to create an identifier, to obtain verifiable claims and to communicate with issuer, verifier, and the blockchain network. The issuer originates and revokes verifiable claims to the user. In the traditional models, the identity or attribute provider fulfils the role. Thereby, the issuer uses an agent to interact with the blockchain network. Moreover, the verifier receives verifiable claims and checks signatures and validity periods. For this purpose, the verifier also implements an agent. The verifier is comparable to the service provider that offers a service. The blockchain network implements the decentralized options of a self-sovereign identity (cf. 2.6) to form a decentralized identity provider. Overall, the agent, the identity wallet and the blockchain network form a self-sovereign identity management system [21] that is used by the actors with changed roles compared to the traditional models. Fig. 2.6 depicts the self-sovereign identity actors and communication paths.

2.8 Summary

In this chapter, we presented foundational topics to outline the background, development and structure of the blockchain-based self-sovereign identity concept. To achieve this, we have introduced identity management, including the existing models and different paradigms. The self-sovereign identity paradigm is a further development of user-centric identity management, and blockchain enables a relevant decentralised implementation model. Additionally, we presented the ten foundational principles that constitute a self-sovereign identity. Moreover, trust is an essential component in identity management between the actors user, service provider and identity provider. The identity provider is a trusted third party for the other entities. If the identity provider abuses the trust, it implies negative consequences. Specific areas of trust in identity management are identity and attribute assurance. In particular, attribute assurance refers to the certainty that the attributes of an identity match the user's real properties. Therefore, identity providers implement verification procedures, and service providers rely on correct attributes for service provisioning. Additionally, we outlined key aspects of blockchain technology for decentralisation of trust and its design in general. Supported by blockchain, a trusted third party can be eliminated, and a network of equitable peers executes the required program. Applied to identity management, the identity provider as a central authority is removed in the setting of blockchain-based self-sovereign identity. In this regard, we have described the components identification, authentication, attributes, storage, execution and organisation as relevant for decentralisation with corresponding implementation approaches, whereas the first four elements compose a self-sovereign identity.

3 Trust Requirements in the Context of Self-Sovereign Identity

In this chapter, trust requirements in the context of blockchain-based self-sovereign identity are described [15]. We start with trust domains, assigned requirements and our evaluation methodology. Subsequently, we define schematic patterns and analyse them according to trust requirements. Finally, we compare the results to the isolated, centralised and federated identity management model and conclude on the shift in trust requirements.

3.1 Motivation and Related Work

As identity management is fundamental for the security of any online service, the understanding of trust requirements between the identity provider, service provider and user is essential to identify dependencies. In particular, the identity provider is a trusted third party to the other entities. Strong dependencies based on trust are favourable for the trustee. The trustee has a powerful position that could be misused to endanger the trustors. The evolution of the self-sovereign identity paradigm eliminated the identity provider as a trusted third party. Thus, trust requirements in this context change significantly compared to the traditional identity management models. A detailed understanding of the trust requirements supports the comprehension of the power structure.

Jøsang et al. [17] analysed common trust requirements in isolated, centralised and federated identity management. Additionally, the authors studied personal authentication management with regards to trust requirements. The analysis perspective encompassed the service provider and the user, whereas the service provider also comprised the identity provider. Centralised identity management is split into the categories common-identifier domain, meta-identifier domain and single-sign on. Moreover, identity federation is organised in constellations with a different quantity of users and service providers. Personal authentication management is analysed regarding the tamper-resistance of devices. Based on Jøsang et al.'s work, Kylau et al. [18] assessed in detail trust requirements in identity feder-

ation topologies. Thereby, Kylau et al. defined direct and further trust patterns. The extended patterns additionally include indirect trust relationships. In conclusion, the authors conducted a trust and risk comparison between the different schemes. The required trust and associated risks increase aligned to the growing complexity of the pattern. Besides the mentioned studies, Ferdous and Poet [74] investigated attribute aggregation models in federated identity management. The analysis considers trust, risk and functional requirements. Ferdous and Poet grouped the examined models in a classification scheme according to the location of attribute aggregation. The positions comprise the side of the service provider, identity provider and the standpoint of the user.

On the contrary, in our study, we concentrate on the blockchain-based self-sovereign identity setting. Within this context, we define schemes and evaluate them for trust requirements. Moreover, we compare the results to the traditional identity management models and consider in particular, the use of attribute aggregation methods.

3.2 Trust Domains and Requirements

The trust requirements are categorised into trust domains. These trust domains reflect situations where a dependency exists between the trustor and the trustee. The misuse of this dependency may result in a negative impact for the trustor as devised by the decision trust definition (cf. Chapter 2.2). For the evaluation of the patterns, we use the following trust domains and the single requirements.

- **Privacy:** The trust domain privacy relates to the confidentiality of user-related information. The attributes of the digital identity are comprised of personal identifiable information of the user. Moreover, the identity provider and the service provider can preserve usage statistics of the digital identity and associated properties over time. The subject of the identity is interested that only the absolutely necessary information is stored and disclosed in case consent is obtained.
  - **T1a:** The identity and the attribute provider protects the privacy of the user.
  - **T1b:** The service provider protects the privacy of the user [17].
- **Credential Management:** The credential associated with a digital identity must be managed securely by the identity provider to avoid impersonation attacks. The generation process, the modification of the credential, the distribution to the user and the storage of verification information requires a secure process. Moreover, the user has an obligation to protect its credential.

In particular, the user has the liability to not deliberately disclose the own credential to other parties. Intentional credential sharing leads to deniability of actions that are conducted by the identity.

  - **T2a:** The identity provider adheres to secure credential management [17].
  - **T2b:** The user protect its credential [17] and does not deliberately disclose it to other parties.

- **Authentication:** The service provider requires user authentication at its services. Upon an authentication request, the user is redirected to the identity provider and proofs to be in possession of the adequate credential for its identity. Subsequently, the user is returned to the service provider and logged in to its service based on a successful authentication result. Furthermore, the service provider maps the identity of the user to an internally administrated data set containing additional information.
  - **T3a:** The identity provider authenticates the user properly [18].
  - **T3b:** The service provider manages the user mapping correctly [17].
- **Attribute Management:** Attributes are a fundamental component of a digital identity. Service provider relies on correct attributes for their service provisioning. The properties that are issued by the identity or attribute provider must reflect the reality. A timely revocation of attested attributes is essential to reflect changes.
  - **T4a:** The identity and attribute provider delivers correct attributes.
  - **T4b:** The identity and attribute provider revokes invalidated attributes in a timely manner.

3.3 Pattern-based Trust Evaluation

We abstract from the peculiarities of the different blockchain-based self-sovereign identity implementations to universalised architecture patterns. These patterns enable the assessment of the trust requirements between the entities. The distinct actors within the architecture patterns are the service provider and the attribute provider. Additionally, the decentralised identity provider solely fulfils identity management functions except for attribute management. The attribute management capability is dedicated to the attribute provider. Fig. 3.1 to Fig. 3.5 depict the patterns. In the diagrams, rectangular shapes with rounded edges represent the service provider and the attribute provider. The decentralised identity provider

is a central circle that is indicated by a dashed line. The dashed line connects the other actors for identity management. An arrow between the service provider and the attribute provider connotes the sharing of attributes. An aggregated usage of attributes is reflected in case the arrow connects several attribute providers with a service provider.

The actors relate differently to each other in the architecture patters. We study bilateral relationships between the actors and assess the trust requirements associated with the connections. Moreover, we determine for each dependency a coarse-grained trust level. The trust level differentiates the strength of the dependency between the actors for a specific requirement. We distinguish the following categories.

- **Absolute:** The dependency requires absolute trust from the trusting actor. The trustor is fully dependent on a trusted third party. There is no major compensating control or the possibility to distribute the trust towards several trusted entities.
- **Limited:** The dependency solely requires a limited amount of trust by the trusting actor. A significant compensating control to verify the behaviour of a trusted third party exists. Besides that, there might be several trusted entities to distribute the dependency to. As a result, the trustor does not rely on a single authority anymore.

3.4 Self-Sovereign Identity Trust Patterns

The patterns reflect schematically trust relationships within the self-sovereign identity context. We start the evaluation with the simple bilateral integration scheme. Subsequently, we outline the multiple aggregated integration, multiple side-by-side integration, multiple service provider integration and the aggregated and side-by-side integration pattern.

3.4.1 Bilateral Integration

The bilateral integration pattern is the simplest scheme in the context of self-sovereign identity. The diagram comprises a service provider and an attribute provider that are connected by a decentralised identity provider. Fig. 3.1 depicts the bilateral integration. The decentralised identity provider implements functions for authentication, credential management and the registration of the identifier. Moreover, the attribute provider delivers the required properties of users. The user registers at the decentralised identity provider to create an identifier and

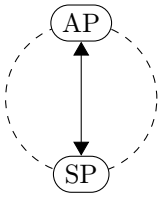


Figure 3.1: Bilateral integration

an authentication credential. Additionally, the user provides information to the attribute provider for the verification of its characteristics and the issuance of verifiable claims. When a user requests access at a service provider, it is redirected to the decentralised identity provider for authentication. The user runs the protocol for the self-authenticating access scheme with the decentralised identity provider. Public key cryptography builds the foundations of the protocol. The user proofs the possession of a corresponding private key. Subsequent to a successful authentication, the user returns to the service provider, and the attributes of the user are conveyed. Based on this, the service provider conducts the access decision for the user.

Considering the trust domain privacy, the user must trust the service provider and the attribute provider to adhere to privacy requirements. The trust requirements *T1a* and *T1b* fully apply from the user as trustor towards the service and attribute provider as trustee. Both entities are not transparent for the user. There is no compensating measure in this domain that limits the dependency. The decentralised identity provider implements transparently credential management and user authentication routines. Based on the properties of blockchain, a public verification of these routines is possible. Therefore, no trust is required between the parties. As a result, the trust requirement *T2a* that targets the user's trust in secure credential management and the trust demand *T3a* that describes proper authentication carry no weight. The applicability of these trust requirements strongly depends on the decentralised identity provider based on blockchain. This trust posture is equal in all patterns as the decentralised identity provider is an integral component in all schemes. On the contrary, trust is demanded by the user towards the service provider in an appropriately implemented user mapping. Due to the closed nature of the service provider implementation, no public verification is possible. Moreover, there is no compensating control that limits the demanded trust. Thus, the trust requirement *T3b* fully applies in this context. Analysing attribute management, the user expects the attestation and delivery of correct and valid attributes. Attributes that become invalid due to external impact must be revoked immediately. Therefore, trust requirements *T4a* and *T4b* are fully relevant in the bilateral integration pattern. No limiting factor exists in this context.

Examining the trust requirements from the perspective of the service provider, the service provider trusts the attribute provider to preserve the privacy of the user. The service provider requires user acceptance of its trusted attribute providers to satisfy its customers. Thus, the attribute provider is a trusted third party towards the service provider. Therefore, the trust requirement *T1a* is completely applicable in this setting. Moreover, the attribute management related trust requirements *T4a* and *T4b* are absolutely relevant for the relationship between the service provider and the attribute provider. The service provider offers its service on the grounds of correct and valid attributes to the user. The attribute provider is fully trusted in this regard. No compensating control or public verifiability exists in this context.

In addition to that, the service provider trusts the user to protect its credential and to not deliberately disclose it (*T2b*). The service provider has no option to verify the user behaviour with regard to this demand. Besides that, an identity that is used by somebody else than the owner due to a stolen credential may significantly harm the service provider. In contrast, the trust requirements regarding the secure credential management (*2a*) and proper authentication (*T3a*) are not relevant due to the decentralised identity provider implementation.

Exploring the attribute provider, it expects that the service provider protects the privacy of the user. The attribute provider is the source of the user's properties that might comprise personal identifiable information. The attributes are transferred to the service provider. Therefore, trust requirement *T1b* is fully applicable. In contrast, the attribute provider is indifferent to further trust demands.

3.4.2 Multiple Aggregated Integration

The multiple aggregated integration pattern is shown in Fig. 3.2. Within the pattern, a service provider and multiple attribute providers exist. The figure depicts paradigmatically two attribute providers. Nonetheless, a multitude of attribute providers can exist in an extended environment. The service provider consumes attributes of the user that are issued by both attribute providers. This form of attribute aggregation is used in a trust-enhanced manner. Trust-enhanced attribute aggregation increases assurance in the validity and correctness of the attribute value. Thus, the required trust in a specific attribute provider is reduced.

Aligned with the bilateral integration pattern, the user is required to trust the service provider and the attribute provider regarding privacy. Both entities must protect the privacy of the user. Moreover, no compensating control exists that limits the demanded trust. Therefore, the trust requirements *T1a* and *T1b* are fully relevant for the user. Additionally, the trust requirement *T3b*, that represents the correct user mapping by the service provider, applies absolutely between the

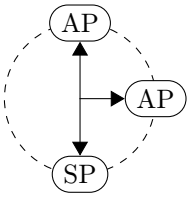


Figure 3.2: Multiple aggregated integration

user and the service provider. Furthermore, the service provider requires absolute trust towards the attribute provider for protecting the user's privacy (*T1a*). Additionally, the user guards the credential of its digital identity (*T2b*). Analysing the attribute providers perspective, the trust requirement *T1b* holds true towards the service provider.

In contrast to the bilateral integration, the domain attribute management demands trust differently between the actors. The service provider uses valid and correct attributes of the user for its service provisioning. Incorrect attributes may lead to a negative impact on the side of the service provider or the user. A trust-enhancing usage of attribute aggregation from several attribute providers reduces the required trust in attribute management. A single attribute provider is neither a trusted third party for the user nor for the service provider. There is no solitary dependency on an attribute provider. Therefore, the trust requirements *T4a* and *T4b* are solely applicable in a limited manner for the user and service provider as trustors.

3.4.3 Multiple Side-by-Side Integration

The multiple side-by-side integration scheme is comprised of a service provider and several attribute providers. Comparable to the multiple aggregated integration pattern, this diagram is visualised in Fig. 3.3 and modelled with two attribute providers. In this pattern, the service provider also receives aggregated attributes from distinct attribute providers. As differentiating factor, the attributes are aggregated to complete the required set of properties at the service provider. A single attribute provider cannot deliver all properties of the user that the service provider demands. Thus, no trust decrease for attribute management is achieved.

The multiple side-by-side integration pattern has the same trust requirements posture as the bilateral integration. The user must completely trust the service and attribute provider with regards to data privacy (*T1a* and *T1b*) and authentication (*T3b*). Service and attribute provider are non-transparent trusted entities for the user. Moreover, the service provider expects absolute adherence to trust demands data privacy (*T1a*) towards the attribute provider and credential management (*T2b*) with regards to the user. The attribute provider fully trusts the service

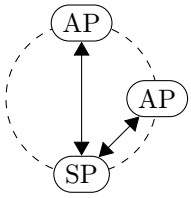


Figure 3.3: Multiple side-by-side integration

provider regarding the privacy protection (*T1b*) of the user. Additionally, the service provider and the user completely trust the attribute provider concerning attribute management (*T4a* and *T4b*).

3.4.4 Multiple Service Provider Integration

The multiple service provider pattern is comprised of one attribute provider and several service providers. The model is presented in Fig. 3.4. The diagram shows representatively two service providers. Despite that, additional service providers can be presumed.

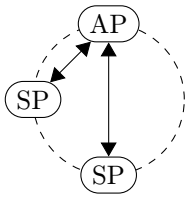


Figure 3.4: Multiple service provider integration

The trust requirements in this scheme are, to a great extent, comparable to the bilateral integration pattern. With regard to data privacy (*T1a* and *T1b*) and authentication (*T3b*), the user has an absolute trust demand towards the service provider and the attribute provider. Moreover, the service provider fully trusts the attribute provider concerning data privacy (*T1a*). Reciprocally, the attribute provider absolutely trusts the service provider considering data protection (*T1b*). Furthermore, both parties, the service provider and the user, completely trusts the attribute provider for its attribute management processes (*T4a* and *T4b*).

In contrast, a difference exists concerning the trust requirement *T2b* that refers to the protection of the credential by the user. It further includes that the user also does not deliberately share its credential to other entities. The digital identity of the user is applicable at a multitude of service providers. Therefore, the user has one identity that represents it at many online services. On the contrary, in isolated identity management, a user has many service-specific identities. Thus,

the value of the identity is significantly higher compared to the usage at a single service provider. A user has immanent interest to protect its credential to avoid losing control and consumption of many services.

3.4.5 Arbitrary Aggregated and Side-by-Side Integration

The arbitrary aggregated and side-by-side integration model is outlined in Fig. 3.5. It is the most complex pattern reflecting realistic situations with plentiful actors. There are several service providers and attribute providers that are connected by a decentralised identity provider. Both attribute aggregation approaches are applied to decrease trust into a single attribute provider and to achieve a complete set of required attributes.

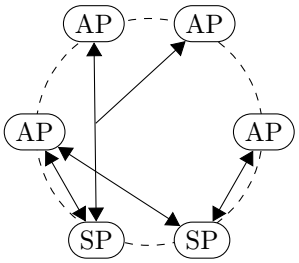


Figure 3.5: Arbitrary aggregated and side-by-side integration

The existence of trust requirements in this pattern is a composition of the trust demands in the previous patterns. The user needs to fully trust the distinct service providers and attribute providers concerning its privacy (*T1a* and *T1b*). Additionally, the user is completely dependent on the various service providers. They are trusted to have properly implemented the user mapping procedures according to *T3b*. In the matter of the trust domain attribute management (*T4a* and *T4b*), the level of dependency between the user, service provider and the attribute provider is twofold. In case, the trust-enhancing attribute aggregation strategy is used, the required level of trust is limited. On the contrary, if the attribute aggregation targets the completion of a property set, the demanded trust is absolute. Additionally, the privacy-related trust requirement (*T1a*) fully applies from the service provider towards the attribute provider. Moreover, the attribute provider completely expects adherence to the privacy requirement by the service provider (*T1b*).

Considering the trust demand for credential management (*T2b*), the user is trusted in a limited manner by the service provider and the attribute provider. The user has a significant interest in protecting the credential of its digital identity because the identity is valid at a large number of service providers. Additionally, plentiful attribute providers issued attestations for this identity.

3.5 Trust Requirements in Traditional Models

In this section, we list trust requirements in the isolated, centralised and federated identity management model to build a foundation for a comparative analysis with the self-sovereign identity patterns in the ensuing section.

3.5.1 Isolated Identity Management

In isolated identity management, the identity provider is service-specific and belongs to the service provider. A digital identity cannot be used at different services. Thus, the identity provider and the service provider are one entity [17]. As a result, no trust is required between the identity and the service provider.

Analysing the user's trust position, the user is faced with the service and identity provider as a trusted third party. Therefore, trust requirements concerning the privacy of the user's data (*T1a* and *T1b)* are fully applicable. Furthermore, the user completely trusts the identity provider for properly implemented credential management (*T2a*) and authentication processes (*T3a*). There is no option to publicly verify the implementation and the behaviour of the central authority. Additionally, the user is also demanded to completely trust the identity provider with regard to attribute management (*T4a* and *T4b*). There is no compensating control that reduces the required trust for the user.

Studying the side of the service and the identity provider, absolute trust is required in the user for credential management (*T2b*). That means, the user protects its credential and does not deliberately disclose it. There is no trust reducing measure as the digital identity is solely applicable at one service. Trust requirements between the service provider and the identity provider are not applicable because both belong to the same entity. Therefore, trust demands concerning privacy (*T1a* and *T1b*), credential management (*T2a*), authentication processes (*T3a*) and attribute management (*T4a* and *T4b*) are not relevant in the isolated setting.

3.5.2 Centralised Identity Management

The centralised identity management model further develops the applicability of a digital identity beyond the boundary of a specific service [39]. Therefore, the identity provider becomes an independent entity. An identity can be used at several services of an organisation or even at the services of different providers. If the identity provider and the service provider belong to the same organisational trust domain, these actors require solely limited trust between each other. In contrast, complete trust is demanded if the identity and service provider reside in different organisations with distinct trust domains.

From the user's perspective, the identity and the service provider is a trusted third party. Therefore, trust requirements concerning privacy (*T1a* and *T1b*), credential management (*T2a*), authentication (*T3a*) and attribute management (*T4a* and *T4b*) apply absolutely. In reverse, the trust prerequisite towards the user concerning credential protection (*T2a*) is twofold.

The trust demand is fully applicable if the identity provider is part of the organisational domain of the service provider. A limited trust requirement exists if the identity provider is a distinct organisation that caters for several service providers. In this situation, a digital identity is more valuable for the user. Therefore, trust is limited with regard to the user not deliberately disclosing its credential.

A comparable twofold situation endures when studying the trust relationship between the identity and the service provider. On the one side, the trust requirements for privacy (*T1a*), credential management (T2a), authentication (*T3a* and *T3b*) and attribute management (*T4a* and *T4b*) apply in a limited manner if both entities belong to the same organisation. Nonetheless, if the identity and service provider reside in distinct organisational trust domains, the trust prerequisites apply absolutely.

3.5.3 Federated Identity Management

In federated identity management, several identity providers are affiliated with each other in a federation. This association builds a circle of trust [38] with the corresponding service providers that trust any of the identity providers. Any identity provider is a trusted third party for the user and the service provider.

Examining the user's trust position, trust requirements concerning data privacy (*T1a* and *T1b*), credential management (*T2a*), authentication (*T3a* and *T3b*) as well as attribute management (*T4a* and *T4b*) are completely applicable towards the service provider and the identity provider.

From the service provider's perspective, the trust demands regarding privacy (*T1a*), credential management (*T2a*), authentication (*T3a*) and attribute management (*T4a* and *T4b*) exist fully towards the identity provider.

Analysing the viewpoint of the identity provider, the trust prerequisites for data protection (*T1b*), credential management (*T2a*) and authentication (*T3b*) completely apply towards the service provider.

Both, the service provider and the identity provider require solely limited trust in the user for credential protection (*T2b*) because the identity of the user can be used at all participants of the circle of trust. Therefore, the credential of the identity is highly valuable for the user.

3.6 Synopsis of Trust Requirements

Table 3.1 to Table 3.4 presents a thorough overview of the outlined trust requirements in the previous sections. The rows list the various identity management models and outlined patterns. The columns reflect the trust domains and categorised requirements. In the detailed matrix, the existing trust requirements between the actors for a specific pattern can be found. Thereby, the actor in the row represents the trustor, and the entity in the column reflects the trustee. Within the trust matrix, a dash (-) indicates no trust. A small dot ($\cdot$) implies limited trust and a large dot ($\bullet$) refers to an absolute trust requirement with no compensating control.

Considering the trust requirement data privacy (*T1a*) for the bilateral integration pattern in Table 3.1, we can deduce that the user and the service provider fully trusts the attribute provider. This is indicated by the large dot ($\bullet$). No further trust relations exist. Therefore, the other cells are marked with a dash (-).

Model		T1a			T1b			T2a			T2b		
		U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP
Isolated	U	-	-	●	-	●	-	-	-	●	-	-	-
	SP	-	-	-	-	-	-	-	-	-	●	-	-
	IdP/AP	-	-	-	-	-	-	-	-	-	●	-	-
Centralised	U	-	-	●	-	●	-	-	-	●	-	-	-
	SP	-	-	·/●	-	-	-	-	-	·/●	·/●	-	-
	IdP/AP	-	-	-	-	·/●	-	-	-	-	·/●	-	-
Federated	U	-	-	●	-	●	-	-	-	●	-	-	-
	SP	-	-	●	-	-	-	-	-	●	·	-	-
	IdP/AP	-	-	-	-	●	-	-	-	-	·	-	-

Table 3.1: Model trust requirements in domain privacy (*T1*) and credential management (*T2*)

Model		**T3a**			**T3b**			**T4a**			**T4b**		
		U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP
Isolated	U	-	-	●	-	●	-	-	-	●	-	-	●
	SP	-	-	-	-	-	-	-	-	-	-	-	-
	IdP/AP	-	-	-	-	-	-	-	-	-	-	-	-
Centralised	U	-	-	●	-	●	-	-	-	●	-	-	●
	SP	-	-	·/●	-	-	-	-	-	·/●	-	-	·/●
	IdP/AP	-	-	-	-	·/●	-	-	-	-	-	-	-
Federated	U	-	-	●	-	●	-	-	-	●	-	-	●
	SP	-	-	●	-	-	-	-	-	●	-	-	●
	IdP/AP	-	-	-	-	●	-	-	-	-	-	-	-

Table 3.2: Model trust requirements in domain authentication (*T3*) and attribute management (*T4*)

Pattern		T1a			T1b			T2a			T2b		
		U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP
Bilateral	U	-	-	●	-	●	-	-	-	-	-	-	-
	SP	-	-	●	-	-	-	-	-	-	●	-	-
	IdP/AP	-	-	-	-	●	-	-	-	-	●	-	-
Multiple Aggregated	U	-	-	●	-	●	-	-	-	-	-	-	-
	SP	-	-	●	-	-	-	-	-	-	●	-	-
	IdP/AP	-	-	-	-	●	-	-	-	-	●	-	-
Multiple Side-by-Side	U	-	-	●	-	●	-	-	-	-	-	-	-
	SP	-	-	●	-	-	-	-	-	-	●	-	-
	IdP/AP	-	-	-	-	●	-	-	-	-	●	-	-
Multiple Service Provider	U	-	-	●	-	●	-	-	-	-	-	-	-
	SP	-	-	●	-	-	-	-	-	-	·	-	-
	IdP/AP	-	-	-	-	●	-	-	-	-	·	-	-
Arb. Aggregated/ Side-by-Side	U	-	-	●	-	●	-	-	-	-	-	-	-
	SP	-	-	●	-	-	-	-	-	-	·	-	-
	IdP/AP	-	-	-	-	●	-	-	-	-	·	-	-

Table 3.3: Pattern trust requirements in domain privacy (*T1*) and credential management (*T2*)

Pattern		T3a			T3b			T4a			T4b		
		U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP	U	SP	IdP/AP
Bilateral	U	-	-	-	-	●	-	-	-	●	-	-	●
	SP	-	-	-	-	-	-	-	-	●	-	-	●
	IdP/AP	-	-	-	-	●	-	-	-	-	-	-	-
Multiple Aggregated	U	-	-	-	-	●	-	-	-	·	-	-	·
	SP	-	-	-	-	-	-	-	-	·	-	-	·
	IdP/AP	-	-	-	-	●	-	-	-	-	-	-	-
Multiple Side-by-Side	U	-	-	-	-	●	-	-	-	●	-	-	●
	SP	-	-	-	-	-	-	-	-	●	-	-	●
	IdP/AP	-	-	-	-	●	-	-	-	-	-	-	-
Multiple Service Provider	U	-	-	-	-	●	-	-	-	●	-	-	●
	SP	-	-	-	-	-	-	-	-	●	-	-	●
	IdP/AP	-	-	-	-	●	-	-	-	-	-	-	-
Arb. Aggregated/ Side-by-Side	U	-	-	-	-	●	-	-	-	·/●	-	-	·/●
	SP	-	-	-	-	-	-	-	-	·/●	-	-	·/●
	IdP/AP	-	-	-	-	●	-	-	-	-	-	-	-

Table 3.4: Pattern trust requirements in domain authentication (*T3*) and attribute management (*T4*)

3.7 Comparative Analysis

The traditional identity management models have evolved from the isolated to the centralised and to the federated identity model. During this development, the trust requirements between the bilateral relations of the actors user, service provider and the identity respectively attribute provider have been constantly increased. In the federated structure, demanded trust is at a culminating point with the topmost number and significance of trust requirements. However, these trust requirements are not evenly distributed on the actors to have an equally entitled setting. In particular, the user is in a discriminated trust position.

From the user's perspective, the highest number of trust requirements in the other actors are expected. Therefore, the dependency of the user is crucially beyond the reliance of the service provider and the identity or attribute provider. The trust position of the identity provider stands in contrast with the user. The identity provider receives the most trust as a trustee. At the same time, the identity provider has the least dependencies towards the other actors. Thus, the traditional identity management models are characterised by a disparate trust distribution with a disadvantageous position for the user. The user and the service provider are highly dependent on the identity provider. In the opposite direction, the identity provider has only a few subjections.

With the invention of the blockchain-based self-sovereign identity model, the overall amount of trust prerequisites between the actors have decreased. This is particularly relevant in the complex arbitrary aggregated and side-by-side integration pattern. Moreover, this situation marks a turning point compared to the trust development in the traditional models. The decentralised identity provider is transparent and publicly verifiable for the user and the service provider. Therefore, no trust is required.

Additionally, the original identity provider is reduced in its functionality to a mere attribute provider for issuing attestations. This reduction and the decoupling of the identifier generation from the attribute issuance enables a trust-enhancing attribute aggregation. In particular, the following two important changes in trust requirements are of special interest.

1. Trust by the user and the service provider in accurate credential management ($T2a$) and proper authentication ($T3a$) by the identity provider is eliminated.

2. Trust by the user and the service provider in attribute management ($T4a$ and $T4b$) is decreased by applying trust-enhancing attribute aggregation to combine the same attribute that is issued by distinct attribute providers.

Notwithstanding that the blockchain-based self-sovereign identity scheme reduces the demanded trust in identity management, a diverging trust distribution

between the user, the service provider and the identity, respectively, attribute provider still exists.

3.8 Summary

In this chapter, we initially described the vital trust domains privacy, credential management, authentication and attribute management. For each of these domains, we listed respective trust requirements that exist between the user, service provider and attribute provider. Subsequently, we presented a way to structure patterns within the blockchain-based self-sovereign identity setting and a trust evaluation scheme according to the requirements based on a limited and absolute trust level. By reference to these preliminaries, we defined the patterns of bilateral integration, multiple aggregated integration, multiple side-by-side integration, multiple service provider integration, as well as arbitrary aggregated, and side-by-side integration. In general, we analysed that the decentralised identity provider eliminates the trust requirements according to proper authentication and credential management. At the same time, the original identity provider is reduced to a pure attribute provider. These characteristics are a significant advantage compared to the traditional isolated, centralised and federated identity management model. Furthermore, we identified that the blockchain-based self-sovereign identity scheme in the multiple aggregated integration as well as the arbitrary aggregated and side-by-side integration pattern enables a trust-enhancing attribute aggregation approach that reduces the required trust into a single attribute provider. Basically, the same attribute is combined from several attribute providers to decrease the dependency on a specific attribute provider. This opportunity enables a further decrease of trust requirements in identity management. In particular, the influence of the attribute provider as the remaining trusted third party can be restricted by applying trust-enhancing attribute aggregation. The reasonable adoption of attribute aggregation requires attribute assurance trust modelling. We take the reduction of trust in the attribute provider as further motivation to investigate this type of trust model in the subsequent chapters.

4 Structure and Assessment of Trust Models in Attribute Assurance

This chapter is dedicated to the analysis of trust models in the domain of attribute assurance [23]. Initially, we examine the structure and composition of these trust models. Thereby, we outline common elements, differentiating factors of trust patterns in other domains and classification criteria [26] to create a meta-framework. Based on the framework, we derive central characteristics of a trust schema. Additionally, we study security objectives and attacks against them.

Hereinafter, we present and evaluate a set of assessment strategies covering classification [26], conceptual analysis, practical investigation and simulation to drive understanding about their virtues and limitations.

4.1 Motivation and Related Work

In the previous chapter, we delineated the strong dependency of the service provider and the user towards the identity provider based on trust requirements. Attribute management is one of the most critical domains of trust between these actors. The self-sovereign identity paradigm does not only significantly restrict the power of the traditional identity provider as a trusted third party, but also reduces it to a mere attribute provider. However, the attribute provider remains as a potential last central authority in the identity management setting. The reliance on a single attribute provider can be restrained by applying trust-enhancing attribute aggregation. Despite that, regular and trust-enhancing attribute usage requires trust by the service provider and the user. The subjectiveness of trust for different entities leads to a formalised consideration of trust in models. Therefore, manifold trust schemes have been proposed by using distinct notations [42]. Besides that, practically implemented identity management schemes lead to the creation of individual attribute assurance trust models. An example in this area is Public Key Infrastructures (PKI) based on X.509 [61]. A detailed investigation of these trust models is relevant to understand better implications as well as the advantages and disadvantages of different schemes.

Additionally, knowledge about important characteristics and security related attacks improve the definition of new trust models. Furthermore, the definition of new patterns must consider all appropriate components or may benefit from existing schemes. Thereby, an evaluation of different assessment strategies lay the foundation for reviewing and categorisation of trust patterns.

Related work exists in a multitude of areas. Taxonomies and surveys are one of the relevant research domains. In 2000, Grandison and Sloman [41] published a survey about trust in internet applications that outlines trust categories, classifications and trust management solutions. In 2005, Sabater and Sierra [16] studied computational trust and reputation models by distinguishing cognitive and game-theoretic approaches. Besides that, Ruohomaa and Kutvonen [75] reviewed trust management frameworks. In 2007, Jøsang et al. [42] published a comprehensive survey of trust and reputation systems for online service provision. Additionally, Yan et al. [76] focused on the Internet of Things when examining trust management approaches. Subsequently, in 2015, Cho et al. [77] researched a study about generic trust modelling, the concept of trust and its fundamental factors. In addition to surveys, trust algebras and calculi is a related research field. In 1999, Jøsang [78] created an algebra to assess trust in certification chains for communication between peers. Furthermore, Yang et al. [79] also proposed a foundational trust algebra to evaluate trust and its propagation in the communication domain. Huang and Nicol [80] defined a formal semantics based calculus to reflect trust relationships and derive trust flows. Furthermore, Ries et al. [81] introduced CertainLogic to include uncertainty into logical trust modelling for compound systems. Aldini [82] also modelled logical trust but for concurrent systems.

In addition to that, researchers worked on the comparison of specific components for general trust models and respective test beds. Trust modelling in dynamic and peer-to-peer networks are the centre of the work of Carbone et al. [83]. Kinateder et al. [84] focus on the evaluation of trust update algorithms. Fragkakis and Alexandris [85] study security and trust in the area of mobile agents. A general framework for trust models is proposed by Moyano et al. [86]. Haydar et al. [87] compare local, collective and global trust models. Jelenc et al. [88] evaluate trust models to improve reasoning for decisions. Additional research activities are covered in [89] [90]. Furthermore, Youssef et al. [91] proposed a test bed based on jade for the evaluation of trust algorithms in dynamic agent systems. The testart evaluation test bed of Fullam et al. [92] concentrates on reputation systems.

Besides research on the general trust setting, extensive studies that scrutinise trust in PKI systems prevail. Bakkali and Kaitouni [93] [94] proposed a logical reasoning calculus to determine trust in PKI from a theoretical point of view. Comparably, Haibo et al. [95] published a descriptive logic for trust domain modelling in this area. Furthermore, Huang and Nicol [96] defined a general calculus

and applied it to identity management. Risk and trust along certification paths lie in the centre of the conducted research. Additional studies review categories of PKIs and their features of trust distribution to build specific trust models. In this field, Maurer [97], Marchesini and Smith [98] and Henderson et al. [99] published articles. Complementary, Perlman [100], as well as Uahhabi and Bakkali [101], compared PKI concepts in a descriptive manner. Beyond a theoretical examination, Ulrich et al. [102] studied an existing representation of the OpenPGP [103] web of trust for its structure and characteristics. The authors practically investigated an instance of the web of trust network and analysed graph properties as well as security characteristics. Furthermore, Alexopolous et al. [104] examined the advantages of applying blockchain to trust management in authentication. The researchers also described security attacks and associated defence strategies.

Overall, previous work target on the one side general trust models, their evaluation and testing. On the other side, related studies concentrate particularly on dynamic agent systems, peer-to-peer environments and the PKI setting. The modelling in the PKI domain relates to identity management and the trust distribution in this realm. However, the constitution of trust in attribute assurance and potential trust-enhancing attribute aggregation are not adequately covered. Especially, existing models cannot properly reflect the proposals of AttributeTrust [105] as well as Thomas and Meinel's Logical Attribute Assurance Framework [46]. Additionally, specific research [104] solely considers identity trust or trust management in authentication systems where the trust context commonly refers to the public key to identity binding. In contrast, our trust model methodology covers the domain of attribute assurance and its specifics. In particular, we concentrate on trust in asserted attributes. Likewise, Gomi [106] proposed the separation between identity and attestation trust.

4.2 Trust Modelling in Attribute Assurance

To support the formalisation of trust modelling in attribute assurance, we outline common elements of trust models and describe differentiating factors outside the realm of attribute assurance. Subsequently, we present our meta-framework, security objectives and attacks as well as desirable properties of a trust model.

4.2.1 Common Elements of Trust Models

Trust models exist in various computer science domains. For instance, trust patterns are researched in agent systems [107], web site ranking [108] and in the Internet of Things [76]. Nonetheless, there exist common elements that are also aligned with the attribute assurance domain. Firstly, entities are a core compo-

nent of trust schemes. Entities act as trustors and trustees. They rely on each other in the situations for that the corresponding trust model has been built. Entities can comprise persons, organisations as well as further objects. Additionally, the relationships between individuals are important. These associations reflect an interaction or dependency between the different entities. Furthermore, usually, trust emanates from one object to the other individual. These relationships found the basis for determining trust in each other by a trust evaluation function. This function defines the composition of trust and its assessment in a given situation. The emphasis of distinct trust aspects and the mathematical representation is also achieved by the shape of the trust function. The outcome of the calculation is used to proceed with an interaction or to discontinue the relationship.

4.2.2 Distinct Factors towards other Domains

In the previous section, we described common components of trust models in computer science. In contrast, there exist distinct factors, as well. We elaborate on the direct feedback and trust ageing as differences towards trust schemes apart from attribute assurance.

4.2.2.1 No Direct Feedback

Trust and reputation are interwoven components that can be hardly separated [42]. Thus, trust schemes can incorporate reputational factors. Reputation considers previous experience between the entities [109]. In a specific scenario, entities interact and communicate with each other. After the interaction is finished, the trusting peer classifies the transaction. The entity provides feedback for the communication event. The feedback can be either positive or negative. Different gradients within these categories are possible. Positive feedback increases trust, whereas negative feedback decisions decrease trust in the interaction partner. The change of trust happens in a timely manner to incorporate the new trust rating in the selection of the subsequent communication partner.

The peer-to-peer file sharing scenario is an example where reputation-based trust models are applied in the domain of agent systems [110]. Within this use case, nodes exchange files upon request. A node trusts another entity in the network to deliver the requested file. Especially, the file should be fully usable and not corrupted. When the file transfer has been completed, it can be directly tested by the receiving node for correctness. If the requested file has been obtained as a working copy, the node grants positive feedback. Otherwise, negative feedback is logged for the sending entity. The stored reputation for all nodes influences the decision for choosing the sending node on the next file transfer. A direct feedback judgement after receiving the file is possible.

However, in attribute assurance, an immediate evaluation of the received attribute with regard to correctness is hardly possible. A service provider, as trusting entity, receives a set of attributes from a user. The attributes can be validated on a structural and superficial level. For instance, the name of a person might be verified that it does not include any numbers. Aside from that, conclusive validation of the name is not directly possible. Finally, the correctness might be determined when an ordered product is returned to the sender due to failed delivery. Thus, there is not generally direct feedback possible in attribute assurance. It depends on the shape of the attribute.

4.2.2.2 No Trust Ageing

Trust patterns that rely on experience can include elements of trust ageing [42]. If this scheme is applied, older interactions contribute less to the trust rating of an entity. In contrast, recent interactions have a higher impact on the trustworthiness. The existence of time decay of trust is the rationale behind it. Trust fades away over time because if a fact holds true in the past, it might not be the case that this fact holds true later on.

In the realm of online marketplace evaluation systems, the communication partners, for instance, the buyer and the seller, can rate each other for their service quality [111]. Amazon [112] uses a five-star measurement approach. Thereby, new appraisals are ordered at the top, and older ratings are moved to the end of the list. Thus, the reviewer can directly analyse the most recent feedback.

In attribute assurance, trust ageing is not formally incorporated in a trust model. However, it can be practically addressed by integrating validity periods into the trust model implementation. Besides that, revocation mechanisms might be applied to invalidate an attestation before the validity period expires.

4.2.3 Classification Criteria

As classification criteria, we determine trust scale, trust applicability, attribute aggregation, trust composition and centralisation of trust as major characteristics. The peculiarities form the foundation for the assessment approach of a taxonomy as one of the evaluation approaches.

4.2.3.1 Trust Scale

The trust scale indicates the different values of trustworthiness for an attribute. The values of the scale serve as the foundation for the final trust decision. We distinguish the discrete and continuous scale as foundational categories. The values have an order for both types of trust scale. Nonetheless, a discrete trust scale has

a limited number of values that are finite. The binary scale is a special case of the discrete scale having only two values. These peculiarities solely indicate trust and no trust. The binary scale is the most coarse-grained classification of trustworthiness. In contrast, the continuous trust scale applies an infinite quantity of trust nuances. It enables a very fine-grained trust classification and decision.

4.2.3.2 Trust Applicability

The trust applicability differentiates two dimensions. On the one side, it relates to the trust rating, and on the other side, it refers to the acceptance threshold. The trust rating is the opinion about the trustworthiness of an attribute issuer. The acceptance rating refers to the threshold on the trust scale when an attribute is considered as trustworthy. For both aspects, we differentiate the values predefined or individual. The term predefined indicates a globally predefined value that is the same for all entities. In contrast, an individual value is not globally alike for all parties, but each entity is able to apply an individual value.

4.2.3.3 Attribute Aggregation

Attribute aggregation describes the usage of attributes from several attribute providers. We differentiate the values completing, trust-enhancing, and none. Completing attribute aggregation refers to the usage of several attribute providers to achieve a comprehensive set of required characteristics [113]. A single attribute provider is not able to deliver all demanded attributes for a service provider. Thus, the properties of several providers are merged. Trust-enhancing attribute aggregation defines the accumulated usage of the same property. The trustworthiness of each provider is aggregated to receive an higher assurance. In case no aggregation approach is applied, we use the value none.

4.2.3.4 Trust Composition

Trust composition refers to the constitution of the trust value. We differentiate a simple or structured composition. A simple trust character derives the value from a single factor. On the contrary, a structured approach composes the trust value from a multitude of elements, for instance, by mathematical aggregation.

4.2.3.5 Centralisation of Trust

The centralisation of trust relates to a centralised or decentralised originating of trust in the attribute of an identity. Usually, a centralised root of trust is reflected by one or more trusted third parties. These central authorities cannot be avoided,

or the impact cannot be restricted by a peer. However, if the trust does not originate from a trusted third party, it is decentralised.

4.2.4 A Meta-Framework

We propose a meta-framework to depict trust models in attribute assurance. The framework utilises a graph-based model to depict entities and relations. Additionally, we integrate calculations for the trust decision process. Our framework contains the attestation and trust network as well as the trust decision process. Related research work exerts a graph network [105] or an algebra respectively a calculus [96]. A directed graph naturally reflects trustors, trustees, and their relationships, whereas a calculus can make the trust determination process transparent. Besides that, the framework is focused on the model character of trust and does not consider derived implementation schemes. In particular, we assume the existence of cryptographic solutions to secure communication between the nodes. Additionally, signatures are applied to determine the origin and authenticity of messages.

4.2.4.1 Attestation Network

Identity providers or attribute providers issue attestations about properties of users. These characteristics are transferred to service providers or any other relying party. In a PKI system, the issuing entities are called a certificate authority. A certificate authority attests the binding of the public key to characteristics. In this sense, the public key ties the distinguished name to an entity. An identity is comprised of properties, e.g. first name, last name and address, that allow the identification of a person. In the context of Pretty Good Privacy (PGP) [114], every user can act as an attestation authority and confirm such a binding. The association exists between a public key and an email address. In addition to that, characteristics are named claims in the self-sovereign identity ecosystem. An asserted claim is referenced as a verifiable claim or credential [69]. Comparable to PGP, every peer can assert claims. A user can also issue intrinsically a self-attested attribute. These attribute attestations build relations between the nodes in an attribute assurance setting.

Definition 4.1 (Attestation network)**.** *An attestation network AN is a directed graph* $AN = (E, A)$ *that expresses attribute attestations as relations A between the nodes E whereas:*

- *Nodes E represent all entities in the network, e.g. identity, attribute and service provider, certificate authorities or users.*

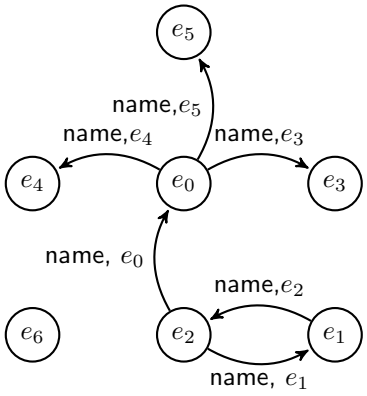


Figure 4.1: Sample attestation network

- *Attestations A reflect asserted attributes by one entity to another. An attestation $a \in A$ is a relation annotated by an attribute tuple $\langle$attribute class, attribute value$\rangle$.*

Concerning the nodes, an entity can be called an attribute provider if it issues at least one attestation. User refers to a node that applies the received attestations. Commonly, a relying party get attestations transferred and trusts in its authenticity. A service provider is usually acting as a relying party. Despite that, this entity can also issue or receive attestations. An attribute attestation relation comprises the two elements attribute class and attribute value. The class specifies a category of a property. For instance, the category can be a name, email address or the postal address. The value reflects the actual value of the asserted property. The attribute class represents the context of the attestation. An attribute provider might be capable of attesting an email address because it has the respective verification procedures implemented. In contrast, the attribute provider might not be able to attest verified names or addresses.

In Figure 4.1 a sample attestation network is shown. The nodes in the structure assert their names. The entity e_0 issues the most attestations. This node can be seen as a certificate authority in a PKI environment. On the contrary, the individuals e_3, e_4 and e_5 reflect normal users that obtained a name attestation. Additionally, the nodes e_1 and e_2 attest each other their names. This constellation paradigmatically reflects a web of trust. The entity e_6 neither obtains an attribute assertion nor provides one to another entity. However, it is part of the network and may use attestations at a future point in time.

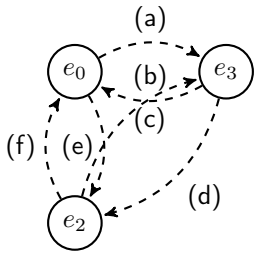


Figure 4.2: Trust views

4.2.4.2 Trust Network

The attribute attestations in an attestation network reflect an interaction. To assess a transition towards mutual trust, we investigate the potential trust situations in a sample setting with regard to the attestations. Fig. 4.2 shows the paradigmatic arrangement as a reduction from Fig. 4.1. It includes an attestation issuer (e_0), an entity that receives the assertion (e_3) and a relying party (e_2). There are several trust dependencies between the nodes with reference to the issued property.

Manifold trust associations consider the agreed use of attribute data and adherence to established processes. This relates to dependencies between e_3 and e_0 (b), mutually among e_2 and e_3 (c, d) as well as within e_0 and e_2 (e). Furthermore, e_0 expect from e_3 that attribute verification processes are not circumvented. However, these associations do not represent the major trust relation in an attribute assurance trust model.

The principal trust dependency occurs when the attribute assertions are shown to a relying party. This link is reflected between e_2 and e_0 (f). The relying party needs to trust the attestation issuer that their processes are working to provide authentic attributes. Additionally, it should not be possible to by-pass the verification processes. This significant trust relation is captured in the trust network.

Definition 4.2 (Trust network)**.** *A trust network TN is a directed graph $TN = (E, R)$ that expresses trust relations R between the nodes E whereas:*

- *Nodes E reflect all relying parties, e.g. service providers, identity or attribute providers and users.*
- *Trust relations R portray the dependency that a trustor relies on the correctness and validity of an attribute that is attested by a trustee. A trust link $r \in R$ is a relation annotated by the tuple $\langle$attribute class, trust rating$\rangle$.*

A relying party is an entity in the trust network that relies on the attribute attestations. Usually, a service provider is a relying party because it depends on

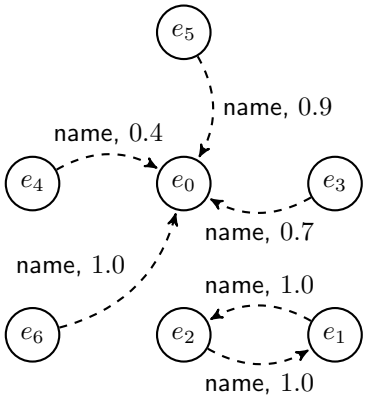


Figure 4.3: Sample trust network

the properties that are issued by an attribute provider. Nonetheless, any entity, be it the user or even the attribute provider, can be a relying party. Besides that, the trust relation is specified as a pair of the attribute class and a trust rating. The attribute class matches the respective category in the associations of the attestation network. The trust rating is an element that belongs to the trust scale (cf. Chapter 4.2.3.1) of the model. The trust scale encompasses all individual trust ratings that can specify the level of trust between two entities. These trust ratings are an ordered set to enable comparisons of trustworthiness.

Fig. 4.3 illustrates an example of a trust network that is aligned with the sample attestation network. Trust ratings lie in the range from 0 to 1. The entities e_3, e_4, e_5 and e_6 trust in different degrees the node e_0. The entity e_0 can be seen as a certificate authority in PKI schemes. Additionally, the nodes e_1 and e_2 trust each other and reflect a peer-to-peer attestation scheme.

Comparing the attestation and the trust mesh, we can investigate relationships between the two graphs. In the attestation network the different entities issue attestations, obtain attributes or do not interact and behave as a quiet observer. The same nodes are also a member of the trust network, because they can trust other nodes or be a trustee. In another case, a node may neither receive trust nor trust others. Therefore, studying the nodes of the networks, we can conclude that the set of nodes in both structures are the same.

Examining the relations in the attestation and the trust network there is less conformity. An attestation expresses the confirmation that an attribute is authentic. A trust relation states a subjective trust appraisal from one entity to another. Proceeding from Fig. 4.1, there is one attestation between node e_0 and e_3. Fig. 4.4 illustrates potential trust situations that emanate from the assertion. An extended listing is available in the Appendix A.1. There can be no trust between the different entities. Thus, issued assertions cannot be used in this setting. Besides

that, solely trust exists between the node e_3 and e_0. As a result, entity e_0 is not able to consume the service of node e_2. Nonetheless, it is very likely that node e_3 obtains attestations from entity e_0 to interact with the relying party e_2 because e_2 trusts issuer e_0. However, any other trust constellation might be appropriate in a general setting.

(a) No trust **(b)** Trust from receiver to issuer

Figure 4.4: Extract of potential trust situations

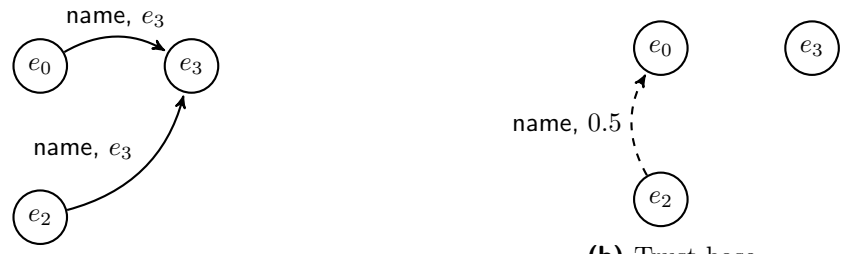
In conclusion, from a superficial perspective, an entity that issues a large number of attestations is also likely to be trusted in the trust network. In contrast, there is no evidence that an entity that does not issue any attestation is not trusted at all. However, besides a shallow trust indication, there is no dedicated dependency between the two relations in a non-restricted transformation from the attestation to a trust network. A bounded trust model may apply a very limited trust decision process, that restricts the options space for transforming the attestation to the trust network.

4.2.4.3 Trust Decision

The attestation and the trust network build the foundation for the relying party to conduct a trust decision. The trust decision is the final judgement to accept or reject an attribute for further processing.

Definition 4.3 (Trust decision)**.** *A trust decision D is a self-evaluating tuple $\langle T, B, V, S\rangle$ that results in a binary trust or no trust outcome. A trust decision D_e is made from a perspective of an entity $e \in E$ of the trust network. The tuple components are:*

- *Trust function T calculates the trust value*
- *Trust base B represents trust ratings towards other entities*
- *Attestation base V encompasses the attribute assertions*
- *Acceptance rules S specifies the acceptance respectively rejection trust condition for an attribute*



(a) Attestation base **(b)** Trust base

Figure 4.5: Sample attestation and trust base

The trust function T is the central component of the trust decision process that is executed by an entity e. It computes the overall trust score of an attribute. The trust base B and the attestation base V are the foundation of the calculation. A reduction of the trust network to a partial graph is reflected by the trust base. Only the trust in the attestation issuers of the attribute is required. Comparable, the attestation base is part of the attestation network. Therefore, it is also a partial graph solely representing the attribute that should be evaluated and its assertions. Fig. 4.5 depicts samples of both components. Based on this information, the trust function computes a trust value that is matched to the attribute specific acceptance rules S. In case the trust score exceeds the threshold of the rule, the attribute is accepted as authentic and processed further by entity e. The node comes to the trust conclusion when running the trust decision process. On the contrary, if the threshold is violated by the evaluated trust level, the attribute is rejected by entity e. The trust decision process finishes with the result of no trust.

4.2.5 Characteristics

Taking the described framework as a foundation, we can devise particular characteristics of the attestation and trust network as well as the trust decision process. These properties are outlined in the following paragraphs.

4.2.5.1 Degree of Centralisation

The degree of centralisation grades the density of relations in a network towards several entities. In the attestation network, our focus lies on the attestation issuing entities. The nodes where the relations originate are of particular interest. In contrast, in the trust network we focus on the entities that receive trust values. In this environment, we consider the nodes that are the target of the relations. In case, a low quantity of nodes compared to the overall amount of entities issue assertions, the trust model tends to be of centralised nature. The same applies to the trust

network, considering the nodes that are the target of trust ratings. The degree of centralisation indicator approaches the value 1. Concerning a development in the opposite direction, the number of trusted nodes come close to the overall quantity of nodes. Thus, the degree of centralisation converges to 0. Referring to the attestation network, if the number of attestation issuers approaches the overall number of entities, the type of the network falls in the category of a web of trust model. In this type, the attestation issuers are not seen as central authorities. We define the degree of centralisation metric for both networks as follows.

Definition 4.4 (Degree of centralisation (AN)). *The degree of centralisation (DoC) in the attestation network is defined as* $DoC_{AN} = 1 - \frac{|P|}{|E|}$.

Definition 4.5 (Degree of centralisation (TN)). *The degree of centralisation (DoC) in the trust network is specified as* $DoC_{TN} = 1 - \frac{|G|}{|E|}$.

The set P refers to all attribute providers in the attestation network and comprise all entities that issue at least one attestation. The set G references the trust receiving entities in the trust network. It encompasses all nodes that obtain at least one trust rating from another node.

4.2.5.2 Degree of Interconnection

The degree of interconnection measures the separation or, on the opposite, the interconnection of the attestation and the trust network. We use the number of separated subgraphs (H) as a metric. An isolated subgraph is a partial graph of the network that has no connection to other nodes in the mesh. This indicator is aligned to the strongly connected component measurement of Ulrich et al. [102]. If the complete network is reflected by solely one graph, the whole mesh is interconnected. Otherwise, the network is fragmented. A subgraph reflects an autonomous community that rely on each other for either attestations or trust ratings depending on the investigated network. If the degree of interconnection is approximate to 1, the network is highly interconnected. On the contrary, if the value is close to 0 the mesh is disconnected. We determine the degree of interconnection for both networks in the following manner.

Definition 4.6 (Degree of interconnection). *The degree of interconnection (DoI) is specified by the metric* $DoI = 1 - \frac{1-|H|}{|E|}$.

4.2.5.3 Issued and Received Attestations

As the attestation network is a directed graph, entity-specific relationships can be easily evaluated. The number of issued attestations from a specific node represents

a level of activity. The higher the number of attestations, the more does the node contribute to the structure of the network. In a comparable manner, the quantity of received attestations is an indicator of the shape of an entity. A node with a large and diverse range of attestations is a pronounced identity.

Definition 4.7 (Received attestations)**.** *The number of received attestations (RA) for an entity e is defined as* $RA_e = |\bigcup_i (e_i, e) \in A|$.

Definition 4.8 (Issued attestations)**.** *The number of issued attestations (IA) for an entity e is defined as* $IA_e = |\bigcup_i (e, e_i) \in A|$.

4.2.5.4 Attestations for Acceptance

The attestations for acceptance metric target the analysis of the trust decision process. It specifies the minimal required number of attestations from distinct providers for the acceptance of an attribute. The lower the metric the fewer attestation issuers are required to be involved for a successful outcome of the decision process. The higher the number, the more attestations must be obtained. Usually, there is a dependency on the trust rating of an entity for acceptance. The acceptance is based on the default rules of the trust model. Therefore, we define it as the minimum number of required attestations.

Definition 4.9 (Attestations for acceptance)**.** *The attestations for acceptance (AfA) metric is defined as* $AfA = \min(V)$ *with* $D\langle T, B, V, S\rangle$ *= trusted.*

4.2.5.5 Trust for Acceptance

The evaluation of the trust decision process is also the intent of the trust for acceptance measurement. This metric defines the minimal required trust rating for one attestation issuer to contribute to the calculated trust value of an attribute. Therefore, the trust scale of a model must be normalised into the interval $[0, 1]$ where 0 represents no trust and 1 reflects the most trust. For instance, discrete scales can be transferred by equal parts into the frame.

Definition 4.10 (Trust for acceptance)**.** *The trust for acceptance (TfA) metric is* $TfA = \min(r)$ *with* $\exists e_i, e_j \in E : V = \{(e_i, e_j)\langle a, r\rangle\}$ *and* $T(B, V) > 0$.

4.2.6 Security Objectives and Attacks

A trust model is a security-relevant feature. In this section, we investigate security objectives and related attacks to compromise it.

Attack	**Security Objective**	**Origin**	**Target**	**Affected Component**
Censorship	Availability	External, AP	AP	Attestation network
Denial of service	Availability	External	AP	Attestation network
Attribute forgery	Integrity	User	AP	Trust decision
Rogue attribute provider	Integrity	AP	RP	Trust decision
Stale information	Integrity	User	AP	Trust decision
Trust base manipulation	Integrity	User	RP	Trust network

Table 4.1: Security attacks on attribute assurance trust models

4.2.6.1 Security Objectives

In information security, the triad of availability, integrity and confidentiality represents the major security objectives [3]. Derived from the general goal of availability, it refers, in attribute assurance, to two aspects. On the one side, the user must be able to retrieve attribute attestations when required. Thus, the attestation issuer is available and provides service. Furthermore, any relying party must have the ability to verify the attestation in the sense that the issuer can be validated at the point in time when it is needed. The objective of integrity is especially important in attribute assurance. The attributes of an identity must reflect reality. Furthermore, in case the attested attribute value becomes invalid, the property must be revoked timely as well. Besides that, during the transmission of the characteristic between different entities, the value should not be illegitimately compromised. The target of confidentiality refers to the protection of attestation's content from unauthorised disclosure. Attributes might be personal identifiable information and therefore, must be only revealed in a consented manner to permitted individuals. The security goals availability and integrity can be analysed on the model and implementation level. In contrast, confidentiality refers to the implementation layer and is not reflected on the level of the trust scheme. As we concentrate on the trust model, we will study attacks against availability and integrity.

4.2.6.2 Attacks

In the following paragraphs, we describe the attacks censorship, denial of service, attribute forgery, rogue attribute provider, stale information and trust base manipulation. Table 4.1 illustrates an overview of the attacks stating the related security objective, the originating entity, the targeted object and the affected respectively manipulated trust model component. Thereby, we consider the user, service provider and attribute provider as relevant entities. Fig. 4.6 to 4.11 schematically illustrate the attacks. A double circle around an entity represents the attacked node.

4.2.6.2.1 Censorship The censorship attack targets the exclusion of an entity from the service of another node. It was proposed by Alexopolous et al. [104], and we transfer it to the attribute assurance domain. The target of the attack is an attribute provider. The node refuses to issue attestations for another dedicated entity. Thus, the affected trust model component is the relations in the attestation network. This behaviour either originates directly from the attribute provider or is externally enforced upon the provider. As a result, the user is not able to obtain required attributes and might be excluded from any service of relying parties where these attributes are required. In the worse case, it could lead to complete isolation

of the node in case the attribute provider holds a powerful centralised position in the attestation network of the trust model. Any service provider that relies on the attestation provider cannot serve the censored user. The attack setting is depicted in Fig. 4.6. The node e_0 issues attestations to nodes e_1 and e_2, but no assertion is targeted towards the censored node e_3.

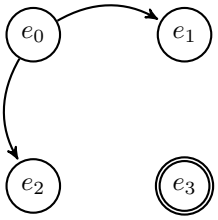


Figure 4.6: Censorship

A counter strategy for this attack aims at the increase of attestation issuers to prevent only a few powerful attribute providers as central authorities. If there is a large number of generally trusted attestation issuers, it is easier for a user to circumvent the censorship attack. The censored node uses the service of another party. Therefore, the attack would require to target several attestation issuers. The effort to execute the attack would significantly increase.

4.2.6.2.2 Denial of Service The denial of service attack is a well-known and popular attack category in various fields of computer science [115]. Generally, the attack tries to impede the regular service of an entity and has the objective to completely deny it. Conveyed to the attribute assurance domain, the attack targets an attribute provider to prevent the correct functioning of its property attestation service. The attack vector targets the attestation network of a trust model. Ultimately, users are not able to retrieve attribute attestations from an attacked attribute provider. Furthermore, the reputation of the attestation issuer also decreases due to the non-functioning of its service. Therefore, the attack is only externally motivated outside the attribute provider. In contrast to the censorship attack, a large number of users and also relying parties are affected. The attack does not only concentrate on a single node but influence the complete network depending on the strength of the attribute provider's position. The attack is illustrated in Fig. 4.7. Node e_0 is the target and cannot provide attestation service to the other nodes anymore.

To reduce the impact of a denial of service attack on a prominent attestation issuer, a multitude of comparable entities with a similar trust posture in the mesh are required. Thus, users and relying parties can consume the attestation service

from another attribute provider. The impact of the denial of service attack for the network and surrounding entities is reduced.

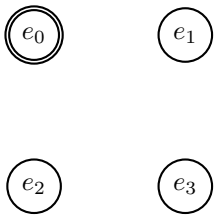


Figure 4.7: Denial of service

4.2.6.2.3 Attribute Forgery The attribute forgery attack targets the authenticity of an attested property. The attestation issuer is tricked into asserting a wrong characteristic by a user. An entity tries to consume a service from a relying party that has as a prerequisite a specific attribute. Thereby, the user circumvents the implemented property verification processes of the attestation issuer. As a result, the entity has a wrongly attested attribute that is presented to the service provider. Subsequently, the entity can illegitimately obtain the respective service. Therefore, a harmful impact exists on the service provider. Additionally, the reputation of the attribute provider decreases. Fig. 4.8 shows the attack setting. The node e_0 is the target. It issues a false claim to node e_3.

A counter-strategy to minimise the attack impact is the adaptation of the trust decision process. In case the trust function and the acceptance rules do not solely rely on one attribute provider to accept an attribute, the barrier to execute the attack is raised. In particular, the user needs to deceive several attribute providers which is more challenging than to delude only one entity.

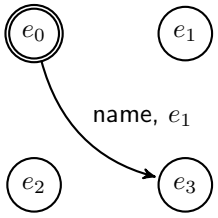


Figure 4.8: Attribute forgery

4.2.6.2.4 Rogue Attribute Provider The rogue attribute provider attack aims at setting up a new attestation issuer by an attacker. This entity's purpose is to assert wrongly attributes to give the attacker an advantage. A service should be

consumed in an unauthorised manner. Such an attack is possible in case the trust model implements a generic trust function that considers any attribute provider as trusted and does not only trust specific issuers.

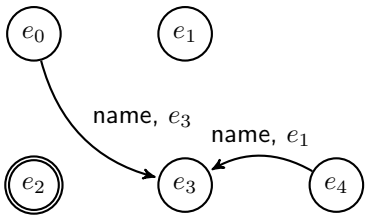


Figure 4.9: Rogue attribute provider

In a peer-to-peer trust scheme, an impostor may create additional dependent entities that span an own attestation and trust network to create an impression of legitimacy. The attack is depicted in Fig. 4.9 and is comparable to the Sybil attack [57]. The targeted entity is node e_2. The node e_4 is the rogue attribute issuer.

To mitigate this attack, the trust decision process needs to consider well-known respectively highly reputable entities as attribute providers. This approach reduces the likelihood to implant a rogue attestation issuer and decreases the influence of this entity.

4.2.6.2.5 Stale Information Executing the stale information attack, an impostor uses obsolete information to gain an advantage. This vector was proposed by Alexopolous et al. [104] for trust management in authentication systems. In attribute assurance, an attacker still uses attribute assertions that are already expired. To achieve this, the impostor thwarts the revocation mechanism of the attribute provider.

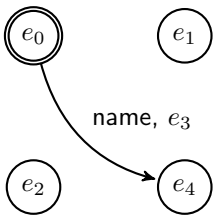


Figure 4.10: Stale information

As an outcome, unauthorised service consumption at a relying party is enabled. Additionally, the reputation of the attestation issuer declines. The scenario is illustrated in Fig. 4.10. The node e_3 has renamed to e_4. Though, the claim still carries the old identifier.

Comparable to the counter strategy of the attribute forgery attack, the impact of stale information is mitigated if the trust decision process considers several attribute providers. Relying on one attestation issuer creates complete exposure to the attack.

4.2.6.2.6 Trust Base Manipulation The trust base manipulation aims at changing trust information for a relying party. The trust base contains the trust relations that are relevant for the trust decision process. An attacker manipulates the trust ratings for a certain attribute provider or incorporates new attestation issuers with a high trust rating. Subsequently, an actually illegitimately acting user might be able to consume a service by having only attribute attestations from manipulated providers in the trust base. The attack targets the trust network. A comparable threat is the root certificate injection into the trust store of an operating system.

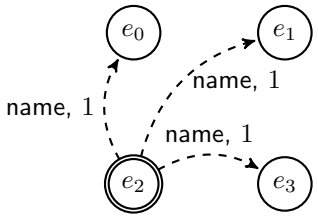


Figure 4.11: Trust base manipulation

The attack scenario is outlined in Fig. 4.11. The node e_2 was attacked to add and increase trust ratings towards the other entities. The trust base is a component of the trust model. However, counter-strategies exist solely on the implementation level. Hardening of the trust store is a sample measure. As we concentrate on the trust model, we do not elaborate further on these defence strategies.

4.2.7 Properties of a Secure Trust Model

The main elements of a trust model are the attestation network, the trust mesh and the components of the trust decision process. Attribute attestations between the entities are the core component of the attestation network. However, the attestation graph only represents interactions between the nodes that are aligned to its preferences. These connections emerge over time. The attestation network is an essential part of the trust model. Nonetheless, there is no space for modelling to shape the overall trust model in a certain direction. Besides that, the trust network encompasses trust ratings between the entities. These values are the subjective opinions from one node to another. The trust scale is determined by the model.

However, the rating originates from the entities. Additionally, different scales can be normalised or transferred to each other. Thus, comparable to the attestation network, the influence of modelling activities is limited. Turning towards the trust decision process, the trust base and attestation base are solely subgraphs of the respective networks. Therefore, they have the same limitations for the influence on the trust model as the complete networks.

In contrast, the remaining elements of the decision process, the trust function and the acceptance rules are central to shape the overall trust model. Thereby, we can neglect the acceptance rules. They define thresholds for the computed value of the trust function to accept or reject an attribute. The acceptance rules are dependent on the modelling of the trust function. Specific manifestations can be normalised into the trust function. In conclusion, the trust function is the major component of a trust model to shape its characteristics.

From our view, a secure trust model is reduced to a secure trust function. An attack-resistant trust function must foster the security objectives availability and integrity. Thus, it must be resilient to the described security attacks. Additionally, the trust function and the overall trust model should be practically applicable and usable. The following characteristics are essential for a secure trust function.

1. The trust function ensures that the attribute is authentic. Therefore, it should consider and promote stronger trust ratings in the individual attestation issuers. This property counters attacks on integrity. In case relying parties want to accept attributes with a low assurance of authenticity, it can be realised via individual acceptance rules.

2. The trust function must have only a low dependency on a single attestation issuer. In case one provider is attacked or behaves malicious, no significant influence on the trust function exists. The higher the quantity of simultaneously considered attestation issuers the lower is the dependency to a single provider. This characteristic condemns attacks on availability and integrity.

These peculiarities of the trust functions can be evaluated with our defined characteristics that aim at the trust decision process. The trust for acceptance metric references the minimum demanded trust in an attestation issuer. Thus, it covers the first property. The second feature is supported by the attestations for acceptance measurement that defines the minimum number of distinct attestations to accept an attribute. The higher the actual characteristic of the metrics for a specific trust model, the higher is the security posture against the described attacks. In contrast, low values of the metrics indicate higher vulnerability.

4.3 Evaluation of Assessment Strategies

The categorisation and assessment of trust models in attribute assurance can be conducted on the basis of classification, conceptual and practical examination as well as simulation. Initially, we present a selection of sample trust models. Subsequently, we describe and evaluate assessment strategies. Additionally, we apply the taxonomy and the conceptual analysis of the selected sample trust patterns.

4.3.1 Sample Models

We selected examples of trust models to show the taxonomy and the conceptual analysis of the assessment strategies. Our choice reflects a diverse range of trust schemes. We pick the PKI based on X.509 [61] as a popular and widespread used representative in the area of chain of trust. Furthermore, PGP [114] is evaluated as a prominent example of a web of trust. Besides that, we selected Thomas and Meinel's Logic-based Assurance Framework [45] and the AttributeTrust [105] scheme as dedicated examples of attribute assurance trust modelling. The PKI based on X.509 PKI and PGP specify all components of a trust model. AttributeTrust does not elaborate on an attestation network. The Logic-based Assurance Framework defines the trust network, trust function and base as well as the acceptance rules. An overview is shown in Appendix 4.2. In the following subsections, we describe the foundational trust aspects of each scheme.

4.3.1.1 Public Key Infrastructure based on X.509

In PKI systems, asymmetric encryption identifies and authenticates entities. Central components of the model are certificates and their issuing certificate authorities. Additionally, revocation mechanisms exist to invalidate certificates. A standard for a PKI is X.509 [61]. The public key of a key pair can be the identifier of an individual. A certificate binds the public key to information that identifies the subject. For instance, this data can be the name for a person or a domain name for a web server or a computer.

Secure communication is the main purpose of certificates. Supported by the public key in the certificate, the identity of the communication partners can be verified, and an encrypted connection is established. The certificate encompasses a signature of the certificate authority to prove the binding between public key and properties. The cryptographic signature can be verified by any entity to validate the certificate. However, the verifier must trust the certificate authority to verify and sign the attributes properly. A certificate authority can issue certificates to intermediate authorities who issue certificates to the principal. Each authority

represents a trusted third party and must be trusted by a verifier to trust the certificate finally. A certificate reflects an attribute attestation in the PKI scheme.

4.3.1.2 Pretty Good Privacy

PGP [114] is a decentralised email address verification scheme that Zimmermann invented. Public key cryptography is used to identify and authenticate users and their email addresses. The public key is bound to the email address by cryptographic signatures. In contrast to the PKI based on X.509, any peer can sign the binding, and no dedicated certificate authority exists. Such a peer is called introducer. These signatures are attribute attestations in the PGP ecosystem. The PGP trust model [116] distinguishes trust in the introducer and trust in the public key to email binding. A PGP certificate comprises the public key, email address and owner as well as the cryptographic signature.

The trustworthiness of a certificate is classified in undefined, marginal and complete. The state undefined does not allow any conclusion about the trustworthiness. Marginal represents a medium state of trustworthiness. A certificate in status complete is fully trusted. This trust categorisation is subjective to a user and can be individually defined. The same applies for the differentiation of the introducer's trustworthiness. The introducer can be rated as full, marginal, untrustworthy or unknown. If a certificate has the defined number of fully or marginally trusted attestations, then the certificate is fully trusted. Per default, if there is only one fully or marginal trusted introducer, then the certificate is marginally trusted [116]. Otherwise, the certificate is not trusted.

4.3.1.3 Logic-based Assurance Framework

Identity federations enable the usage of several identity providers. Thomas et al. [46] discovered that attribute assurance models only allow to trust the identity provider completely or not at all. There is no fine-grained decision on attribute level possible. Thus, Thomas and Meinel [46] proposed a Logic-based Assurance Framework to specify trust on property and provider level. The trust model encompasses service and identity providers as well as further entities. Besides these actors, organisational trust levels, attributes, attribute types and attribute-based verification classes are included in the framework. The connections between the different elements are established by relationships. The identity provider has assigned an organisational trust level. Additionally, it can only attest attributes of a specific type and verification class. A relying entity, for instance, a user or a service provider, keeps an individual knowledge base to store identity provider and attribute related trust characteristics. Furthermore, rules for acceptance are pre-

served as well. Upon determining the trustworthiness of attributes, the knowledge database is evaluated.

4.3.1.4 AttributeTrust

Mohan and Blough [105] published the AttributeTrust framework to determine trust in the attributes of an identity based on a reputation system. Drawbacks of previously existing attribute assurance schemes served as motivation for the new trust model. Mohan and Blough listed the bundled issuance of several attributes as a dedicated disadvantage. The AttributeTrust framework is a directed graph where nodes reflect the actors and edges represent confidence paths. User, service provider and attribute providers are actors. The confidence paths indicate a trust relationship about the attribute assurance. Each path is weighted by a confidence value in the range between 0 and 1. New actors that join the network start with rating 0. The confidence value of a node is calculated based on the edges and their values leading to the node itself. A relying party determines trust in an attribute by evaluating all confidence paths between the entities until a defined depth. The reconciled score is compared to a threshold which leads to acceptance or rejection of a property.

4.3.2 Taxonomy

We present the approach of a taxonomy and apply it to the described sample models. Table 4.2 presents an overview of the examined sample models and their characteristics.

4.3.2.1 Approach

The classification of objects by characteristics leads to the creation of a taxonomy [117]. A taxonomy arranges these artefacts towards each other. This approach enables a coarse-grained categorisation. However, it can uncover blank spots, respectively research gaps in a study field. Clustering trust models in a taxonomy provides a high-level differentiation according to their core properties.

We have identified trust scale, trust applicability, attribute aggregation, trust composition and centralisation of trust as main characteristics of trust models in attribute assurance. These classification criteria are described in Section 4.2.3. Based on these properties, a general taxonomy to describe attribute assurance trust schemes is established. Furthermore, our proposed meta-framework supports a categorisation of trust models according to their components. A published trust model may devise components of the trust decision process. However, the attestation or trust network of an existing PKI system can be utilised.

Trust Model	Trust Scale	Trust Applicability	Attribute Aggregation	Trust Composition	Centralisation of Trust
PKI based on X.509	Binary	Predefined/ predefined	None	Simple	Central
PGP	Discrete	Individual/ individual	Trust-enhancing	Simple	Decentral
Logic-based Assurance Framework	Discrete	Individual/ individual	Completing	Structured	Central
AttributeTrust	Continuous	Individual/ individual	Completing	Simple	Central

Table 4.2: Characteristics of sample trust models in attribute assurance

As an advantage of the method, the taxonomy constitutes a simple overview of the trust model research field and can contribute to systematisation of knowledge. Beside that, focused research areas and non-studied fields are disclosed. On the contrary, a taxonomy only allows high-level insights. Detailed conclusions can hardly be investigated.

4.3.2.2 Practice of Approach

In the following paragraphs, we evaluate the sample models according to the classification criteria.

4.3.2.2.1 PKI based on X.509 The PKI trust model implements a discrete trust scale. Solely the values trusted and not trusted are differentiated. The service provider or user either trusts or does not trust the certificate authorities within the chain of trust. Additionally, the discrete trust scale is binary. With regards to trust applicability, the rating and the acceptance level are globally predefined for all entities. To participate in the PKI, the service provider and the user can only trust a certificate authority. Even the trusted certificate authority are shipped as a central trust store within the operating system and browsers [118]. Trusting a certificate authority leads to the acceptance of the issued certificates. A certain certificate authority has the same trust rating for all entities. Different service providers or users cannot individually define distinct trust ratings or acceptance levels.

The PKI trust model does not implement attribute aggregation techniques. A certificate may comprise several attributes. However, the certificate is issued from a single authority. There are neither attributes aggregated from different authorities nor a trust-enhancing methodology is applied. The composition of trust is simple. There is no structured composition of different factors within the trust model to obtain an overall trust score. The certificate authority must be trusted or is not trusted. Additionally, the PKI trust model is centralised because the certificate authorities are the trusted third parties responsible for proper attribute assurance.

4.3.2.2.2 Pretty Good Privacy The trust scale of PGP is discrete and not binary. The trust of a certificate is evaluated according to the levels complete, marginal and undefined. Concerning trust applicability, the rating and acceptance level are individual for all peers. Each user defines its trusted introducer independently and assigns them the trust rating. In case a certain introducer is not specifically considered, the default unknown value is applied. Besides that, the trust acceptance is characterised by the number of marginally and fully trusted

introducers. These quantities serve as thresholds for acknowledging the certificate. The user is able to set these values individually in its PGP client.

Furthermore, PGP applies the trust-enhancing attribute aggregation methodology. The signature of the different introducer support the certificate and represent the trust. The higher the number of signatures of category marginal and full trust, the higher is the trust into the certificate. Completing attribute aggregation is not applied based on the specific purpose as email address scheme. Additionally, we evaluate the trust composition as simple because there is no structured derivation of trust. The trust model solely differentiates marginally and fully trusted introducers. The PGP trust model is decentralised due to the non-existence of trusted third parties. Every peer can attest certificates of other peers and therefore, act as an introducer. Thus, no theoretical trust concentration on a minor number of central authorities exists.

4.3.2.2.3 Logic-based Assurance Framework The Logic-based Assurance Framework of Thomas and Meinel [46] applies a discrete trust scale. Depending on the rule set in the knowledge base, the organisational trust rating and the federation property of the identity provider, an attribute is trusted. Several trust levels of the identity provider that are inherited by the issued attribute, are differentiated. Therefore, no binary trust scale is used. We evaluate the category trust applicability with its sub domains rating and acceptance to individual. Under the assumption that each user or service provider maintains an independent knowledge base, the individual trust rating and acceptance holds true. The assurance framework uses a completing attribute aggregation strategy because the knowledge base is designed to include distinct identity providers and to specify trust in these entities and their provided attributes.

However, there is still a final decision about the trustworthiness on attribute level. Thus, no trust-enhancing attribute aggregation approach is applied. Furthermore, the trust in attributes still originates from a limited amount of entities as identity providers. These providers can be individually classified in the knowledge base. However, this setup still implies centralisation of trust towards these providers.

4.3.2.2.4 AttributeTrust The AttributeTrust [105] framework implements a continuous trust scale. The range of this trust scale lies between 0 and 1. The result of the trust function is obtained by calculating the product of the node's in-degree, and the average confidence value of all received confidence paths. In the category of trust applicability, both the rating and acceptance values are individual to all entities in the AttributeTrust network. Each entity can define respectively generate their own confidence paths. Furthermore, the acceptance of an attribute

at a certain confidence value is in the responsibility of the specific entity. Thus, the actor defines the acceptance threshold. Furthermore, the evaluation depth of the confidence path is also entity-specific.

AttributeTrust applies a completing attribute aggregation methodology because characteristics from distinct attribute providers can be forwarded to the service provider. There is no usage of a trust-enhancing attribute aggregation strategy. As trust is derived from the confidence paths, the trust composition is shaped in a simple manner. There is no structured composition based on different underlying trust factors. The AttributeTrust framework implements a centralised trust model because dedicated attribute providers are the origin of trust in the characteristics of an identity. There are no limiting measures.

4.3.3 Conceptual Analysis

We describe the conceptual analysis approach and provide advantages and disadvantages. Furthermore, we use the methodology to examine the sample models.

4.3.3.1 Approach

The conceptual analysis strategy is a theoretic approach to model, design and conclude on a trust model in attribute assurance. It is a paper-based approach to examine the main characteristics and deduce implications. For instance, for a trust model, it can be decided to belong to the chain of trust or web of trust category. Modelling approaches encompass graph schemes, formal logics or narrative descriptions. Huang and Nicol [96] created a formal logic to calculate trust theoretically. Uahhabi and Bakkali [101] compare PKI trust models textually.

With the conceptual analysis strategy, the attestation network, the trust mesh and the trust decision process can be investigated. The attestation and the trust network can be visualised as a diagram to show the main characteristics of the devised trust model. This method fosters the understanding of the reader. However, the illustrated nodes and relations in the graphs are intentionally selected. Therefore, the created diagram results in a dedicated vision of the researcher. The same applies to the network-based characteristics of the model. The metrics degree of centralisation, degree of interconnection and the issued, as well as the received attestations can be evaluated on the designed sample networks. Nonetheless, the expressiveness of the result is limited. On the contrary, a theoretical examination of the trust function and acceptance rules of the trust scheme is valuable. Boundaries of the function and different acceptance rules can be assessed. Additionally, the analysis can determine the attestations for acceptance and the trust for acceptance measurements.

A benefit of the approach is the evaluation of the trust model without having a data source and to make conclusions on a theoretic level. Especially, the trust function and acceptance rules can be analysed. On the contrary, the outcome of studying the attestation and the trust network of a dedicated model is limited.

4.3.3.2 Practice

In this section, we conceptually analyse the sample models according to the components of the trust model meta-framework. Thereby, we concentrate on the trust function and acceptance rules as these elements can be genuinely analysed with the assessment strategy. An overview is presented in Table 4.3.

4.3.3.2.1 Public-key Infrastructure based on X.509 The trust function and acceptance rules in the PKI based on X.509 are simple. If an attribute is issued by a trusted certificate authority, the attribute is accepted. Otherwise, the attribute is rejected. Therefore, the attestations for acceptance metric is 1. A single assertion is sufficient for successful property acceptance. Furthermore, the trust for acceptance measurement is also 1. By normalising the binary trust scale of not trusted and trusted into the interval of 0 and 1, the minimal trust rating to contribute to the property acceptance is 1 as well. Where a large trust value supports security, the dependency on solely one attribute provider can be improved to increase security.

4.3.3.2.2 Pretty Good Privacy The interaction of trust function and acceptance rules in PGP is more complex compared to the PKI. The trust function calculates a trust rating based on the marginal or fully trusted introducers. The acceptance rules accept if the predefined quantity of marginal and fully trusted introducers is exceeded. Per default rules, the characteristic is accepted if two fully trusted introducers assert the value. Thus, the attestations for acceptance metric is 2. The descriptive trust levels no trust, marginal and full can be normalised by assigning the values 0, 0.5 and 1 to them in the respective order. As marginally-rated attestation issuers increase the trust rating towards acceptance, the trust for acceptance measurement is 0.5.

4.3.3.2.3 A Logic-based Assurance Framework The Logic-based Assurance Framework specifies trust in an attribute and its provider based on logical Horn clauses. As analysed in the taxonomy classification section (cf. 4.3.2.2.3), there is no trust-enhancing attribute aggregation implemented. Thus, the attestations for acceptance metric is 1. Furthermore, the trust for acceptance measurement is not definable because the different trust levels are specified by Horn clauses. However, an aggregation of trust levels to form another is not foreseen in the applied logical

Trust Model	AfA	TfA
PKI based on X.509	1	1
PGP	2	0.5
Logic-based Assurance Framework	1	n/a
AttributeTrust	1	> 0

Table 4.3: Trust decision process metrics for sample models

reasoning calculus. If there are three logical organisational trust levels, it cannot be simply deduced that two of them can be aggregated to the third one.

4.3.3.2.4 AttributeTrust The AttributeTrust framework specifies the trust in an attribute provider through aggregated confidence paths. The basic trust in a node is a confidence value in the interval $[0, 1]$. As a relying party needs to decide if a certain attribute provider is accepted for a property, the attestations for acceptance metric is solely 1. Several attestations are not combined together. Likewise, we concluded on the attribute aggregation methodology only on completing in the classification approach (cf. 4.3.2.2.4).

Besides that, the AttributeTust applies concatenation and aggregation for the confidence values. Therefore, all confidence values that are larger than 0 contribute to the acceptance of a property. Thus, the trust for acceptance metric is defined in a similar manner.

4.3.4 Practical Study

The practical analysis approach directly investigates an existing instance of a trust model that results from a used implementation. Hereby, a data source must be identified to obtain all information about the current state of the model. The data is parsed into a structure for evaluation. Ulrich et al.'s [102] analysis of the web of trust is an example of this assessment strategy. Besides that, Capcun [119] pursued the same approach.

Applying the practical analysis approach, the examination strongly depends on the available information for any researcher. Especially if data about all entities is retrievable or if solely a limited set of data can be studied. The practical assessment concentrates on the attestation and the trust network. We can parse the data into the mesh components of our meta-framework. Therefore, the measurements about the degree of centralisation, degree of interconnection as well as issued and received attestations can be properly evaluated. Further network-based properties are also examinable. As realistic data is used, the expressiveness of the characteristics is meaningful.

The attribute assertion information is usually available because it is used by any relying parties. This circumstance holds true for PGP and the PKI based on X.509. For instance, from the SKS server network [120] the PGP attestations can be downloaded. Nonetheless, schemes with the use of stronger privacy mechanism solely make the attestation base during a trust decision process available. The entire attestation network is hidden. On the contrary, mutual trust ratings that can only be anticipated if they are globally pre-defined as per classification criteria. If individual trust ratings are applied that are locally stored, a practical analysis is not possible due to a lack of information. Considering the trust decision process, the practical evaluation cannot provide additional insights based on the default values compared to the conceptual examination strategy. Thus, the characteristics attestations for acceptance and trust for acceptance metric are out of scope.

As an advantage of this strategy, we can gain a realistic view of the network. Thus, conclusions on the devised characteristics are possible. On the downside, holistic data availability is required for a complete examination. That is a contradiction to privacy-preserving implementations.

4.3.5 Simulation

A simulation realistically emulates a specific environment [121]. Conveyed to attribute assurance, the simulation approach uses artificial data to mimic the evolvement of the attestation and trust network, starting from a certain base. Furthermore, the simulation allows practical testing of different trust decision processes. During an iteration within the simulation, all trust model-specific information is assessable. For instance, general agent trust model simulation is implemented by Youssef et al. [91] and Fullam et al. [92].

We start with a definition of the simulation environment as the central element of the methodology. The environment encompasses all static and modifiable information of an attribute assurance trust model.

Definition 4.11 (Simulation environment)**.** *An attribute assurance trust model simulation environment $E = \langle AN, TN, D \rangle$ is a tuple that comprises the attestation network AN, the trust network TN and the trust decision processes D for all entities.*

There are different connected instances of the simulation environment involved in the actual simulation process. A function is applied to transform one instance into another instance of the environment.

Definition 4.12 (Simulation function)**.** *The simulation function f_{sim} transforms an instance of the simulation environment E_i to another instance E_j*

Actually, the simulation function can be seen as a set of functions that transforms every component of the simulation environment in the next state. For instance, a specific function may evolve the attestation network structure by adding further attestations or removing expired assertions. Thereby, the functions can be configured to enable adjustments of the simulation process.

Definition 4.13 (Simulation)**.** *An attribute assurance trust model simulation* $SIM = [E_0, \ldots, E_t]$ *is a series of simulation environment iterations* $0 \ldots t$ *where* f_{sim} *transforms the environments and* $\forall_{i \in [0,t-1]} f_{sim}(E_i) = E_{i+1}$ *holds.*

Overall, a simulation executes several times the simulation function starting on a predefined base environment. The base environment evolves within several iterations. The repeated action stops when the simulation process is interrupted. The construction of the base environment is fundamental for the simulation.

During the simulation process, data access to all trust model components is given. Therefore, the network-related characteristics degree of centralisation and interconnection as well as issued and received attestations can be determined. Furthermore, the metrics attestations and trust for acceptance can be calculated.

Access to all components of a trust model and the execution of different scenarios are advantages of this assessment strategy. The use of artificial data is the disadvantage that limits the expressiveness of the results.

4.3.6 Synopsis

We have presented an evaluation of assessment approaches for trust models in attribute assurance. The study encompassed classification, conceptual analysis, practical study and simulation. A summary of the applicability of the methodologies, except classification, is shown in Table 4.4. The table provides an overview of the assessment approaches and their relevance to the components of a trust model. The sign − indicates that no insight can be gained on a specific trust model element by applying the technique in the column header. For instance, this is the case for the practical analysis with regard to the trust function. A • connotes a restrained applicability. As an example, the elements of a trust network are only assessable in a limited manner by applying the conceptual approach as the knowledge gain is confined. In contrast, the symbol $\surd$ represents full applicability. For instance, it is the case when using the conceptual analysis methodology on the trust function. Using the practical study approach, several elements can either be not at all or fully evaluated depending on the available information.

Trust Model Component		Conceptual Analysis	Practical Study	Simulation
AN	Nodes	•	-/√	•/√
	Attestation	•	-/√	•/√
TN	Nodes	•	-/√	•/√
	Relation	•	-/√	•/√
TD	Function	√	-	•/√
	Trust Base	•	-/√	•/√
	Attestation Base	•	-/√	•/√
	Acceptance Rules	√	-	•/√

Table 4.4: Applicability of trust model assessment approaches

4.4 Summary

In this chapter, we outlined common elements and differentiating factors of trust models towards schemes outside attribute assurance. Furthermore, we defined the characteristics trust scale, trust applicability, attribute aggregation, trust composition and centralisation of trust as classification criteria of trust models in attribute assurance. Based on this information, we proposed a meta-framework to define and analyse trust schemes. The framework encompasses the attestation and trust network as well as the trust decision process, whereas the last part comprises the trust function and acceptance rules as major components.

Using the framework, we devised characteristics and additionally described attacks on trust models that compromise the objectives of availability and integrity. Attacks comprise censorship, denial of service, attribute forgery, rogue attribute provider, stale information and the manipulation of the trust base. The security of a trust model against these attacks depends on the construction of the trust function. The trust function must ensure that attribute providers with a high trust rating are considered. Additionally, the function should not solely rely on one attribute provider. The more attestation issuers are considered, the higher is the resiliency against certain attacks. We pursue this rationale as motivation for the next chapter to shape a secure trust function for self-sovereign identity.

In the second passage of the chapter, we evaluated assessment strategies for trust models. Thereby, we initially presented a diverse range of sample models. Subsequently, we elaborated on the methods classification, conceptual and practical analysis as well as simulation. Classification enables only a very coarse-grained view on trust models. The conceptual analysis allows examining the networks and

the trust decision process. However, the network results are limited due to the modelling of the mesh. In contrast, from the practical analysis approach, detailed insights in the networks can be concluded. Nonetheless, the non-availability of information restricts the potential of the analysis. Finally, the simulation approach allows insights in all areas of the trust model. However, as artificial data is used, the results are also synthetic to a certain extent. We apply classification and the conceptual analysis methodology in the next chapter to evaluate and compare the proposed attribute aggregation trust function. In particular the conceptual analysis approach is superior to evaluate the trust function. A practical analysis is prevented by missing data and the simulation introduces vagueness due to the modelling data.

5 Trust-Enhancing Attribute Aggregation

This chapter presents the foundations for shaping trust-enhancing attribute aggregation [25]. We propose a probabilistic modelling approach for a trust function of this type that is based on validity and correctness. Subsequently, the defined function is extended to a holistic trust model. We conceptually analyse and classify the model in the context of self-sovereign identity. Additionally, we compare it to the sample schemes from the previous chapter (cf. Chapter 4.3.1). Finally, we illustrate its practical use in the implementation of an identity broker and conduct performance measurements.

5.1 Motivation and Related Work

In the previous chapters, we have analysed trust requirements in the context of self-sovereign identity. In conclusion, trust-enhancing attribute aggregation can reduce the dependency towards a single attribute provider. The attribute provider is the remaining trusted third party in the identity management setting despite using a decentralised identity provider. Additionally, we investigated in detail the structure and assessment strategies of trust models in attribute assurance. Here we concluded, that the trust function is the major element. This trust function must feature a low dependency towards a single attribute provider and utilise high trust ratings towards them to be secure against described security attacks. Besides these conclusions, there is a general lack of flexibility in the current use of identity providers as trusted third parties. A service provider chooses a strong identity provider for its service and therefore trusts it. The integration with several identity providers is not conducted due to the effort. If a user wants to interact with the service provider, it must enrol with the selected identity provider. Otherwise, the consumption of the service is not possible. Starting at the side of the user, if a certain identity provider has an enormous user base, the service provider might be forced to integrate into it, to attract this user base. For instance, the secure communication to web servers via HTTPS [122] applies certificates that are part of a PKI based on X.509. The certificates of the root certificate authorities are shipped with the major browsers and operating systems [118]. Therefore,

users trust these certificates by default. Hence, there is a strong interdependency between the identity provider selection on the side of the service provider and the viewpoint of the user. Considering the web of trust, PGP [114] is limited to secure email communication and email address attestations. A wide breakthrough has not happened. Usability issues of the end-user clients seem to be one reason [123] [124]. Additionally, from our point of view, the equality of peers, that is the foundation of a web of trust, and the limited radius of trusted entities are also a drawback of the scheme. A neighbour of a user has the same status as an organisation. A relying party may assign the marginal or full trust rating. However, a service provider might have low trust in case a neighbour of the user attests an attribute. These conclusions motivate us to scrutinise the status quo and work towards a trust function in the self-sovereign identity context to overcome the limitations. The trust function should enable the composition of different providers in a trust-enhancing manner to span a relying party-driven web of trust that may originate from a chain of trust scheme.

Related research work covers the fields of attribute aggregation and attribute assurance trust modelling. Regular attribute aggregation has the objective to complete a set of required properties because one provider is not able to deliver all demanded characteristics [113]. Chadwick et al. [125] present the concept of a linking service for this purpose. The linking service holds the credentials of the different user accounts. Furthermore, it mediates the communication to the identity providers and the service provider. The linking service collects the required attributes and transfers them to the service. Ferdous and Poet [74] published a taxonomy of attribute aggregation models. The location of the aggregation process serves as classification criteria. The side of the user, the identity provider or the service provider is differentiated. Additionally, the authors evaluate security, functional and risk-related characteristics. In 2013, Chadwick and Inman [126] proposed the Trusted Attribute Aggregation Service (TAAS). The TAAS represents an additional authority in the identity management setting for the conflation of properties. Besides that, Ferdous et al. [127] concentrated on a hybrid aggregation model using the Security Assertion Markup Language version 2 (SAML2) [128] to reduce complexity. Identity federation topologies have also been scrutinised by Klingenstein [19]. In addition to that, authors worked on attribute aggregation for a specific use case or protocol [129] [130] [131] [132] [133].

Besides that, the research field of attribute assurance trust modelling encompasses the popular approaches of the PKI based on X.509 [61] as well as the PGP [114] web of trust. Furthermore, Thomas et al. [45] [46] proposed a property-specific assurance framework. This setting enables the specification of trust on attribute level. Additionally, different levels of trust for the identity provider can be assigned. Mohan and Blough published the AttributeTrust [105] scheme to

generate trust in an attribute provider based on reputation that is distributed by confidence paths. Addressing the limitations of the simplistic PGP trust pattern, Jonczy et al. [134] implemented a probabilistic trust aggregation function in the GnuPG [135] client. The approach interprets the trust environment as a network reliability problem. Furthermore, government agencies [50] and other organisations [47] presents simple assurance frameworks (cf. Chapter 2.3) to standardise trust in properties of a user.

Our research differs from related work due to the specific target of trust in attribute assurance. Thereby, the model can be generally used, independent from an attribute class, in the self-sovereign identity context. Previous attribute aggregation models retrieved properties from different attribute providers to complete a set of required characteristics. On the contrary, our aggregation scheme targets the trust increase by providing the same characteristic from different providers. To achieve this, we apply a probabilistic accumulation strategy for trust in the context of the self-sovereign identity paradigm. The probabilistic trust pattern is based on the validity and correctness of an attribute.

5.2 Probabilistic Modelling of Trust in Attributes

Initially, we list notations that will be used in the ensuing sections. Subsequently, we define the building blocks of our trust view in attribute assurance and combine them to determine trust in a single attribute provider. Following this, trust is specified in conjunction of several providers.

5.2.1 Notations

The following notations are used to describe the composition of trust in the ensuing sub sections.

- Capital calligraphic letters, e.g. $\mathcal{A}$, $\mathcal{B}$, indicate stochastic events. A small letter of the same type, e.g. a, b, reflect a specific instance of the event.
- Θ connotes the probability function. Thus, $\Theta(\mathcal{A})$ refers to the probability of event $\mathcal{A}$.
- P denotes the set of attribute providers, whereas $p \in P$ refers to a single provider. Subscripts, e.g. p_1 or p_n, refer to different issuers.
- A specifies the set of attribute types, whereas $a \in A$ references a single attribute type. Different properties are separated by subscripts, e.g. a_1 or a_m.

- The symbols $\wedge$ and $\vee$ represent the logical and- respectively or-conjunction for events.

5.2.2 Correctness and Validity

The user and the service provider strongly depend on authentic attributes for service consumption and provisioning. The dependency is expressed in the trust requirement domain attribute management (cf. Chapter 3.2). Within the domain, two trust prerequisites are essential [15]. First, the property of an attribute must reflect the real-world value. For instance, the first name of an identity must coincide with the real first name of the subject. We denote this demand as correctness. The second trust requirement targets the revocation of an invalid property. Conversely, the property must not be expired. Therefore, we denominate this prerequisite as validity. The validity predicate is important for the properties of an identity that may change. For instance, the address of a person changes if that individual moves to another apartment. Overall, correctness and validity of attributes ensure that user and service provider can interact frictionless with each other.

We shape the correctness and validity of an attribute as random variables with a binary sample space. The result is either true or false depending on if the property is correct respectively incorrect or valid, respectively invalid.

Definition 5.1 (Correctness)**.** *Let $\mathcal{C}$ be a binary random variable that depicts the correctness of an attribute. $\mathcal{C}_p^a$ denotes the correctness of a specific attribute $a \in A$ from an attribute provider $p \in P$. The outcome c of $\mathcal{C}$ (or $\mathcal{C}_p^a$) specifies a single attribute usage at a service provider with the subsequent potential values:*

- *$c = 1$ implies that the attribute is correct*
- *$c = 0$ implies that the attribute is not correct*

Analogue to the definition of correctness, we specify the random variable for validity.

Definition 5.2 (Validity)**.** *Let $\mathcal{V}$ be a binary random variable that depicts the validity of an attribute. $\mathcal{V}_p^a$ denotes the validity of a specific attribute $a \in A$ from an attribute provider $p \in P$. The outcome v of $\mathcal{V}$ (or $\mathcal{V}_p^a$) determines a single attribute usage at a service provider with the subsequent potential values:*

- *$v = 1$ implies that the attribute is valid*
- *$v = 0$ implies that the attribute is not valid*

The usage of attributes is referenced within the definitions of the random variables. Already Maurer [97] observed that a well-defined scenario is required for probabilistic modelling. We denote the usage scenario to a specific action at a dedicated point in time at a relying party. During authentication, the identity provider asserts the user's identity and also transmits its attributes to the service provider.

In the self-sovereign identity context, the application of the service provider requests a set of properties. The user selects corresponding claims and consents the transmission. An attribute-based access control [136] system can then use the properties for authorisation. We consider the authentication process, including the conveyance of the characteristics as a random lottery. Each time a user logs into a service is a new random event. For an instance of the event, the validity and correctness of an attribute can change. Therefore, we declare the quality of an attribute provider as the probability of both peculiarities.

The setting can be transferred to the urn model [137] as a standard stochastic experiment for drawing with replacement. The balls in the urn are either labelled with correct or incorrect, reflecting the correctness of a specific attribute. A pull depicts an authentication process. If a correct ball is taken, the transferred attribute is correct. The quantity of correct and incorrect balls illustrate the specific probability of a provider. The urn model setting can be transferred to the validity in a similar manner.

Definition 5.3 (Probability of correctness)**.** *Let $\Theta(\mathcal{C})$ be the probability for correctness $\mathcal{C}$. $\Theta(\mathcal{C}_p^a)$ defines the probability for an attribute $a \in A$ from the attribute provider $p \in P$ with*

- $\Theta(\mathcal{C}_p^a = 1)$ *represents the probability that the attribute a is correct*
- $\Theta(\mathcal{C}_p^a = 0)$ *represents the probability that attribute a is not correct*

In a similar manner, we define the probability of the validity of an identity's characteristic.

Definition 5.4 (Probability of validity)**.** *Let $\Theta(\mathcal{V})$ be the probability for validity $\mathcal{V}$. $\Theta(\mathcal{V}_p^a)$ defines the probability for an attribute $a \in A$ from the attribute provider $p \in P$ with*

- $\Theta(\mathcal{V}_p^a = 1)$ *reflects the probability that attribute a is valid*
- $\Theta(\mathcal{V}_p^a = 0)$ *reflects the probability that attribute a is not valid*

We omit the attribute superscript and the provider subscript on the event if the ownership is clear. Otherwise, the indicators are added as well. In both scenarios,

a high probability is clearly favourable for the user and the service provider. Low probabilities indicate that it cannot be rely on the quality of the attribute. For instance, if the probability is 0.5 it is likely that the attribute's correctness can change with every authentication procedure.

5.2.3 Trust in an Attribute Provider

We have defined the core characteristics of trustworthiness in attribute assurance as correctness and validity. These values are specific to each attribute provider. To calculate an overall score for a provider, we need to combine the probability values of these properties. A joint probability can be computed aligned with two major ways [137]. The calculation prescript is subject to the dependency between the events of correctness and validity. From our perspective, both events are dependent based on the following rationale.

1. An attribute provider is run by one organisation. This entity is responsible for property verification and revocation processes. One organisation adheres to the same guidelines, has the same workforce and is run under a consistent management.

2. The correctness and validity of an attribute are logically dependent. A properly attested attribute might reasonably expire in case the underlying fact change. A falsely issued characteristic cannot be assessed in the same manner. A solely technical expiration is possible. However, the underlying fact was already different at the point of issuance.

The joint probability of dependent events is calculated based on the following formula [137]. The probability of correctness is multiplied with the probability of the event validity under the condition of the event correctness.

$$\Theta(\mathcal{C}_p^a \wedge \mathcal{V}_p^a) = \Theta(\mathcal{C}_p^a) \cdot \Theta(\mathcal{V}_p^a \mid \mathcal{C}_p^a)[137] \tag{5.1}$$

We assume that the probabilities of both events for a certain attribute provider are known. Therefore, solely the conditional probability must be determined to calculate the overall score for an attribute provider. There is a lower and upper border for the likelihood [138].

$$\Theta(\mathcal{C}_p^a) \cdot \Theta(\mathcal{V}_p^a) \leq \Theta(\mathcal{C}_p^a \wedge \mathcal{V}_p^a) \leq min(\Theta(\mathcal{C}_p^a), \Theta(\mathcal{V}_p^a))[138] \tag{5.2}$$

The lower bound is the product of the probabilities. It reflects the case if both events would be independent. The upper border is represented by the minimum of either the probability of correctness or validity. We approximate the actual

value by a function that incorporates a dependency factor. This item influences the calculation towards the lower or upper bound. The dependency factor reflects the correlation between correctness and validity at a provider. The approximation function is derived from Thomas et al.'s [138] approach.

Definition 5.5 (Approximation function f_{d_p})**.** *Let f_{d_p} be a conditional probability approximation function that is parametrised by $d_p \in (0, 1]$ for an attribute provider $p \in P$. The factor d_p reflects the relation between correctness and validity at the attribute provider p.*

$$f_{d_p}(\Theta(\mathcal{C}_p^a), \Theta(\mathcal{V}_p^a)) = \Theta(\mathcal{C}_p^a) + d_p \cdot min(1, \frac{\Theta(\mathcal{C}_p^a)}{\Theta(\mathcal{V}_p^a)} - \Theta(\mathcal{C}_p^a))[138] \tag{5.3}$$

A dependency factor that is close to 1 illustrates a very high connection of both events. On the contrary, a low value reflects a low interdependence of both events. Applying the dependency factor approximation, we can compute the adjacent joint probability by the subsequent formula.

$$\Theta(\mathcal{P}_p^a) = \Theta(\mathcal{C}_p^a \wedge \mathcal{V}_p^a) \approx \Theta(\mathcal{C}_p^a) \cdot f_{d_p}(\Theta(\mathcal{C}_p^a), \Theta(\mathcal{V}_p^a)) \tag{5.4}$$

Thus, the trustworthiness in an attribute provider lies in the range from 0 to 1. A low value indicates low trust and a score that is approaching 1 denotes a high overall trust value. Further, we denote the joint probability for a provider p as $\mathcal{P}_p$

5.2.4 Conjoin several Attribute Providers

Having outlined the combination of correctness and validity at one attribute provider, we determine the composition of several attestation issuers. In case several attribute providers attest the same property and convey it during the authentication process, the probabilities of these issuers can be combined together. Thus, this approach enables us to apply the trust-enhancing attribute aggregation. If a provider $p_1 \in P$ and a provider $p_2 \in P$ issues the attribute $a \in A$, we can calculate the combined either probability [137] according to the subsequent equation.

$$\Theta(\mathcal{P}_{p_1} \vee \mathcal{P}_{p_2}) = \Theta(\mathcal{P}_{p_1}) + \Theta(\mathcal{P}_{p_2}) - \Theta(\mathcal{P}_{p_1} \wedge \mathcal{P}_{p_2})[137] \tag{5.5}$$

We see both events $\mathcal{P}_{p_1}$ and $\mathcal{P}_{p_2}$ as overlapping in the outcome space. The property might be correct and valid at the same time delivered by the two issuers. Furthermore, situations might exist where only one characteristic applies simultaneously. The joint probability of $\mathcal{P}_{p_1}$ and $\mathcal{P}_{p_2}$ is calculated based on independence. We assume the different attribute providers with their management processes to be autonomous from each other. Because different organisations with separate

staff are responsible. Thus, we calculate the overall probability in a subsequent manner.

$$\Theta(\mathscr{P}_{p_1} \vee \mathscr{P}_{p_2}) = \Theta(\mathscr{P}_{p_1}) + \Theta(\mathscr{P}_{p_2}) - \Theta(\mathscr{P}_{p_1}) \cdot \Theta(\mathscr{P}_{p_2}) \tag{5.6}$$

Having three attribute providers, the equation is extended the following way by an attestation issuer $p_3 \in P$.

$$\begin{aligned} \Theta(\mathscr{P}_{p_1} \vee \mathscr{P}_{p_2} \vee \mathscr{P}_{p_3}) = \Theta(\mathscr{P}_{p_1}) + \Theta(\mathscr{P}_{p_2}) + \Theta(\mathscr{P}_{p_3}) \\ -\Theta(\mathscr{P}_1) \cdot \Theta(\mathscr{P}_2) - \Theta(\mathscr{P}_1) \cdot \Theta(\mathscr{P}_3) \\ -\Theta(\mathscr{P}_2) \cdot \Theta(\mathscr{P}_3) + \Theta(\mathscr{P}_1) \cdot \Theta(\mathscr{P}_2) \cdot \Theta(\mathscr{P}_3) \end{aligned} \tag{5.7}$$

Finally, we can generalise the composition to n attribute providers and derive the subsequent formula for calculation. The equation serves as trust function for us. We simply identify it by Θ. The proof is provided in Appendix A.3.

$$\Theta(\mathscr{P}_{p_1} \vee \ldots \vee \mathscr{P}_{p_n}) = \sum_{i=1}^{n} \sum_{j=1}^{n-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} \tag{5.8}$$

5.3 Trust Model Expansion

In the previous section, we developed the probabilistic trust function Θ for trust-enhancing attribute aggregation. The trust function utilises the verifiable claims in the self-sovereign identity context (cf. Chapter 2.6.3). In particular, the 1 to n relation between a single claim and several attestations serves as the foundation. Nonetheless, the trust function is solely one component of the trust decision process that exists besides the attestation mesh and the trust network. To obtain a complete trust model, we define the remaining elements that are also embedded in the self-sovereign identity context.

The attestation network is reflected by all claims and their assertions. The claims are stored in the identity wallet or in another user-defined location. Additionally, the asserted claims can be inscribed in a claims registry. The actual implementation is specific to the self-sovereign identity solution. An entity that is enrolled with a solution possesses a decentralised identifier that acts as a designator for issuing claims.

The trust network is implicitly given by the relying party's trust opinions towards the attestation issuers. The trust ratings are locally stored at the service provider. When a user authenticates with its self-sovereign identity and presents verifiable claims, the trust information is used to execute the trust decision process.

The trust decision process applies the previously defined trust function Θ. The verifiable claims for a certain attribute that are disclosed by the user form the

attestation base. Additional key input information is the trust base that is built by the locally stored trust information. Finally, acceptance rules are required. We define acceptance rules as a set of logical conclusions to accept a certain property based on a threshold.

Definition 5.6 (Acceptance rules)**.** *Let S bet a set of acceptance rules to decide on the use of an attribute $a \in A$ under n attestations of distinct providers $p_1 \dots p_n \in P$. An element $s_i \in S$ is defined as follows.*

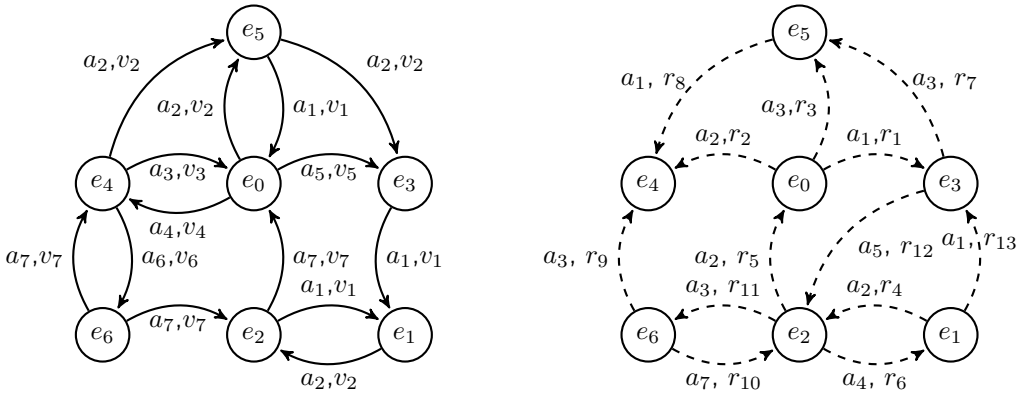
$$s_i : \Theta(\mathscr{P}_{p_1}, \dots, \mathscr{P}_{p_n}) \geq t_{a_i} \Rightarrow a_i \tag{5.9}$$

The threshold t is the configuration element to adjust the accepted attribute for high risk or low-risk scenarios. It can take a value between 0 and 1 which is comparable to the range of the calculated probabilities of the trust function Θ. The value 1 must be chosen in case a high-risk scenario applies to ensure high trustworthiness of the property. Lower trustworthiness is embodied by a reduced threshold that converges to 0. However, the increased risk of a wrong attribute must be considered. We generally assume a default threshold of 1. Furthermore, we solely use Θ, omitting the attribute providers, in an acceptance rule to express the focus on the actual supplied values during an evaluation process.

5.4 Conceptual Analysis and Security

The trust model's attestation network is based on verifiable claims of the used self-sovereign identity solution. Thereby, every participant is able to issue attestations. Besides regular users also organisations can attest properties. For instance, certificate authorities from a PKI can participate as a verifiable claim provider, too. These entities may enjoy a higher level of trust within a larger group of relying parties. In contrast, a self-attested property or a verifiable claim that is issued by a neighbour is scarcely trusted. In case it is trusted, the group of relying entities might be very limited. General trust in a locally known individual seems to be unlikely. A sample attestation network is shown in Fig. 5.1. It depicts typical characteristics of the PGP setting as well as core characteristics of a PKI based on X.509. Node e_0 is comparable to a certificate authority due to a large number of issued attestations. On the contrary, entity e_5 is a regular user, but also asserts an attribute.

Moreover, the trust network is composed of locally stored trust information at the side of the relying parties. Here, we can expect a mixture that unites characteristic elements of PGP and the PKI based on X.509. There are entities that accumulate higher trust values by a larger number of nodes. This is comparable to certificate authorities in the PKI setting. For instance, node e_2 belongs to this



(a) Attestation network **(b)** Trust network

Figure 5.1: Schematic self-sovereign identity trust and attestation network

category. Additionally, there are instances that are solely trusted in a limited manner by a few nodes. The entity e_6 falls in this category.

Considering the network-related properties, there can be a low grade of centralisation in the attestation network due to the general possibility of issuing attestations for all entities. The trust network is structured differently. A centralisation of trust may exist towards a small number of nodes because a certain number of certificate authorities may act as trusted attribute providers. Additionally, there is a high degree of interconnection, as all nodes are able to attest attributes. The majority of the entities issue attestations and receive assertions by other entities. Thus, there is no explicit separation between provider and receiver as compared to a PKI based on X.509.

The security of the trust model depends on the adherence of the trust function to two characteristics (cf. Chapter 4.2.7). First, the usage of high trust values must be ensured. Additionally, the trust function should not only rely on a single or a limit amount of attribute providers. Concerning the first property, the trust function Θ does not aggregate zero trust. In case, the trust base only contains issuers that are rated with a probability for correctness and validity of 0 then the overall calculated trust score is also 0. Medium probabilities have an intermediate influence, and higher likelihoods have a more significant impact. Sample calculations are shown in Table 5.1. The different examples utilise one or three attribute provider to calculate an overall trust score. The dependency factor is chosen as 1.

With regard to the second peculiarity, a single attribute provider is only accepted if the overall trust score equals 1 aligned to the default acceptance rules. Nonetheless, the lower the single trust rating, the more attribute providers are required to exceed the threshold. Therefore, the trust function can be secure but depends on the individual trust base and acceptance rule configuration for a certain attribute.

$\mathbf{p_1}, \ldots, \mathbf{p_n}$	$\mathbf{\Theta}(\mathcal{C}_{\mathbf{p_i}})$	$\mathbf{\Theta}(\mathcal{V}_{\mathbf{p_i}})$	$\mathbf{f_{d_p}}$	$\mathbf{\Theta}(\mathscr{P}_{\mathbf{p_i}})$	$\mathbf{\Theta}(\mathscr{P}_{\mathbf{p_1}}, \ldots, \mathscr{P}_{\mathbf{p_n}})$
$n = 1$	0.8	0.8	1	0.8	0.8
$n = 3$	0.2	0.2	1	0.2	0.488
$n = 3$	0.4	0.4	1	0.4	0.784
$n = 3$	0.9	0.9	1	0.9	0.999

Table 5.1: Sample calculations for trust function Θ

5.5 Comparison with Sample Models

To compare the proposed probabilistic attribute aggregation trust model, we examine the classification criteria and compare it to the sample models. The trust scale is of type continuous. Aggregated probability values of the trust function can range between 0 and 1. Furthermore, the trust applicability in terms of rating and acceptance is individual. A trust rating is defined by the probability of correctness and validity. Additionally, the dependency factor is specific for each attribute provider. Therefore, a relying party can express individual trust ratings to participate in the trust model. In the same way, acceptance is also individual because the thresholds are set independently. Regarding the attribute aggregation method, the focus is clearly on the trust-enhancing approach. Additionally, the completion of a set of properties from distinct providers is also supported by the self-sovereign identity paradigm. The probabilities of different attribute providers are joint to receive an overall rating. In addition to that, the trust composition follows a structured approach as trust is composed theoretically grounded on correctness and validity. With regard to the centralisation of trust, the trust model is decentralised. Attestation issuers comprise distinct entities. Regular users can assert attributes as well as traditional attribute providers. Technically, the classical providers seem to be trusted third parties. However, the impact of these central authorities is limited by having individual ratings and acceptance thresholds. Thus, the complete attribute assurance trust model is decentralised. Table 5.2 illustrates an overview of the characteristics.

Characteristic	**Probabilistic Attribute Aggregation**
Trust scale	Continuous
Trust applicability	Individual/ individual
Attribute aggregation	Trust-enhancing/ completing
Trust composition	Structured
Centralisation of trust	Decentral

Table 5.2: Characteristics of the attribute aggregation trust model

In Chapter 4, we introduced and classified the sample trust models PKI based on X.509, PGP, Logic-based Assurance Framework and AttributeTrust. AttributeTrust employs a continuous trust scale that is comparable to the proposed probabilistic attribute aggregation model. Additionally, the scale is also aligned in the range between 0 and 1. On the contrary, other models apply a binary or a fine-grained discrete trust scale. Concerning trust applicability, the PKI based on X.509 solely uses predefined ratings and acceptance levels. In this category, our trust model is in line with PGP, Logic-based Assurance Framework and AttributeTrust. Furthermore, trust-enhancing attribute aggregation is implemented by PGP and the proposed trust scheme. Besides that, the PKI based on X.509 does not apply any pattern in this regard, and the remaining models integrate the completing approach. With reference to trust composition, we apply a structured approach that is comparable to the Logic-based Assurance Framework. The other models imply a simple comprehension solely. A decentralised trust setting is implemented by our trust model and by PGP. On the contrary, the other trust schemes implement a centralised environment.

Overall, the proposed probabilistic attribute aggregation model shares characteristics with PGP, Logic-based Assurance Framework and AttributeTrust. However, there is no common peculiarity with the PKI system based on X.509.

5.6 Application and Use

We developed an identity broker for the practical application of the trust model in the self-sovereign identity context. A detailed illustration of this broker service is given in the next Chapter 6. The broker is positioned at the side of the service provider and mediates the communication between the user, its self-sovereign identity solution and the services. Major components of the broker are the integration in different self-sovereign identity solutions as an input channel, a trust engine for evaluation and authentication protocols as output route.

The input channel receives the verifiable claims of the user. The claims and additional information about their issuers are forwarded to the trust engine. The trust engine evaluates the attestations for every claim according to a trust function that is implemented as trust module. For this purpose, the trust engine accesses the local trust base and acceptance rules of the service provider. These elements are part of the broker's configuration database. Fig. 5.2 shows the entity-relationship diagram of the trust base and acceptance rules. An attribute provider is addressed by its decentralised identifiers. Several designators per issuer are supported because an attribute provider may register multiple times at a decentralised identity provider or enrols at different ones. The assumed probabilities for correctness and validity can be specified on the issuer and attribute level. This provides flexibility

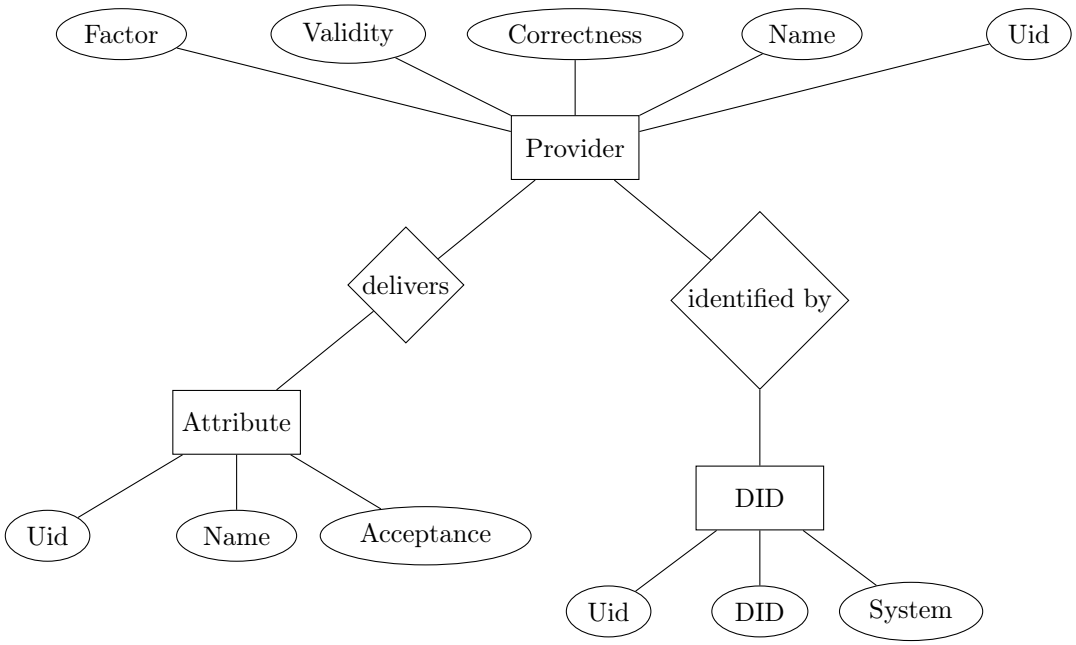


Figure 5.2: Trust base and acceptance rules entity-relationship diagram

for the configuration. The dependency factor is provider-specific and can only be configured on the respective tier. Additionally, an acceptance score for the attribute can be recorded. In case the calculated score during the trust evaluation exceeds the acceptance threshold, the attribute is routed to the output protocol.

5.7 Performance Evaluation

The identity broker, its trust engine and the respective trust module is implemented in the Python language. Furthermore, the Tornado [139] web application framework is used. We use as a test environment a virtual machine with 1024 MB main memory and one CPU having 2.4 Ghz clock rate. The operating system of the virtual machine is Ubuntu 18.04. The database schema, including the trust base of the identity broker, is implemented in a PostgreSQL 10.15 database. The database environment is installed on the same virtual machine.

We conducted several test scenarios to evaluate the execution time of the implemented algorithm for trust function Θ. Appendix A.4 depicts the pseudo code of the algorithm. For the distinct scenarios, we increased the number of attribute providers and reviewed a different number of delivered characteristics. Fig. 5.3 presents the collected measurements. Within the graph, the y-axis depicts the number of delivered attributes, whereas the x-axis shows the number of attribute

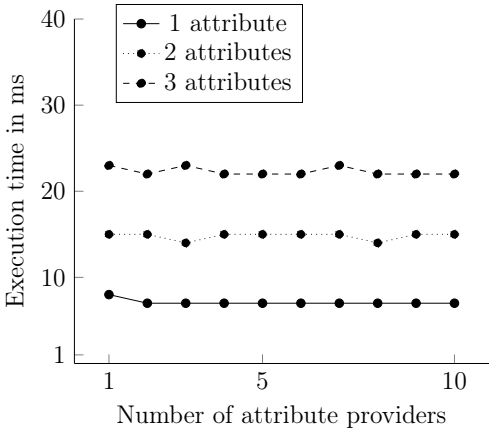


Figure 5.3: Attribute aggregation execution times

providers that delivers these properties. The solid line shows the verification of a single property. An average execution time of 7 milliseconds (ms) is captured despite a varying amount of attestation issuer. The dotted line shows the evaluation of two attributes. The mean execution time results in 15 ms. The last curve reflects the calculation for three attributes. 22 ms is the average processing time. An increase in the processing time that significantly depends on the number of attributes can be deduced. On the contrary, the number of issuers that deliver an attribute has no measurable impact. The influence might be minor and below the measurement accuracy of the environment. The execution time of the initial run for all scenarios is minimally higher than the next rounds. Caching mechanisms of the database system might be the reason for this.

Overall, we can conclude on the practical applicability of the algorithm based on the limited measured execution times. The result predominantly depend on the number of aggregated attributes and is impartial from the provider quantity.

5.8 Summary

In this chapter, we proposed a probabilistic trust function based on correctness and validity of a property delivered by a specific attribute provider. Furthermore, the probabilities of a single provider are combined from several issuers to enable trust-enhancing attribute aggregation. We embedded the trust function into the self-sovereign identity context to form an entire attribute assurance trust model. Thereby, we can use verifiable claims as attestation network and a locally stored trust base as well as acceptance rules. Moreover, we conducted the conceptional analysis, including security characteristics of the trust model. We can conclude that the previously worked out security characteristics hold true for our

probabilistic trust function under certain conditions. Besides that, we examined the classification criteria and compared the peculiarities with the sample trust models. Additionally, we showed the practical application of the trust model as implemented trust module of an identity broker at the side of the service provider. Thereby, we presented the schema of the configuration database. Thus, the service provider can drive a web of trust that comprises regular users as well as traditional trusted third parties from the self-sovereign identity context. Besides the practical use of trust-enhancing attribute aggregation, the identity broker provides further significant benefits, for instance, in the area of interoperability. Therefore, the close elaboration of it serves as motivation for the next chapter.

6 Attribute Trust-Enhancing Identity Broker

This chapter depicts the design and assessment of the Attribute Trust-Enhancing Identity Broker (ATIB) [24] [22]. We categorise the concept as interoperability approach [21], describe the current challenges for service providers and summarise requirements. Subsequently, we outline the architecture and implementation of ATIB. Finally, we conduct a security analysis using the attack tree methodology.

6.1 Motivation and Related Work

The postulation of the self-sovereign identity paradigm came at the same time as the advancement of blockchain technology. Smart contracts or dedicated blockchains were ready to build the foundation for a decentralised identity provider. As a parallel development, the blockchain offered the initial coin offering [140] approach as a new investment vehicle for projects. This progress attracted continuously growing interest by communities and the general public. As a consequence thereof, the sparked hype about blockchain leads to the creation of numerous self-sovereign identity implementations [20]. We also pursued initially research on a blockchain-based identity provider [27] that was integrated into Ethereum's state machine. The solutions come with dedicated application libraries for integration into the services and applications of the service provider. Furthermore, existing standard protocols for authentication, for instance, Security Assertion Markup Language version 2 (SAML2) [128] or OpenID Connect (OIDC) [141], have been largely disregarded. Besides traditional efforts, new protocols, e.g. DIDAuth [142], are being developed that are devoted to self-sovereign identity. However, the adoption of new standards requires an extensive period in enterprises. Beyond the mere integration effort, the new paradigm also requires attributes that are issued as Verifiable Claims (VC). Service providers must be enabled to issue claims for participation. These shortcomings of the self-sovereign identity paradigm might result from the unconditional focus on the user. However, the identity and attribute ecosystem solely thrives in case users are able to login at relying parties. Aside from pure adoption, we are also motivated by fostering the practical use of trust-enhancing attribute aggregation. We outlined the theoretical concepts

about the calculation and the foundation of trust models in the previous chapters. ATIB allows the practical application of trust-enhancing attribute aggregation. These circumstances found the motivation to research on an identity broker for self-sovereign identity.

Related work spans two major fields. On the one side, researchers and projects work on mediated self-sovereign identity solution integration. The objective is to abstract from a specific integration and broker the usage of implementations. One approach to achieve decoupling is the Universal Resolver [143]. A digital identity of the self-sovereign identity paradigm is distinguished by its Decentralised Identifier (DID) [67]. The DID provides a standard for identifiers in the blockchain-based self-sovereign identity setting. A component of the DID is a designator to determine the applied decentralised identity provider. The main purpose of the Universal Resolver is the translation of a DID into a corresponding DID document [67]. In this process, the respective self-sovereign identity solution is queried to deliver the record or contained information. The DID document encompasses public keys for verification, supported protocols or communication endpoints. The login process, supported by the Universal Resolver, requires initially as input the DID by the user. Subsequently, the implemented authentication process is run. Besides the Universal Resolver, Hyperledger (HL) Aries [144] is a client to integrate self-sovereign identity solutions into the regular application landscape. It is built for the usage of HL Indy [72] which is a set of blockchains that are dedicated to identity management. The vision for HL Aries is the brokered integration of a range of solutions. However, it currently only supports HL Indy.

In addition to implementation approaches, the development of standards and protocols drive the interoperability of self-sovereign identity implementations. For instance, DIDAuth determines an authentication flow. The DID standard defines the structure of identifiers. The Verifiable Credential [69] standard specifies the composition of verifiable claims. Furthermore, attribute aggregation services have been investigated (cf. Chapter 5.1). Chadwick and Inman [126] developed the TAAS to combine attributes from different attribute providers. Additionally, Ferdous and Poet [74] published a categorisation of aggregation schemes based on the location where the accumulation occurs.

Our proposed self-sovereign identity broker, ATIB, is a distinct mediation approach. Where the Universal Resolver demands additional steps for authentication, ATIB directly utilises the process that a self-sovereign identity solution implements. Furthermore, protocols and standards require time for adoption. ATIB bridges this duration gap and enables the use of traditional protocols. Additionally, ATIB consists of a component-based architecture, including the activated trust model. Thus, trust-enhancing attribute aggregation can be applied by using the self-sovereign identity attribute ecosystem.

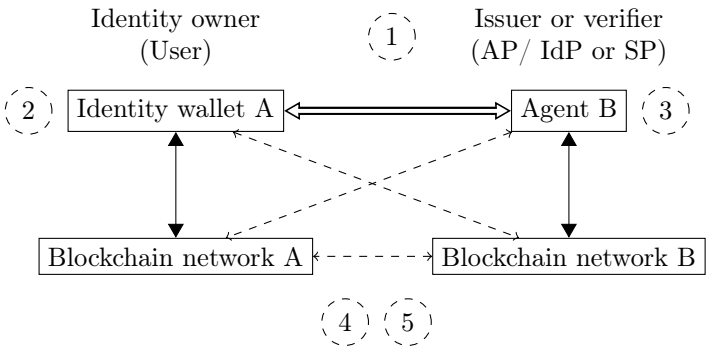


Figure 6.1: Interoperability scenarios

6.2 Interoperability Approaches

The general objective of interoperability is determined by the ability of two entities to communicate [145]. Therefore, various domains consider interoperability as a fundamental principle. Researchers define layered interoperability models to structure research in this domain [146]. These layers may group technical, semantic, syntactical, or cultural characteristics. Koussouris et al. [147] nominate identity management as one of the principal areas for interoperability studies. The inherent challenge of authentication and authorization beyond system and trust boundaries is the underlying driver. Along the same lines, Cameron [148] identified the need for interoperability between identity management systems in his laws of identity. For the self-sovereign identity paradigm, Allen [11] stated interoperability as one of the essential principles (cf. Section 2.5).

A regular self-sovereign identity interaction is characterised by communication within a single management system. The service provider and the attribute provider uses an agent of the specific system. Moreover, the user applies the corresponding identity wallet of the same solution. In addition to the direct communication between the two parties, a common blockchain network that is comprised of the overall system is integrated into the process and serves as decentralised identity provider (cf. Section 2.7). Proceeding from the regular interaction, we define interoperability between two self-sovereign identity solutions A and B as a working interaction when the service provider or attribute provider uses an agent of system A, and the user applies the identity wallet of system B. The same interoperable communication can exist vice versa.

Definition 6.1 (Self-sovereign identity management system interoperability)**.** *The self-sovereign identity management systems A and B are interoperable if an entity with identity and agent on system A can interact with another entity that has an identity on system B and is applying the corresponding identity wallet.*

Fig. 6.1 depicts the described interoperability scenario. Furthermore, the diagram outlines different positions where concepts to achieve across interaction can be implemented. The location implies a dedicated interoperability solution. Protocols and standards (1) reflect agreements for data structures and process flows to enable interoperable communication. The entities speak the same language to interact with each other. We can differentiate the identity provider interaction and routing class. The first category describes the communication rules between the identity wallet, agent, and the blockchain network. If self-sovereign identity solutions adhere to the same rules, the agent and identity wallet can seamlessly interact with different blockchain networks. The routing class comprises protocols and standards that facilitate direct information exchange between the blockchain networks. Thus, the user and the service provider communicate with their self-sovereign identity management system. The distinct blockchain networks are connected to forward requests and responses.

Moreover, the concept of an identity broker provides interoperable communication. The identity broker mediates the communication towards different self-sovereign identity solutions. Thus, a distinct integration is not required. We can distinguish the user-side (2) and the service provider-side identity broker (3). The first broker type abstracts from a single self-sovereign identity system at the location of the user. The broker might be integrated as a generic identity wallet. The latter type enables the service provider to offer an arbitrary solution to its customers. Thereby, the application of the service provider integrates to the broker, and the broker integrates lightweight to the self-sovereign identity systems. ATIB is a service provider-side identity broker.

Additional interoperability concepts are the hub (4) and the paring (5). The hub and the pairing connect blockchain networks of different self-sovereign identity management systems to forward messages between the distinct solutions. These schemes are comparable to the routing protocols and standards but rely on a software component, and not upfront agreed rules. For both concepts, we can differentiate between a decentralised and a centralised variant. The decentralised version is implemented as a separate blockchain and therefore does not create a trusted third party. In contrast, the centralised approach must be run by a dedicated entity. Furthermore, a pairing exactly connects two blockchain networks, whereas the hub is not limited to this ratio. A direct pairing is implemented on the blockchain networks to exchange messages and does not require an additional component.

Overall, protocols and standards are a favourable interoperability approach but require time to develop and get adopted by the solutions. The broker concept can directly foster interoperability and can also react to implementation differences of the different systems.

6.3 Challenges for Service Provider

Service providers face several challenges during the adoption of self-sovereign identity solutions. These obstacles encompass the number of solutions, divergent trust in attribute providers, existing application landscape and the verifiable claims as properties of a user.

6.3.1 Multitude of Self-Sovereign Identity Solutions

The emerging hype about blockchain and its applications lead to the creation of a myriad of self-sovereign identity solutions [20] [149]. These implementations cover identity wallets, application-related software, a decentralised identity provider or all components to build a holistic ecosystem. An identity wallet refers to an application that enables the user to control its self-sovereign identity. Decentralised identity providers based on blockchain are realised in different manners. uPort [65] is implemented as a set of smart contracts on the unpermissioned blockchain Ethereum. HL Indy constructs a set of permissioned blockchains for identity management. One chain is used for administration. Additional chains carry the actual identity data. Besides that, Blockstack [66] implements identity management on top of several existing blockchains. Where Ethereum and therefore uPort, has no governing entity, an instance of HL Indy requires a consortium that runs stewards. As diverse are the concepts, as different are integration strategies. Each solution provides an individual integration library or Application Programming Interface (API). Thus, a service provider needs to spend enormous effort for integration. Additionally, there is not yet a single decentralised identity provider with a large user base that could be a clear winner. Overall, the fragmented self-sovereign solution landscape imposes a significant challenge for the service provider and may impede the general ecosystem development.

6.3.2 Divergent Trust in Attribute Providers

In the classical identity management models, the identifier and the attributes are issued by the same provider. The service provider and the user have to decide which identity provider they trust. Subsequently, the service provider integrates these identity providers in its applications. In parallel, the user enrols at its favourite entities. Thereby, both parties are subject to constraints. A service provider prefers an identity provider with a large user base to increase access to its application. A user has different motives choosing a provider. For instance, data privacy might be a relevant concern. Thus, a strong mutual dependency exists that might contradict the preference of service provider and user. In the worst case, an identity provider is chosen that is hardly trusted by any of both parties. Independent flexibility to

choose trusted attribute providers will be a benefit for the user and the service provider and ensures better matching of their trust preferences.

6.3.3 Existing Application Landscape

Over time, the service provider establishes numerous applications to serve its client. In particular, large enterprises have a tremendous application landscape encompassing heterogeneous technologies and potential technological debt. Furthermore, this complex set of applications may adhere to outdated implementation patterns and uses deprecated software. Transforming the existing application landscape towards the use of self-sovereign identity solutions requires an enormous endeavour in case major adjustments are necessary. Current self-sovereign identity systems primarily apply proprietary integration libraries. Additionally, new protocols are developed. This evolution demands great integration efforts from the service provider. To facilitate the adoption of self-sovereign identity, the integration efforts must be minimised. Using established identity and access management protocols is a general option to be independent of technology and application changes.

6.3.4 Attributes based on Verifiable Claims

The traditional identity provider receives information about the user, verifies the data and makes it available as attributes for the service provider. Data verification procedures are fundamental to provide authentic properties for secure service provisioning of the relying party. Especially in high-risk settings, the proof operations have significant importance. In the self-sovereign identity ecosystem, attributes of an identity are represented as verifiable claims. These claims can be self-asserted by the user or issued by another entity. Self-asserted properties have limited applicability due to unknown trustworthiness. Besides the attribute provider, the service provider has benefits of issuing claims to the user. The service provider possesses original data, for instance, subscriptions to memberships or generally access to restricted areas, that can be used as verifiable claims for authorisation decisions at a later point in time. Owning the information does not directly imply to be able to issue verifiable claims. Besides that, if a user creates a new self-sovereign identity, there are no attributes. Without attributes, an identity is of no use at relying parties. To address these challenges, service providers and other entities require verifiable claims issuance facilities. On the one side, entity-specific data can be issued. On the other side, claims about public verifiable information can be attested. General availability of such applications supports the self-sovereign identity setting. Additionally, it fosters new attribute assurance trust models to enable emerging attribute providers.

6.4 Requirements

The previously described challenges for service providers lay the foundation for developing our identity broker ATIB. Furthermore, we consider the new options of trust-enhancing attribute aggregation in the self-sovereign identity setting. To formalise the objectives of ATIB we define the following requirements as the basis.

- **R1 Authentication:** Our identity broker must support the authentication process for applications. The user should be able to use its favourite self-sovereign identity solution for authentication with the respective wallet.
- **R2 Authorisation:** Subsequent to the authentication, the service should be able to conduct authorisation decisions according to the attributes of the user. Attribute-based access control must be supported by using verifiable claims. In particular, the user can convey its verifiable claims to the application for authorisation.
- **R3 Verifiable Claim Issuance:** The identity broker enables the service provider to easily issue claims based on its own or publicly available data.
- **R4 Self-Sovereign Identity Independence:** The implemented functions of the broker, for instance, authentication, authorisation and claim issuance, must be independent of the used self-sovereign identity solution of the user. Thus, the service provider does not need to integrate several solutions. Additionally, the user can freely choose their favourite application.
- **R5 Flexible Attribute Trust:** ATIB should enable the user and the service provider to individually decide on their trusted attribute providers without a force to make a corresponding choice.
- **R6 Application Technology Autonomy:** Our identity broker should be independent of the used applications at the side of the service provider. This requirement supports both a heterogeneous technology stack as well as a loosely coupled architecture to foster change.
- **R7 Non-impairment of Self-Sovereign Identity Principles:** The broker should not harm the self-sovereign identity principles. The principles are fundamental for a new age of identity management, and therefore, they should not be compromised. ATIB fosters the development of the ecosystem by driving the adoption at the side of the service provider.
- **R8 Security:** The implementation of ATIB should adhere to security best practices. In particular, the communication between application and identity wallet must happen securely.

6.5 Architecture

In this section, we describe the architecture of ATIB. Initially, we present the concept, and subsequently, we elaborate on the components and interfaces.

6.5.1 Concept

Our Attribute Trust-Enhancing Identity Broker is an identity provider that acts as a proxy between web applications and self-sovereign identity solutions. The communication towards web applications is performed via traditional identity and access management protocols. However, ATIB does not comprise a user identity store that contains credential and attribute information. When a user authenticates at an online service, the user gets redirected to ATIB. Ensuing, the identity broker mediates the communication with the self-sovereign identity solution that is chosen by the user. The user logs in with its identity wallet. After successful authentication, the identifier is conveyed to the web application via ATIB. Additionally, requested attributes of the user are transmitted within the protocol flow. The user properties are derived from the verifiable claims. Moreover, the user consents the conveyance of the attributes. ATIB follows a component-based architecture and consists of several interfaces. The interfaces connect the components inside ATIB and enable the outside communication with the interaction partners. Fig. 6.2 provides an architecture diagram of ATIB.

6.5.2 Components

The components of ATIB comprise the namespace translator, trust engine, protocol manager, self-sovereign identity manager and the verifiable claim issuer.

6.5.2.1 Namespace Translator

The namespace translator component converts claim names between different domains. Usually, the same property is addressed by different references in various contexts, protocols or realms. The component enables interoperability between the claim names. Other components can utilise the service of the namespace translator via the name translation interface to retrieve the correct property identifier for the expected use.

6.5.2.2 Trust Engine

The trust engine is the principal component to evaluate trust in the verifiable claims of an identity. Within the trust engine, trust modules, that are the imple-

mentation of different trust functions, are used to determine trust in an identity's property. Thereby, the issuers of the variable claims are assessed according to the used trust base and acceptance rules. The applied trust module reflects the subjective trust opinion of the hosting entity of ATIB. The protocol manager component interacts with the trust engine through the attribute trust interface.

6.5.2.3 Protocol Manager

The protocol manager is the central element of ATIB. It can encompass the implementations of various standard identity and access management protocols. Furthermore, the protocol manager orchestrates the main process flows and interactions between the other components. The central element invokes the namespace translator, the self-sovereign identity manager and the trust engine to execute user authentication as well as authorisation processes. Via the self-sovereign identity manager, the verifiable claims of the user are retrieved. After evaluation of the trustworthiness in the trust engine, the requested attributes are transmitted during the conducted protocol flow.

6.5.2.4 Self-Sovereign Identity Manager

The self-sovereign identity manager mediates communication with the distinct self-sovereign identity solutions. For this interaction, a generic wrapper interface exists. Additionally, for each self-sovereign identity implementation, a specific adapter must be created. The adapter abstracts from implementation libraries on the one side. On the other side, it realises the generic wrapper interface. An adapter must support the following core functions.

- **Create Identity:** The function creates a new identity on the self-sovereign identity solution. ATIB requires an identity to issue claims, request claims during a process or serve generally as a communication endpoint. The identity is shown as a requesting entity when obtaining user consent. Overall, the identity actually represents the organisation that is running ATIB.
- **Create Challenge:** The purpose of the create challenge function is the generation of authentication challenges. The user can respond to the authentication challenge with the support of its identity wallet. The challenge may already include an attribute request to process them further.
- **Verify Challenge:** The verify challenge function testifies the response to an authentication challenge. In particular, structural completeness as well as signatures of the sender, are proven.

- **Request Verifiable Claim:** The request verifiable claim function enables ATIB to start the communication to the self-sovereign identity solution for requesting verifiable claims. That is the case, if the authentication is separated from retrieving attributes or if additional properties are required at a later point in time.
- **Verify Verifiable Claim:** Through verify verifiable claim function, ATIB can testify received verifiable claim information. Depending on the self-sovereign identity solution, the respective blockchain network is involved in the verification process.
- **Create Verifiable Claim:** The create verifiable claim function allows ATIB to issue a verifiable claim with a certain value for the user. The user can obtain the claim into its identity wallet.

6.5.2.5 Verifiable Claim Issuer

The verifiable claim issuer component encompasses all functions to steer the claim issuance process. Thereby, verification procedures for attribute values are executed. Hence, a user can retrieve a verifiable claim that is issued from the identity of ATIB and benefits from the trustworthiness of the hosting entity.

6.5.3 External Interfaces

Our identity broker ATIB defines several interfaces for communication with the surrounding environment and its administration. The external interfaces of ATIB comprise the admin, the VC issuer presentation, VC verifier and protocol interfaces.

6.5.3.1 Admin Interfaces

The different admin interfaces that are shown in Fig. 6.2 serve the administration and configuration of the components of ATIB. Configuration options comprise the management of the supported self-sovereign identity solutions, available verifiable claim issuers, and the used digital identity.

6.5.3.2 VC Issuer Presentation Interface

The VC issuer presentation interface enables the user to obtain claims and guide it through the complete process. The interface depicts the claim graphically, and the user can store it in its identity wallet. Additionally, the user might provide further data that must be verified for the claim to be trustworthy.

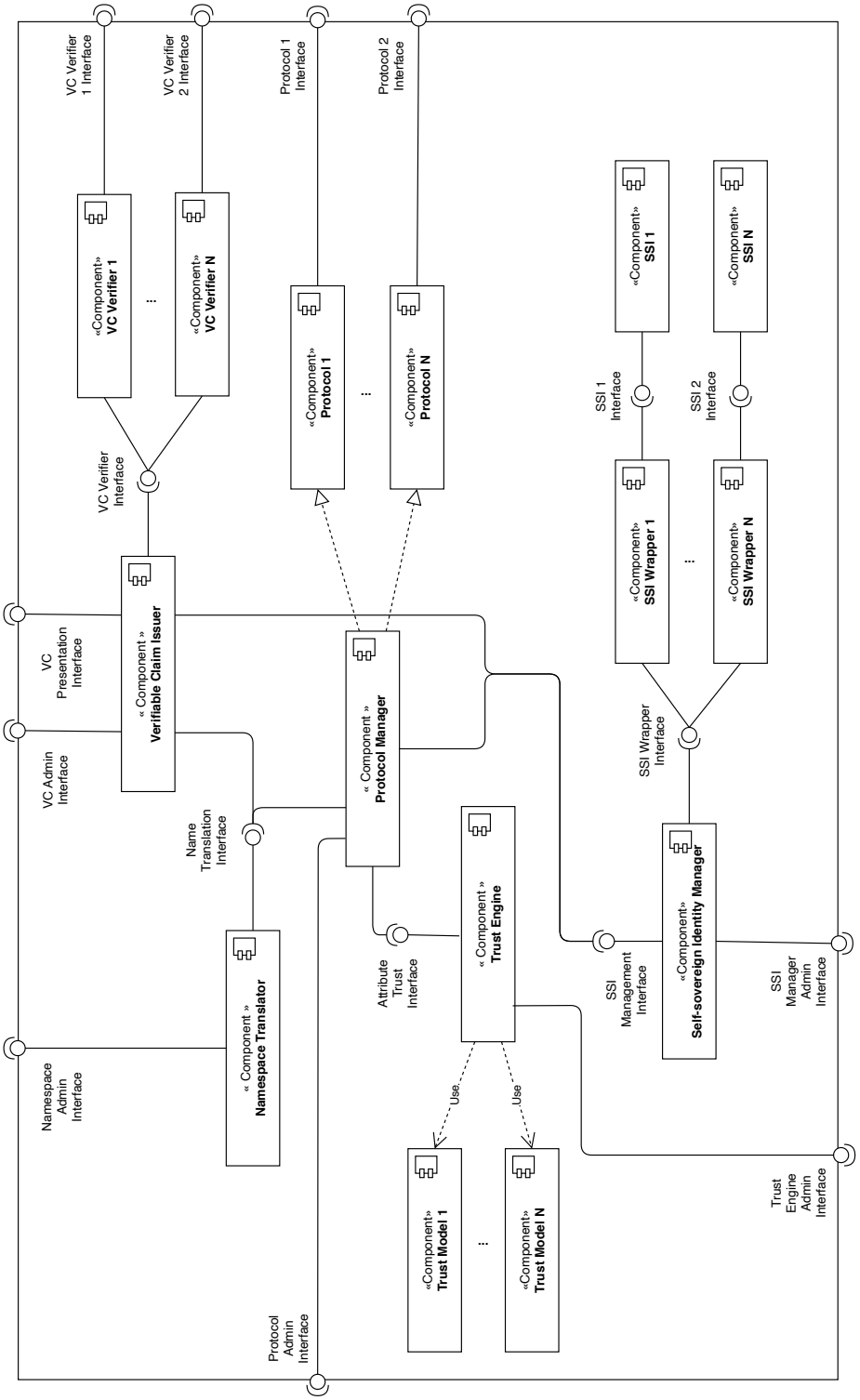


Figure 6.2: Component view of ATIB architecture

6.5.3.3 VC Verifier Interfaces

The VC verifier interfaces communicate with surrounding applications to retrieve or validate information. Data verification build the foundation to properly issue verifiable claims. A strong verification process supports the trust in the identity and its organisation.

6.5.3.4 Protocol Interfaces

The various protocol interfaces belong to the implemented established identity and access management protocols. They serve as communication endpoints for web applications for identity and attribute assertions to support the authentication and authorisation process.

6.6 Deployment Patterns

ATIB can be deployed according to three major patterns. The patterns differ on the impact of the trust boundary between the user and the service provider. We describe the schemas as user-centric, dedicated to a service provider and independent. Fig. 6.3 to Fig. 6.5 show the different deployment schemes. The dashed circles represent the trust boundary. The arrows illustrate communication paths.

6.6.1 User-Centric

The user-centric deployment pattern is depicted in Fig. 6.3. In this scheme, the user installs its own ATIB instance. Thus, ATIB runs in the trust boundary of the user and outside the trust realm of any service provider that the user intends to communicate with. The approach is comparable to an user-operated identity provider in the context of OpenID [150]. Furthermore, the applied trust module and configuration fully represents the subjective trust opinion of the user. This situation is very advantageous for the user and completely supports the self-sovereign identity principles.

Besides that, the user holds solely verifiable claims that are issued by trusted attestation issuers. There seems to be no incentive for the user to obtain attributes from non-trusted providers. Taking this in consideration, the user's trust opinion towards attestation issuers is already satisfied, and this rationale does not support for choosing the user-centric location of ATIB. Furthermore, the trust rating of a user is unlikely to match the trust perception of one service provider. It is even more unlikely that the trust opinion is aligned with all service providers a user wants to interact with. As a result, the service provider may reject the communication offer of the user. In addition to that, a service provider would

need to integrate to an enormous number of identity brokers. This might be infeasible even if solely minimal configuration effort is considered.

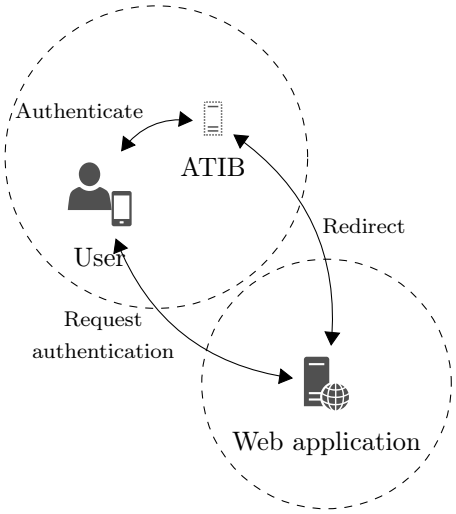


Figure 6.3: User-centric location

6.6.2 Dedicated to Service Provider

Fig. 6.4 illustrates the deployment scheme that is dedicated to a service provider. The identity broker ATIB belongs to the trust domain of the service provider. All applications of the service provider can integrate into the broker for authentication and authorisation. The executed trust module and configuration reflect the specific opinion of the service provider towards the attestation issuers. Additionally, acceptance rules are adapted to risk levels for the used attributes in the applications. Upon executing the authentication and authorisation process, the user can still supply its trusted verifiable claims. In case, the trust perception of the user and the service provider matches, the interaction can commence. Additionally, the user can directly start with the interaction. No specific integration towards ATIB is demanded.

6.6.3 Independent

The independent location of ATIB is visualised in Fig. 6.5. In this setting, an independent party hosts the identity broker outside the trust domain of the user and the service provider. Hence, a new trust realm is established. The hosting party can be seen as a new identity provider that acts as a trusted third party. The implemented trust module and configuration may not match either the user

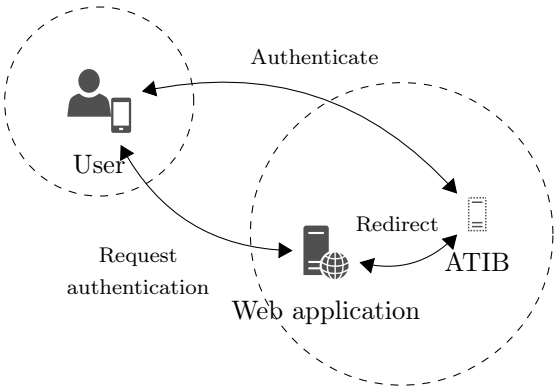


Figure 6.4: Location at a service provider

and the service provider. Thus, the user is forced to obtain verifiable claims of issuers that are accepted by the ATIB host. Furthermore, the service provider must also trust this entity itself and its attribute provider selection. This deployment pattern provides the least effort for the user and the service provider to participate in the self-sovereign identity paradigm. However, the significant disadvantage is the re-establishment of a trusted third party that counteracts the core principle of self-sovereignty.

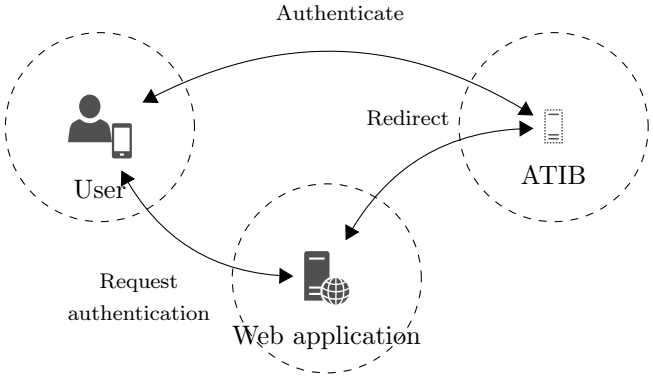


Figure 6.5: Independent location

6.6.4 Synopsis

We outlined the three deployment patterns for the identity broker that comprises the user-centric, dedicated to a service provider and the independent approach. The user-centric setting significantly increases the integration effort for the service provider to an impractical level. At the same time, no serious advantage for the self-sovereign identity paradigm is gained. Besides that, the independent setting

reinstates a trusted third party and does not deem to be acceptable when using self-sovereign identity solutions. Finally, the location that is dedicated to a service provider is optimal for the identity broker. There is no impairment of the self-sovereign identity paradigm. Additionally, it provides a low integration effort for a service provider and reflects its subjective trust opinion. For the remaining chapter, we assume this deployment pattern.

6.7 Fulfilment of Requirements

Subsequent to the description of the general architecture and the potential deployment options, we can discuss the realisation of the previously formulated requirements for the identity broker (cf. Chapter 6.4). The processes of authentication (**R1**) and authorisation (**R2**) should be supported. The protocol manager component of ATIB implements these procedures and makes them available for other applications. Depending on the actually implemented protocol, the specific authentication and authorisation characteristic varies. In general, these procedures are routed to the self-sovereign identity solution and required attributes are requested. These properties can also serve for attribute-based access control. Additionally, the protocol manager component also facilitates the fulfilment of the technology autonomy requirement (**R6**). The usage of protocols and standards enables cross-technology interaction and independence from the technology stack. The requirement about verifiable claim issuance (**R3**) is satisfied by the verifiable claim issuer component. Ensuing to potential data verification processes, claims to the self-sovereign identity solution can be issued.

According to requirement (**R4**) the implementation must be independent of any specific self-sovereign identity solution. This prerequisite is core for the identity broker. The decoupling is achieved by the component-based architecture of ATIB. Additionally, the self-sovereign identity manager controls the communication and a thin wrapper around the API library of the solution lifts the actual integration. Thereby, the wrapper is as lightweight as possible to decrease effort in case the self-sovereign identity library changes. In contrast, the generic wrapper interface stays constant over time. The trust engine itself, but also the support of different trust modules, implement the flexible trust requirement (**R5**). The flexible trust usage allows the specification of a composed set of attribute providers with varying trustworthiness.

Moreover, we describe the fulfilment of adhering to the self-sovereign identity principles (**R7**) in the next section and demonstrate a security analysis (**R8**) in Section 6.12.

6.8 Conformance to Self-Sovereign Identity Principles

Allen [11] proposed the self-sovereign identity paradigm on the basis of foundational principles. We described these axioms in Chapter 2.5. Moreover, we set the non-impairment of the principles as a requirement (**R7**) for our identity broker. A fostered service provider adoption should not counteract the core of self-sovereignty.

The principles of existence, persistence, portability and protection, refer to the self-sovereign identity solution itself and are not affected by the use of an identity broker. Concerning existence, the solution's constitution ensures that a digital identity refers to a specific subject. In the same context, the objective of persistence is also realised by the implementation. The user should be able to decide how long the identity endures. ATIB has no influence in this regard. Besides that, the characteristic portability is also independent from the identity broker. The identity wallet or the complete self-sovereign identity solution must ensure portability. ATIB does not prevent any portability schemes. In contrast, the verifiable claim issuance facility supports certainly the transfer of claims. An attribute can easily be issued to another solution by ATIB if the user makes this decision. Protection references to the precedence of user rights before the network. ATIB does not interfere between the connection of the identity wallet, the user and the blockchain network.

Furthermore, the control axiom refers to the user's eventual control about its digital identity. This control is practised via the identity wallet. Additionally, the blockchain network removes the central authority. ATIB is solely positioned at the side of the service provider and bridges the gap towards its applications. The control lies still with the user. The control principles are further strengthened by enabling diverse trust models.

Considering the access principle, the user must be fully aware of its verifiable claims. The identity broker requests the demanded attributes for the applications from the user. There is no additional stored user information. When issuing attributes, the verifiable claim is provided to the user's identity wallet. The user can accept or reject the retrieval of the property.

Moreover, interoperability must be a strong characteristic of self-sovereignty to enable widespread usage. ATIB does not prevent any interoperable application. On the contrary, due to missing standards and non-adherence to established protocols, ATIB fosters brokered integration to close this gap. Therefore, it supports interoperability between self-sovereign identity solutions as well as towards the legacy application landscape. The first mentioned scenario is supported, for

instance, by issuing claims to multiple solutions, whereas the latter one is in particular endorsed by the use of OIDC and SAML2.

Furthermore, the peculiarity consent has high significance for user acceptance. Each use of the identity and disclosure of attributes must be approved by the user. User consent mechanisms are usually integrated into the identity wallet. These mechanisms are triggered by ATIB during authentication and attribute retrieval processes. As the attributes of the user are only routed through ATIB, the consent principle is not impaired.

Besides that, ATIB adheres to the minimalisation proposition. The identity broker requests and therefore reveals only the necessary attributes for application access. No additional properties of a user are requested or required for other purposes.

Finally, the transparency principle determines an open and honest functioning of the identity broker. ATIB does not have any hidden services. Additionally, the functional layer is thin, and routes attributes directly from the user to the targeted web application. Likewise, we published the source code of ATIB on GitHub[1] and made it open source.

6.9 Implementation

In this section, we describe the general technical architecture to provide an overview. Additionally, we delineate the covered functionality of the ATIB components.

6.9.1 Technical Architecture

The technical architecture of ATIB is shown in Fig. 6.6. ATIB is split into the ATIB Core and the ATIB User Interface. Both elements are hosted on the ATIB server. We use a virtual machine with the operating system Ubuntu 18.04. The ATIB Core element is implemented in Python by using the Tornado [139] web application development framework. The configuration is stored in a locally protected file by using the YAML [151] standard. Additional information, for instance, trust models or OIDC data, is stored in a PostgreSQL database. The database is co-located in the same virtual machine. Additionally, the ATIB User Interface interacts with the ATIB Core module based on the web service paradigm. The implemented self-sovereign identity wrappers run as a web service. Connectors exist for uPort, Jolocom and HL Aries. Exemplary, ATIB Core can connect to a mail server via SMTP [152] and to a directory service through the LDAPS [153]

[1] https://github.com/agruener2000/ssixa-core

protocol for the execution of data verification procedures. This serves as the foundation for the verifiable claim issuance.

Despite that, the OIDC and SAML2 protocol are offered for related identity and access management processes for other applications. In addition to that, the network communication from a user towards the ATIB User Interface and from an application to the mentioned protocols is channelled through a proxy.

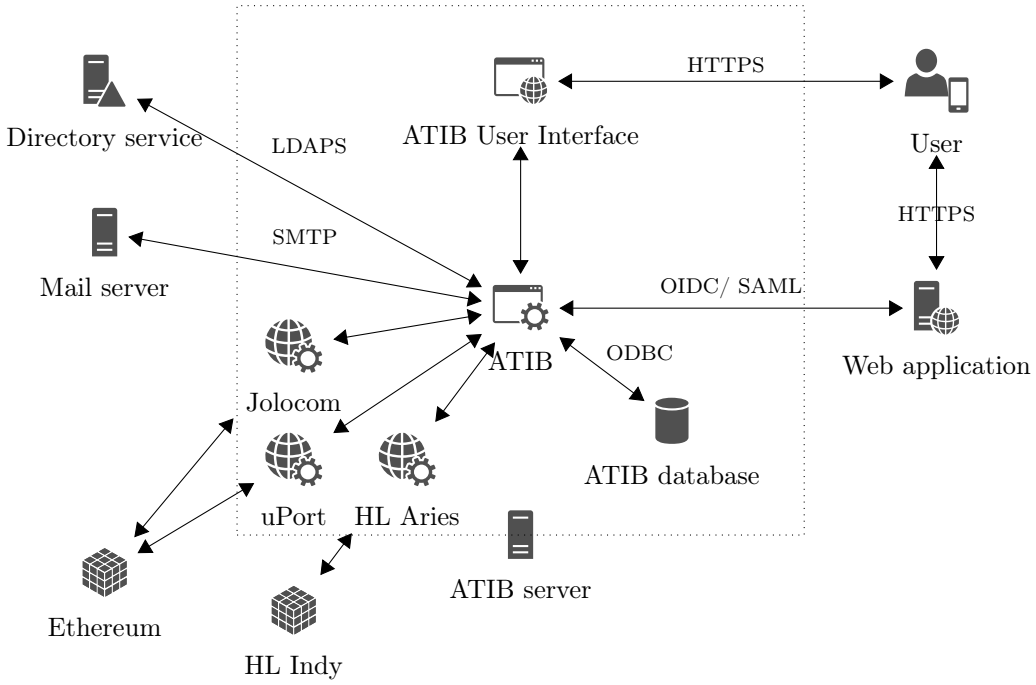


Figure 6.6: Technical architecture of ATIB

The technical architecture makes use of the microservice paradigm and implements the single elements in a decoupled manner. This solution fosters on the one side load balancing in larger environments and certain independence towards the programming language used by the self-sovereign identity solution. Concerning load balancing, in the depicted setup, all components are installed on one virtual machine. Nonetheless, horizontal and vertical scaling opportunities exist at different positions. The used web proxy can distribute the traffic to several ATIB User Interfaces and ATIB Core entities. Furthermore, the self-sovereign identity wrappers can be scaled in quantity to distribute requests. The independence to the used programming language of the self-sovereign identity solution is achieved by the light-weight wrappers. The wrapper must use the supplied library, but ATIB Core accesses it as web service.

Claim	uPort	Jolocom	OIDC
Email	email	ProofOfEmailCredential	email
Name	name	ProofOfNameCredential	name
Firstname	firstname	ProofOfFirstnameCredential	given_name
Lastname	lastname	ProofOfLastnameCredential	family_name

Table 6.1: Claim names in distinct domains

6.9.2 Realised Components

In the following sections, we outline the specifically implemented capability of the components of ATIB.

6.9.2.1 Namespace Translator

Our identity broker implementation can translate the names for the attributes email address, name, first name and last name. A synopsis is shown in Table 6.1. The property email address is referenced by email from uPort and the OIDC protocol. In contrast, Jolocom refers to it with ProofOfEmailCredential. Additionally, HL Indy and SAML2 do not define specific claim names. However, entities can define their proprietary schemas. HL Indy uses the notation of schema definitions. Derived from the schema, credential definitions are created. The claim name is specified by the schema creator. This freedom creates a lot of flexibility for issuing and proofing claims. Nonetheless, the flexibility may create a large amount of different schemas and definitions. Thus, translation has even higher importance.

6.9.2.2 Trust Engine

For the trust engine component, we implemented two trust modules. The trust modules represent a simple trust model and an extended scheme for the trust-enhancing attribute aggregation. The simple trust module accepts only verifiable claims from the identity of ATIB itself. Therefore, the issuer of the verifiable claim is validated against the decentralised identifier of the identity broker. This approach is aligned to the isolated or centralised identity management model. An organisation only permits the asserted attributes from one authority.

Additionally, the extended trust model supports trust-enhancing attribute aggregation (cf. Chapter 5). This module applies the defined trust function Θ by using an expanded local trust base and attribute-specific acceptance rules. Using this setup, a trust behaviour that is comparable to the simplified model can be achieved. The identity of ATIB is rated with 1 for the probability of correctness and validity in the local trust base. Additionally, the dependency factor is also

configured to 1. In case the acceptance rule solely admits attributes at a threshold of 1.0, only properties are permitted that have been issued by ATIB. However, adding additional attributes with an acceptance threshold of 0 leads to the admittance of any issuer. Thus, self-attested properties are also transmitted. Fig. 6.7 presents a query on the ATIB database to retrieve the stored trust information. In this environment, ATIB has an identifier for uPort and Jolocom.

```
6
7 SELECT did, correctness, validity, factor FROM tm_did d
8 LEFT JOIN tm_providertodid ptod ON d.id = ptod.did_id
9 LEFT JOIN tm_provider p ON p.id = ptod.provider_id;
```

Query History

did	correctness	validity	factor
did:ethr:0x3469acbadd3f33e84a15c894aa5536a4af243...	1.0	1.0	1.0
did:jolo:aa5108d798411b90d3c7f4615e82eeb72b86d0a...	1.0	1.0	1.0

Figure 6.7: ATIB identifier and trust ratings

6.9.2.3 Protocol Manager

The protocol manager component of ATIB supports the OIDC and SAML2 protocol for identity and attribute assertions. The implementation of the OIDC standard is based on the pyoidc [154] Python library. For SAML2 we use the corresponding pysaml [155] library. With regard to OIDC, ATIB provides the following endpoints.

- **Authorisation Endpoint:** The authorisation endpoint is the start for executing the authentication process. The service provider application redirects the user to the relative URL of ATIB */oidc/authorization* when the login process is started. The implemented authentication method Blockchain is requested by default. Subsequently, the authentication process is handed over to the self-sovereign identity solution.
- **Token Endpoint:** The relative URL */oidc/token* is the entry point for the token endpoint. At this communication interface, an access or an ID token of the user can be retrieved. The ID token comprises attributes of the user that have been composed of the verifiable claims of the user.

- **UserInfo Endpoint:** The userinfo endpoint is accessed via */oidc/userinfo* This communication interface enables an application to retrieve further user attributes. The requested properties trigger the *Self-Sovereign Identity Manager* to retrieve verifiable claims accordingly from the user.
- **Further Endpoints:** Besides the previously described endpoints, additional endpoints of the OIDC protocol for session management exists. For instance, the user session can be terminated, or a new application can be registered to ATIB. These endpoints solely apply default OIDC functionality and do not exchange data with the self-sovereign identity solution.

Concerning the SAML2 protocol, ATIB implements the following communication endpoints.

- **Single-Sign On Service:** The single-sign on service is called by an application to obtain an identity assertion. If the user is not yet logged in the authentication process will be started. The user must log in with its self-sovereign identity solution and convey the required verifiable claims. The communication endpoint is available under the relative URL */saml/sso*.
- **Single Logout Service:** The single logout service terminates a user session when it is requested by a service provider application. The endpoint is accessible via the ATIB URL */saml/slo*.

The coordinated use of the endpoints for OIDC and SAML2 during user authentication is depicted in Section 6.10.

6.9.2.4 Self-Sovereign Identity Manager

The self-sovereign identity manager component implements the usage of uPort, Jolocom and HL Indy. The communication towards the solutions is mediated by the respective wrapper services. Fig. 6.8 presents the ATIB generic wrapper function to create a new authentication challenge for uPort. The signature contains a callback address for the authentication response and the required verifiable claims. Self-attested properties and attributes that are attested by other parties are differentiated.

The wrapper of uPort itself is implemented in Node.js [156]. We use the official library imports (uport-credentials) and connect to the Rinkeby [157] test network. Likewise, the Jolocom wrapper is based on Node.js using the library jolocom-lib. Furthermore, Jolocom's decentralised identity provider components are implemented on Ethereum's Rinkeby test network. Fig. 6.9 shows uPort's

```
1  def createChallenge(self, callback, claims, claims_verified):
2    result = self.executeWSCall('createchallenge',
3            appname=self.app,
4            did=self.appid, privatekey=self.key,
5            claims=json.dumps(claims),
6            claims_verified=json.dumps(claims_verified),
7            callback=callback)
8    try:
9      return json.loads(result)['jwt']
10   except Exception as e:
11     log.exception("uPort WS Call for Create Challenge failed")
```

Figure 6.8: Generic wrapper interface's create challenge call

wrapper code to create an authentication challenge. Initially, a new credential object is initiated by providing ATIB as application name, the DID of its identities and the secret to proof control of the DID (line 2-7). Subsequently, the actual authentication challenge is generated (line 11-21). uPort calls it a disclosure request. This information request contains the demanded claims and the callback URL. A disclosure request is a signed JSON Web Token (JWT) [158]. After generating the token within the wrapper, the token is sent back to the self-sovereign identity manager component.

Besides uPort and Jolocom, the wrapper for HL Aries is implemented in Python. We use the offered cloud agent [159] for interaction and connect to the Verifiable Organisation Network [160]. The network is a test environment of HL Indy that is initiated by the Government of British Columbia. On contrast to the other self-sovereign identity solutions, HL Aries demands two process steps for the disclosure of verifiable claims. First, a connection between the agents is established via an authentication challenge. However, the challenge does not contain the request for attributes. After the successful creation of the connection, a credential disclosure request can be issued.

For the user, the JWT token is presented as Quick Response (QR) [161] code on ATIB's user interface. The user can capture the code with its identity wallet that is generally a mobile application on the smartphone. In Fig. 6.10 sample authentication challenges for uPort and Jolocom are shown. Both JWT tokens encompass standard information. The tag iss refers to the issuer of the token and contains the DID or a solution specific identifier. The issuance (iat) and expiration time (exp) describe the validity period of the challenge. Besides the standard token fields, the commonly used notations have already ended. Either

uPort and Jolocom specify a token type. However, where the first solution refers with tag type, the latter one calls it typ.

```
app.route('/createchallenge').get(function create(req,res){
    const cred = new credentials.Credentials({
        appName: appname,
        did: did,
        privateKey: privatekey,
        resolver: new didresolver.Resolver(ethrdidres ...))
    })
    var credentialList = JSON.parse(claims);
    var verified_credentialList = JSON.parse(claims_verified);

    cred.createDisclosureRequest({
        requested: credentialList,
        verified: verified_credentialList,
        notifications: true,
        callbackUrl: callback
    }).then(requestToken => {
        res.send(JSON.stringify({jwt:requestToken}))
    }, function(err){
        res.sendStatus(400);
        console.log(err);
    })
})
```

Figure 6.9: uPort wrapper create challenge function

Additionally, the values are distinct (shareReq and credentialRequest) although the purpose is similar. This situation continues considering the further elements of the token. Verifiable claims are requested in uPort by including a list of property names with either the tag request or verified. The identifiers differentiate self-attested and third party attested attributes. In contrast, the creators of Jolocom used a complex structure referred by interactionToken. In addition to that, uPort allows the specification of permissions and Jolocom implements constraints for further communication activities. Both solutions provide within their tokens a callback address. The address refers to a web service that waits for the response token.

Analysing the authentication challenge of HL Aries, we can determine further differences. Such a token is depicted in Fig. 6.11. The HL Aries communication starts with an invitation request. This JWT omits the standard tags. The tag @type defines the tag category for communication. Additionally, information to establish an interaction channel is provided. The data encompasses ATIB's DID

```
{
 "iss":"2oyjAieDKi...U3QZN",
 "iat":"1548559818",
 "requested":["email"],
 "verified":["email"],
 "permissions":["notifications"],
 "callback":"https://...Gli1k",
 "net":"0x4",
 "exp":"1548588618",
 "type":"shareReq"
}
```

(a) uPort

```
{
 "interactionToken":{
  "credentialRequirements":[{
   "type":["Credential",
    "ProofOfNameCredential"],
   "constraints":[]}],
  "callbackURL":"https://..."},
 "typ":"credentialRequest",
 "iat":"1548581905743",
 "exp":"1548585505743",
 "iss":"did:jolo:3c79034...",
 "jti":"b53f28695fe0a"
}
```

(b) Jolocom

Figure 6.10: Authentication challenges

```
{
 "@type": "did:sov:...;spec/connections/1.0/invitation",
 "@id": "913fdc8a...883d8793bc25",
 "recipientKeys": ["2SxP6LaWzK...k4G1eDdHweBwpCyB"],
 "label": "ATIB",
 "serviceEndpoint": "http://.../blockchain/aries/routing/"
}
```

Figure 6.11: Hyperledger Aries invitation challenge

(@id), keys (recipientKeys), an identifying label (label) and the service endpoint (serviceEndpoint). The user can send a response back to the specified endpoint.

6.9.2.5 Verifiable Claim Issuer

The verifiable claim issuer component processes data verification routines and can publish an attested claim to the user. Each claim may require different input information and access to surrounding applications to validate data. We have implemented two claim verification processes.

The email address of a user is generally required at a web application. On the one side, the email address is used to communicate with the user. Additionally, the password reset process depends on a verified email address to send a reset link to. The usage of a self-sovereign identity solution makes a password reset process

obsolete. However, communication with the user is still important. Fig. 6.12 outlines the verification process of an email address. The user authenticates at the ATIB user interface. When logged in, the user enters the email address, where it claims ownership (1). Hence, ATIB sends a verification email to the address (2). It includes a previously generated random number. If the user is indeed able to access the mailbox, the link can be opened. This action serves as a confirmation of ownership (3). Subsequently, the user can retrieve a verifiable claim that attests the email address for the authenticated self-sovereign identity.

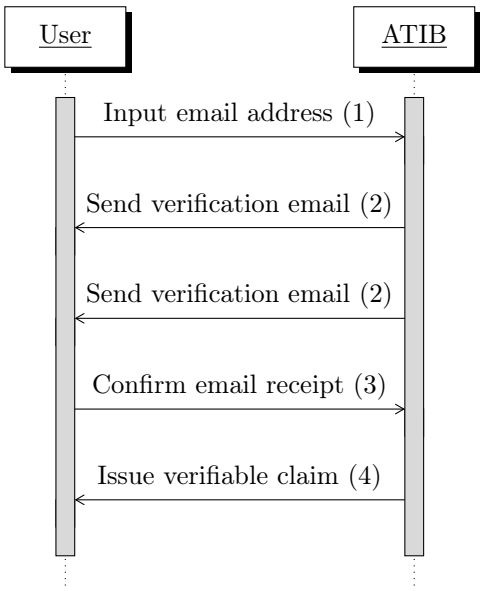


Figure 6.12: Process sequence for email verification

Besides the email verification, we developed a module that connects to a directory service. The directory service regularly stores user information in corporations and may serve as an authentication solution. We use the directory service to issue a verifiable claim about the name of a user. The user provides its distinguished name and the password. The verification module executes a bind against the directory service to determine validity. Additionally, ATIB searches for the value of the displayname attribute, that belongs to the InetOrgPerson class, for the provided user name. Subsequently, the user is able to obtain the corresponding claim.

6.10 Authentication Flows

In this section, we describe certain prerequisites and the authentication flow for the OIDC and the SAML2 protocol when accessing a restricted area of an application.

6.10.1 Prerequisites

Before running the authentication process, a set of requirements have to be fulfilled. The user must install a self-sovereign identity wallet and create a new identity. An existing identity can also be used. Furthermore, the user prepares self-asserted claims or obtains verifiable claims from a trusted issuer that is demanded by the web application.

On the side of the service provider, the application must be configured in ATIB. A unique identifier and a common secret are required. Fig. 6.13 presents sample information that must be stored for a client application to facilitate the OIDC protocol flow.

Figure 6.13: OIDC client information

6.10.2 OpenID Connect

Ensuing the preparations, a user can start the authentication process. We use the ATIB User Interface to demonstrate the OIDC process flow. The complete sequence of steps is depicted in Fig. 6.14. The user opens the ATIB User Interface and decides to authenticate (1). The ATIB User Interface solely supports the OIDC protocol and the ATIB backend as an identity provider. Thus, the user's browser is redirected to the authorisation endpoint of ATIB. The redirection call comprises the client_id, scope and response_type parameter. The client_id is the identifier of the application to verify stored secret information and login as well as logout URLs. Required attributes for the ATIB User Interface are transmitted with the scope parameter. ATIB requests the given_name, family_name or name. Additionally, the parameter value comprises openid to indicate the openid specification. Finally, the response_type is set to code. This configuration requests the authorisation code flow.

Upon calling the authorisation code handler, ATIB verifies if there is already an existing authenticated session. As the user is not yet authenticated, a new session is instantiated. ATIB chooses as default user authentication method our implemented Blockchain module. The Blockchain authentication method can instantiate uPort, Jolocom and HL Aries as supported solutions. uPort is used as the default solution that will be initially selected. Afterwards, the namespace

translator components transpose the requested properties into the terms firstname, lastname and name. Based on these attributes, the self-sovereign identity manager creates an authentication challenge. The challenge is shown as QR code to the user on ATIB and requests the actual authentication (3). The user opens the uPort identity wallet (4). The identity wallet is usually a smartphone app.

Subsequently, the user scans the authentication challenge (5). The identity wallet decodes the QR code (6) and extracts the required properties. The demanded attributes are shown to the user to obtain its consent for transfer (8). In the case the user agrees to the transmittal, the uPort app generates a signed response token and sends it to the callback URL from the challenge (9). For these callbacks, ATIB uses the relative URL */oidc/blockchain/verificationside/*. A large random number is unique to each authentication request and connects the challenge with the response. If the user does not consent, the authentication process stops. An ATIB internal time out will reset data structures and discard the random number. The received response token will be verified by ATIB's self-sovereign identity manager with support of the respective libraries of the self-sovereign identity solution. In case the token's authenticity has been successfully verified, the token information is parsed.

uPort separates the delivered attributes in the category verified and unverified. Both classes are evaluated by the trust engine. As the requested attributes are solely presented in the welcome message on ATIB's user interface, the acceptance threshold is with zero trust. Thus, even self-asserted properties are positively evaluated in the trust module. After finishing the parallel authentication response processing, an existing web socket connection is used to redirect the user's browser again to the authorisation handler. Additionally, the session cookie is created in the browser. Executing the authorisation handler, the established authenticated session leads to the redirection of the login URL of ATIB's User Interface (10). This URL has been transmitted initially and is also stored in the client configuration as redirection destination after the login. The route includes the code parameter that encompasses an access token credential.

Subsequently, the user interface calls the token endpoint of ATIB by providing the code token as a credential (11). ATIB verifies the token and selects the firstname, lastname, name from a temporary memory store for the user. The attributes are then transferred the user interface (12). As the user interface only uses these properties for the welcome message, the attributes are not mandatory. ATIB's User Interface checks which attribute is available and presents them (13). If no attribute has been delivered, no welcome message is shown.

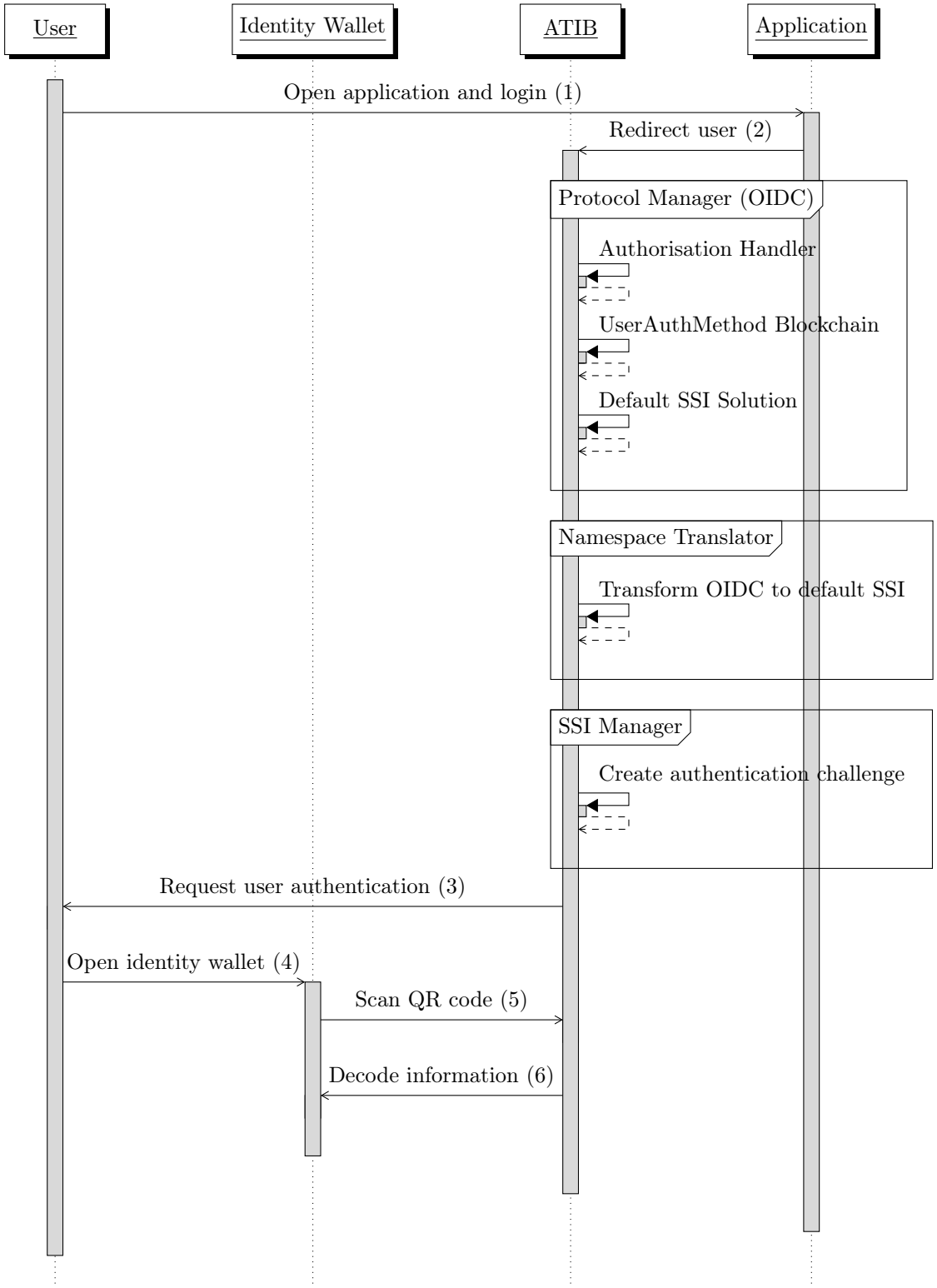


Figure 6.14: Authentication process sequence with OpenID Connect (1/3)

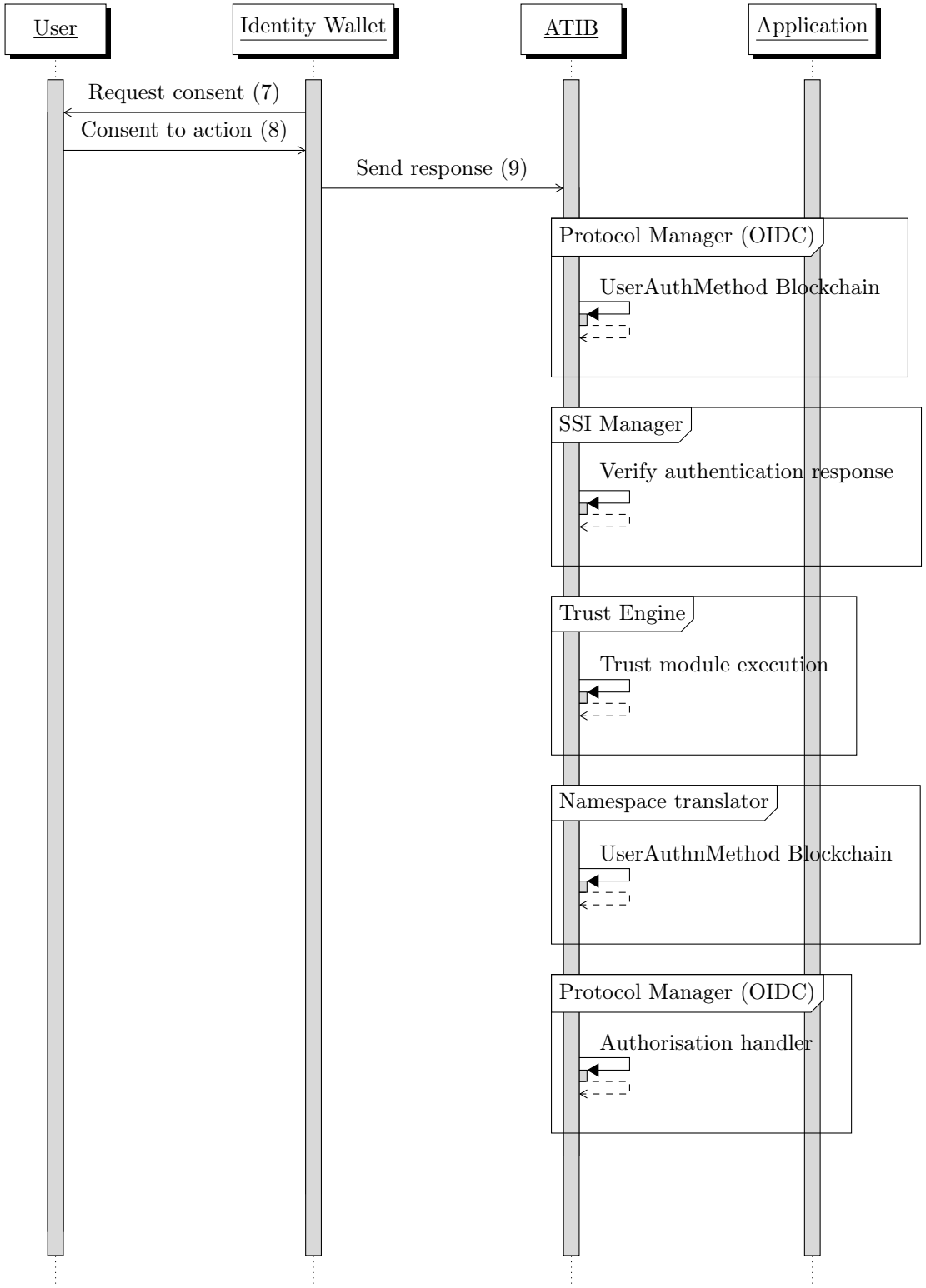


Figure 6.15: Authentication process sequence with OpenID Connect (2/3)

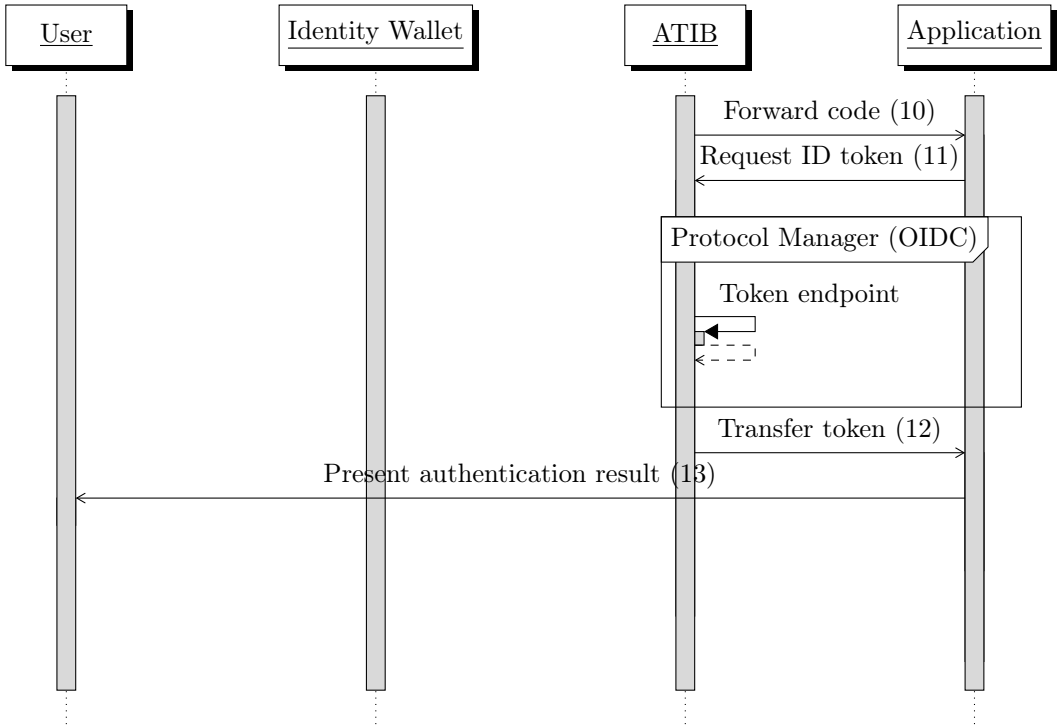


Figure 6.16: Authentication process sequence with OpenID Connect (3/3)

6.10.3 SAML Version 2

The authentication process for SAML2 emanates similarly compared to the OIDC process. The user requests authentication at an application and selects ATIB as an identity provider. Thus, the user is redirected to ATIB. For the demonstration, we use a sample service provider application. The redirection call comprises a SAML2 authentication request. The destination of the invocation is the single-sign on endpoint of ATIB. Afterwards, the request is parsed, and ATIB verifies if an existing authenticated session exists. This is not the case during the first call. Therefore, ATIB internally redirects to the Blockchain authentication method. Supported by uPort as the default self-sovereign identity solution, the authentication challenge is created and presented to the user. The user can change to another solution if preferred.

Subsequently, the user scans with its identity wallet the authentication challenge and consents to the transfer of the requested verifiable claims. ATIB receives the response and verifies it within the self-sovereign identity manager. Ensuing, the claims are extracted and evaluated by the trust engine. In case the response has been successfully verified, ATIB triggers the user's browser via a web socket connection to reload the single-sign on endpoint. At this time, an authenticated

session is recognized by ATIB, and the necessary properties are packaged and returned to the assertion consumer service of the sample application.

6.11 Performance Evaluation

We execute the ATIB proof of concept application on a virtual machine with 1024 MB main memory and one CPU having 2.4 Ghz clock rate. Additionally, Nginx is installed as a reverse proxy to distribute the web requests to ATIB. Furthermore, a separate virtual machine with the same specification serves as a platform to execute test scenarios. This virtual machine is hosted on the same network. We conduct three test scenarios and determine the respective duration of the request. The tests are run with the support of the Locust [162] load testing framework.

1. **Load main page:** The first test case measures the loading of the main page of ATIB. The duration for opening this page serves as baseline.

2. **Generate authentication challenge:** Within this scenario, the authentication page is demanded. The request generates the authentication challenge for uPort.

3. **Perform authentication process:** The test scenario performs a complete authentication process with uPort. The authentication challenge is generated and the response is processed.

We execute each scenario with an increasing number of concurrent user in the Locust framework. Furthermore, we conduct every test case repeatedly to calculate an average duration and to exclude one-time effects. Fig 6.17 presents the results of the analysis. The chart shows on the x-axis the number of concurrent users and on the y-axis the execution times. The execution times of the first scenario (solid line) are constant at about 10 milliseconds. The duration of requesting the authentication page (dotted line) starts with about 1.4 seconds by one concurrent user and increases up to 33 seconds by using 40 concurrent user. In a comparable manner, the execution time for the complete authentication process (dashed line) lasts approximately 5 seconds with 1 concurrent user and raises to about 111 second per request for 40 concurrent user. As a result, the raise of concurrent users significantly increases the execution times in scenario 2 and 3. Besides that, the execution time of scenario 1 is substantially lower compared to the other scenarios.

The consideration of a maximum of 40 concurrent users is not realistic for a widely used identity management component. In particular, if we respect the C10K problem [163] as a benchmark. However, it indicates a performance curve

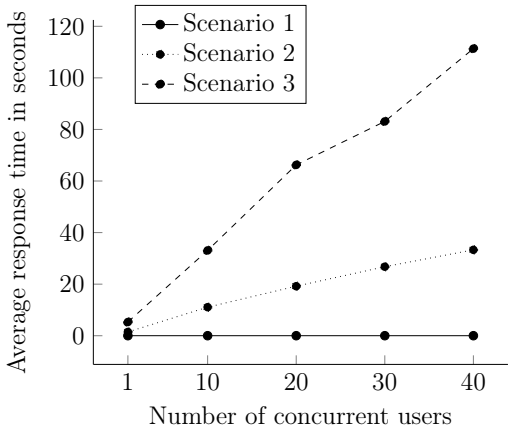


Figure 6.17: ATIB response times

and significantly increased execution times for our proof-of-concept ATIB implementation. For a general production use, further improvements require incorporation.

6.12 Security

Identity management is a security-relevant function. Therefore, the security of ATIB has also significant importance in the application landscape of a service provider. We review ATIB's security by the identification of different attacker types. Based on the adversaries, we define attack vectors and countermeasures. Furthermore, the attack tree methodology [164] is applied to closer elaborate on the illegal service consumption offence. This attack vector has a special meaning to the service provider as it targets the valuable goods. In our security review, we concentrate on the additionally introduced attack surface of ATIB. In particular, we do not elaborate on the self-sovereign identity solution's security posture.

6.12.1 Attacker Types

Two types of attackers can be distinguished when using ATIB. There are internal and external adversaries. The category of internal attackers comprises the user and the service provider because they are actively involved in a mutual relationship with the support of the identity broker. However, the service provider has no interest to circumvent ATIB's functionality. On the one side, ATIB is hosted in the service provider's organisational trust boundary. On the other side, the identity broker supports the applications of the service provider. In contrast, the user might have a malicious interest to by-pass the correct working of the implemented

Security Objective	Attack	Attacker	Countermeasures
Integrity	Verifiable claim spoofing	User, external	Encrypted, signed data exchange; access control
Integrity	Illegal service consumption	User, external	Encrypted, signed data exchange; access control
Privacy	Retrieval of session info	External	Encrypted data transfer
Privacy	Retrieval of usage statistics	External	Encrypted data transfer
Availability	Service interruption	External	Increased scalability

Table 6.2: Overview of attacks against ATIB

identity management processes. For instance, the attacker intends to gain extended privileges in the domain of the service provider. The external attacker category comprises individuals that are not involved in the relationship between the user and the service provider. An external adversary might be motivated to gain illegitimate access to a service provider application or extract knowledge about an authorised user. Overall, we see the threat from the user or an external attacker as most prominent for ATIB and evaluate it in the further sections.

6.12.2 Attacks and Countermeasures

An overview of attacks against ATIB is presented in Table 6.2. For hosting and using ATIB, the general security objective triad of confidentiality, integrity, and availability is important. Additionally, the security goal of privacy is of special interest to the user. ATIB must prevent the illegitimate modification of data, for instance, verifiable claims or configuration. Additionally, privacy must be respected to keep the user's application usage statistics and verifiable claim values private. The goal of availability is fundamental for consuming ATIB's service and the accessibility of the service provider's application.

We clustered the attacks and countermeasures according to the security objectives. Concerning the security objective integrity, the verifiable claim spoofing and the illegal service consumption are the main attack vectors. Verifiable claim spoofing targets the circumvention of claim verification processes and the receipt of a counterfeited attribute that does not reflect reality. Illegal service consumption targets the usage of service in an unauthorised manner. To counteract these threats, we implemented encrypted and signed data exchange protocols. Cryptographic protection measures are offered by the established identity and access protocols. Additionally, ATIB conducts signature verification of verifiable claims to determine illegitimate modification. Furthermore, communication encryption

ensures confidentiality and privacy with regard to session information and transferred data. Besides that, attacks to impede ATIB's availability also target the availability of all connected services. As countermeasures, we have implemented scalability solutions to defeat, for instance, resource exhaustion or flooding attacks.

6.12.3 Illegal Service Consumption Analysis

Illegal service consumption is the major concern of a service provider as ATIB guards its applications. ATIB supports the service provider to adopt self-sovereign identity. Furthermore, the best location of ATIB is within the organisational trust boundary of the service provider (cf. 6.6). Thus, the service provider is our major focus, and we concentrate our further analysis on this attack vector. To understand the illegal service consumption attack vector in detail and potentially required steps, we apply the attack tree methodology [164]. The respective attack tree is shown in Fig. 6.18. Our considerations scrutinise discrete ATIB functionality. General attack vectors that also apply for other identity management systems, for instance, identity theft, session take-over and authentication or authorisation by-passing, are not examined further. These sub attack vectors are denoted with a grey veiling in the attack tree figure. Existing literature [165] [166] sufficiently focus on these topics.

On the contrary, attribute spoofing is a significantly different attack vector in connection with ATIB and the self-sovereign identity ecosystem. The properties of a user are essential for service provisioning. Attribute forging, changing or exploiting the trust model and tampering with the name translation are the categories of different attribute spoofing strategies. Within the ensuing paragraphs, we describe these attack vectors.

6.12.3.1 Forge Attribute

The attack category forge attribute references approaches that target the counterfeiting of a verifiable claim. Comparable methodologies can be applied in the verifiable claim spoofing class. However, these attacks are targeted to the ATIB issuance facility for verifiable claims. The following methods may intent any issuer as preparation for providing such a claim to the entity that is running ATIB.

- **Manipulate Attribute Verification:** The issuer of a verifiable claim must run a verification procedure to validate the actual claim value. An attacker exploits the verification process to retrieve an attested attribute that does not correspond the reality. Thereby, data sources can be manipulated, or communication processes are redirected. The verification process depends on the attribute type.

- **Obtain Issuance Key:** A verifiable claim comprises a signature of the issuer to protect the content of the claim cryptographically. The public key identifies the issuer and can be used for verification. In case the corresponding private key is disclosed to the attacker, the adversary may issue arbitrary claims without any verification procedures.
- **Manipulate Verifiable Claim:** A verifiable claim that is legitimately issued to an attacker might be prone to manipulation. The claim is protected by a cryptographic signature. In case a vulnerable hash function is applied, the claim value or other metadata can be substituted by other values.

6.12.3.2 Change Trust Model

An additional attack vector is the alteration of the used trust model in ATIB. The modification of the trust model leads to a change in the subjective trust opinions towards the attestation issuers. The attacker could acquire illegitimate access to the ATIB instance to manipulate the trust model itself. Besides that, methods of social engineering [167] could be used to trick administrators into changing ATIB's trust model.

- **Reduce Threshold:** The acceptance rule for an attribute contains a threshold. A calculated trust score that is above the threshold is accepted by ATIB as a trustworthy attribute of the user. An attacker may reduce the threshold to a lower level or completely to 0. Thus, lower trusted attribute provider or even self-attested claims are accepted.
- **Add Trusted Attribute Provider:** ATIB stores the trusted attestation issuers and their ratings for the trust model. An attacker may add additional attribute providers or an issuer that is run by the adversary as well. Thus, ATIB permits attributes by these issuers.
- **Change Attribute Provider Composition:** The modification of the attribute provider composition, for instance, changing the trust function, impede with the established trust model. Thus, attestation issuers with error-prone verification procedures can be preferred. In conjunction, with an attribute forge attack on the respective issuer, illegitimate service consumption can be achieved.

6.12.3.3 Exploit Trust Model

In addition to the direct manipulation of the trust model, the exploitation of weaknesses of the implemented scheme can lead to an advantage for the attacker. Specific characteristics of the trust model are targeted.

- **Obtain Verifiable Claims from Artificial Identities:** A trust model may consider a certain quantity of unknown attribute providers as trustworthy. An attacker creates new issuers and attests itself properties. Thus, the adversary presents these verifiable claims during the authentication process.

6.12.3.4 Tamper with Name Translation

ATIB uses claim name translation to mediate between standards and protocols. An intervention in the name translating process poses an attack vector.

- **Change Claim Name:** An attacker by-passes access controls to ATIB and changes the name translation tables. In particular, a self-sovereign identity claim name will be translated to a different identity and access management protocol name. Additionally, the used attribute name has lower trust demands than the attacked property. As a result, ATIB calculates and accepts a low trust value but conveys the value as another high trustworthy attribute.

6.12.4 Synopsis

The security analysis of ATIB showed that an external adversary or the user has interest in undermining confidentiality, integrity or availability. In particular, the illegal service consumption objective may attract potential interest. As the attack tree analysis of this objective has shown, the trust engine and the namespace translator component of ATIB extend the attack surface of the identity broker. Thus, these components and ATIB overall is a security critical element within the service provider's application landscape.

Figure 6.18: Attack tree for illegal service consumption

6.13 Summary

In this chapter, we presented the Attribute Trust-Enhancing Identity Broker (ATIB) as an interoperability concept that mediates the communication between self-sovereign identity solutions and existing applications. Thereby, the identity broker targets several challenges for the service provider. Besides the existing application landscape, a multitude of different self-sovereign identity solutions, divergent trust in attribute providers and the issuance of verifiable claims belong to these obstacles. Furthermore, we described the underlying requirements that serve as the basis for the component-based architecture of the broker. The general concept of ATIB comprises the extraction of user's attributes from the identity wallet and convey them to the flow of established identity and access management protocols during authentication. The verifiable claims run through a trust evaluation process to determine their trustworthiness. To enable the overall functioning, the identity broker encompasses the components namespace translator, trust engine, protocol manager, self-sovereign identity manager and verifiable claim issuer. Additionally, external interfaces exist to communicate with the surrounding environments. Furthermore, we evaluated the deployment location of ATIB. An installation within the organisational trust boundary of the service provider is most favourable and does not undermine the user's self-sovereignty.

After presenting the architecture, we elaborate on the fulfilment of the requirements and the self-sovereign identity principles. Moreover, the technical architecture and implementation details are described. ATIB supports a simple and an extended trust module besides the name translation of email address and the user's name. In addition to that, uPort, Jolocom and HL Aries are supported as self-sovereign identity solutions. For application authentication, the OIDC and SAML2 protocol can be used. Furthermore, we outlined the respective authentication flows with the protocols. Additionally, we conducted a security analysis that starts with different attacker types. Here, the user and an external adversary are the prevalent actors. Their potential attacks and implemented countermeasures are further described. We use the attack tree approach to analyse the illegal service consumption attack that is, in particular, relevant for a service provider. ATIB demands specific protection for the additionally introduced components.

7 Case Study: Authentication with Self-Sovereign Identity

This chapter depicts the practical application of our identity broker ATIB within a case study [22]. Therefore, we describe the integration into different applications and provide certain usage statistics of ATIB.

7.1 Introduction

Building an application, software engineers naturally focus on the implementation of functional requirements that are the driver for the new project [168]. Fulfilment of requirements in the area of security as well as identity and access management is usually treated with a subordinate priority. There might also be the situation that these demands have not been recorded at all. Latest whilst security and compliance reviews, the used identity and access management schemes move to a higher priority. A similar situation may occur during the refactoring of existing applications. Within this process, the applied user store and authentication methods can be revised.

Developers have great flexibility to select an appropriate identity management solution. In the realm of isolated identity management, an application-specific user database with stored authentication credentials is a straightforward solution. This option might be advantageous during the development because fewer interdependencies towards other systems exist. Additionally, testers can create arbitrarily user accounts for test cases. However, latest in the production environment, a local user store provides a number of disadvantages. The application development team needs to implement secure registration and credential management processes. Furthermore, users require a secure password reset process.

Therefore, centralised and federated identity management schemes provide relief. Web application frameworks [139] may directly support widespread protocols, e.g. OIDC [141]. Additional, social login providers, for instance, Facebook or Google, offer code snippets to integrate their service as easy as possible. Besides that, an organisational owned identity provider may exist. On the same lines, the HPI offers its own HPI identity provider [169] that can be integrated with the OIDC protocol to other applications.

Within this chapter, we demonstrate the use of self-sovereign identity management solutions supported by ATIB. We present the integration to existing applications that are already served by other means of identity management to show the ease of transition.

7.2 Application Integration

We describe the practical integration of the ATIB User Interface and tele-TASK into ATIB. Furthermore, we outline the conceptual use of ATIB for OpenHPI.

7.2.1 ATIB User Interface

The ATIB User Interface represents the end user interface of the identity broker. Fig. 7.1 shows a screenshot. The web application presents some general information and news on the main page. Additionally, the web site displays usage instructions and contact information. Besides the publicly available information, the web application enables the user to obtain verifiable claims and run the underlying data verification procedures. Thereby, the user must authenticate at the ATIB User Interface.

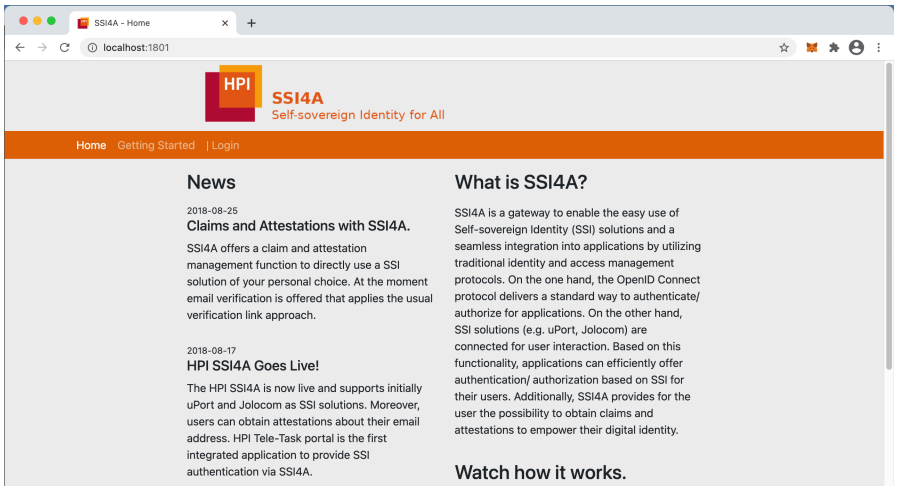
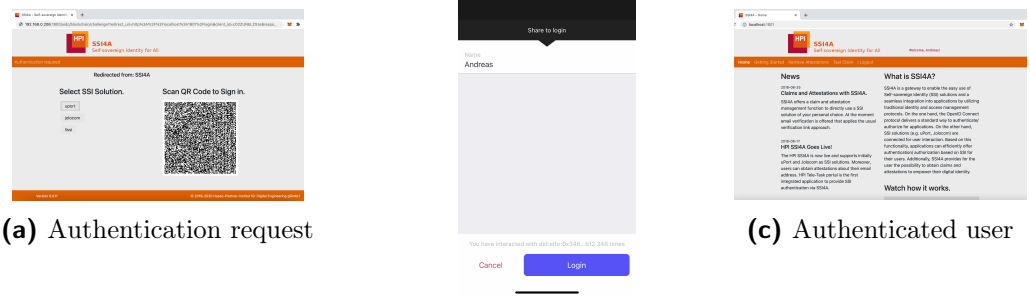


Figure 7.1: ATIB User Interface

The ATIB User Interface has a very limited dependence on attributes of the user. The only functionality that requires a user property is the salutation that is presented after successful authentication. This salutation is either composed of the name, first name, last name or first and last name if both properties are available.

There is no risk involved because the attributes are not used for access control. The properties solely serve for the welcome message. Thus, attribute values that do not match the reality can be accepted without a risk. In the worst case, the user would be welcomed with the wrong name. Table 7.1 illustrates the trust settings for the ATIB User Interface. The set of attribute providers encompass ATIB itself and the variable anonym that represents any provider that is not listed with a specific probability. Acceptance rules accept each property if trust function Θ calculates at least to 0. As a result, no trust is required for these peculiarities. Furthermore, the ATIB User Interface gracefully handles situations if the user does not provide the requested attributes. This necessity arises because assumptions about the user attributes may not hold true. However, the application itself must adapt to the situation. The authentication process with ATIB cannot solve this dependency. In particular, at execution time, it is unknown if the user has the required claims or consents to the disclosure.



(a) Authentication request

(b) uPort consent

(c) Authenticated user

Figure 7.2: ATIB User Interface authentication journey

The technical integration uses the OIDC protocol and follows the process description that is presented in Chapter 6.10.2. In Fig. 7.2, we show screenshots from the user's authentication journey. When selecting the login link on the ATIB User Interface, the redirection to the login screen occurs (a). Subsequently, we scan the QR code with the uPort identity wallet (b). As we only have the self-attested claim name, solely consent for this claim is requested. Finally, we are authenticated at the user interface (c).

7.2.2 tele-TASK

tele-TASK [170] is a video recording and streaming platform at the Hasso Plattner Institute (HPI). Thereby, tele-TASK offers various features to support the

Attributes	Providers	Acceptance Rules
$\mathbb{A} = \{name, firstname, lastname\}$	$\mathbb{P} = \{ATIB, anonym\}$	$\mathbb{S} = \{\Theta \geq 0 \Rightarrow name, \Theta \geq 0 \Rightarrow firstname, \Theta \geq 0 \Rightarrow lastname\}$

Table 7.1: Trust model characteristics for ATIB User Interface

user to track and collect their favourite series. Recorded streams encompass, for instance, lectures, seminars, conferences or videos of further occasions. Furthermore, selected videos are only available to a limited group of users. Successful authentication is required to use the personalisation features and to watch restricted streams.

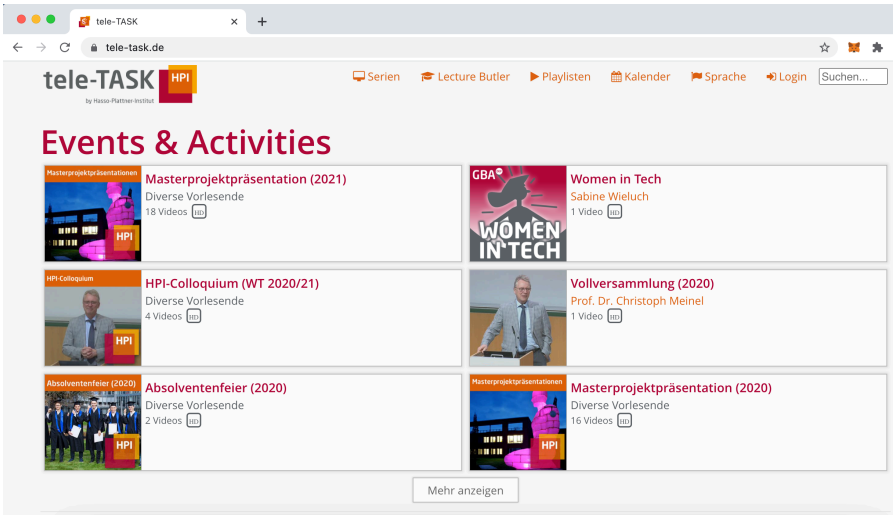


Figure 7.3: tele-TASK

The access model of tele-TASK distinguishes two user groups. These categories are any external user and persons that are associated with the HPI. The differentiation is bound to the domain of the used email address. Where an email address with HPI domain identifies the second group, any domain characterises the arbitrary external user. An attribute-based access control scheme derived from the email address is applied. Therefore, the email address is an important property of the user when authenticating at tele-TASK. Any further characteristic is not demanded. This setting is reflected in the configured trust rules within ATIB. Table 7.2 provides an overview of it. The known providers (ATIB and anonym) are comparable to the setting for the ATIB User Interface. On the contrary, the acceptance rule for the attribute email includes a threshold of 1. Thus, the highest

Attributes	Providers	Acceptance Rules
$\mathbb{A} = \{email\}$	$\mathbb{P} = \{ATIB, anonym\}$	$\mathbb{S} = \{\Theta \geq 1 \Rightarrow email\}$

Table 7.2: Trust model characteristics for tele-TASK

trust in the attribute is required. A self-attested claim is not sufficient in this scenario. Besides that, the technical integration uses the OIDC protocol as well.

7.2.3 OpenHPI

The OpenHPI [171] platform is an online learning system that offers Massive Open Online Courses (MOOCs) for all Internet users. Users must register and create a profile on the platform for participation in an online course. Subsequently to the registration, the user can enrol on learning courses. After successful completion of a track, a certificate is automatically issued to the user under its name.

Attributes	Providers	Acceptance Rules
$\mathbb{A} = \{email, name\}$	$\mathbb{P} = \{ATIB, anonym\}$	$\mathbb{S} = \{\Theta \geq 1 \Rightarrow email,$ $\Theta \geq 0 \Rightarrow name\}$

Table 7.3: Trust model characteristics for OpenHPI

During registration on OpenHPI, the users must provide their name and a email address. The validity and control of the user about the email address is proven by sending a verification email. On the contrary, the name is accepted without additional verification procedure. The user has a strong self-interest in providing a correct name because the certificates of course completion are issued to this name. Thus, no risk exposure is created for the OpenHPI platform. When transferring the attribute requirements into the trust module settings of ATIB, the acceptance rules for the properties name and email are different. The name attribute is permitted at a threshold of 0. Thus, self-attested claims are allowed. On the contrary, the email attribute demands high trust and is accepted at a trust level of 1. Similar to the other integrated applications, the provider ATIB and anonym exist.

7.3 Usage Statistics

We installed our proof-of-concept ATIB implementation on a production environment and made it available on the Internet in March 2019. For this ATIB instance, we reserved the domain ssixa.de and called it Self-Sovereign Identity for All. At the same time, we published the initial paper about ATIB's architecture

[24] at a conference. This version of ATIB offers solely uPort and Jolocom as possible authentication solutions. Thereby, uPort was one of the most mature and developed open-source self-sovereign identity solutions comprising a functioning identity wallet at this point in time. Nonetheless, the complete paradigm and the implementations were and are still in their infancy.

Furthermore, teams develop solutions and concepts energetically. For unknown reasons, later on, the progress seems to stop and the project becomes orphaned. For instance, the uPort identity wallet for iOS was regularly updated until the beginning of 2020[1]. Subsequently, the identity wallet might be abandoned, and the team starts the new project Veramo [172]. A similar situation exists for Jolocom. The identity wallet was continually updated until spring 2020. After a year, a compatibility breaking update was issued[2]. The discontinuity of the development and low maturity of the implementations lead to a not yet production-ready use for these self-sovereign identity solutions. The user creates an identity within a wallet and may lose access to the identity, e.g. due to unsolved bugs in the identity wallet. Furthermore, obtained verifiable claims are meaningless for the user.

Besides that, the service provider who applies these abandoned self-sovereign identity solutions faces challenges, too. Suppose the user interacts with the service provider using an identity, data sets are mapped to the identifier [17]. In case the user is not able to use its identity anymore or must create a new identity, the access to the service provider internal data is lost. Considering this situation, the landscape of self-sovereign identity solutions and their adoption is immature. We recommend to wait until the market gets more mature and functionally rich leading to a wider breakthrough in the usage.

Due to these challenges, we did not make ATIB for tele-TASK or openHPI publicly available as an identity provider. Nonetheless, ATIB and its user interface are accessible by any Internet user. For these elements, we can present certain usage statistics.

In Fig. 7.4 and Fig. 7.5, we outline data for creating authentication challenges with uPort and Jolocom. The numbers for uPort are highlighted in blue colour where the information for Jolocom is represented in red colour. Months without any activity are hidden in the diagram. In the year 2019, the overall number of created challenges is higher compared to the year 2020. Additionally, the statistics of uPort are continuously above the quantity of Jolocom. The default usage of uPort is a rationale for it. We observe a spike in the usage of ATIB in 2019 after publishing the corresponding paper at the conference.

[1]The information is based on the uPort iOS app version history (as of 2021-02-12).

[2]The information results from on the Jolocom Smartwallet iOS app version history (as of 2021-02-12).

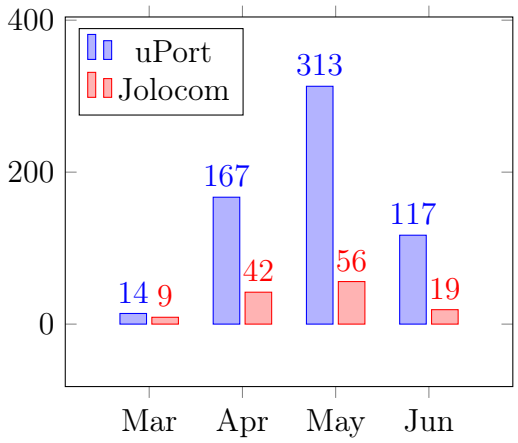


Figure 7.4: ATIB challenge creation statistics for year 2019

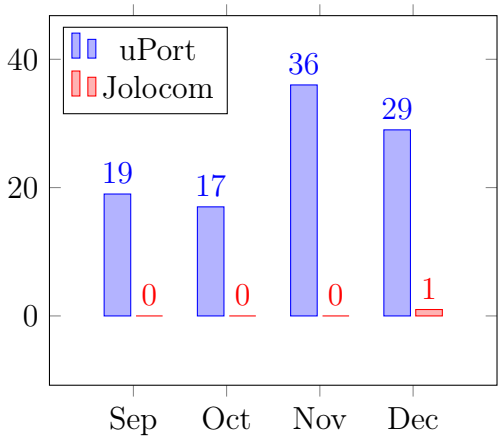


Figure 7.5: ATIB challenge creation statistics for year 2020

Fig. 7.6 presents the statistics for authentication at the ATIB User Interface. The number of authentications is below the number of created challenges.

7.4 Summary

In this chapter, we described the application of self-sovereign identity solutions for authentication and attribute-based access control supported by ATIB. We investigated the attribute requirements for the ATIB User Interface, tele-TASK and OpenHPI. Subsequently, we transformed the requirements into acceptance rules for the trust engine depending on their criticality for the functioning of each ap-

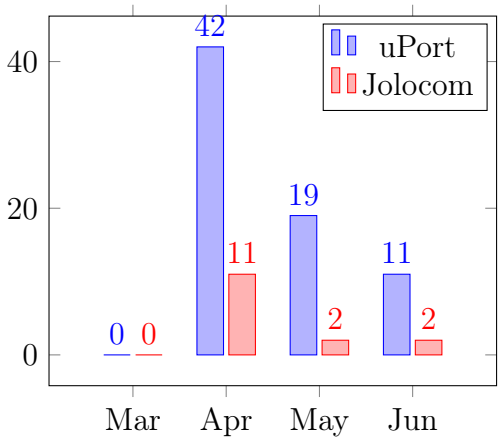


Figure 7.6: ATIB User Interface authentication statistics year 2019

plication. Finally, we presented usage statistics of the publicly available ATIB instance that runs under ssixa.de since March 2019.

8 Summary, Conclusion and Future Work

This chapter provides an overall summary of the thesis, conclusions of the work and topics for future research activities.

8.1 Summary and Conclusion

Identity management is a core component of the online service's security posture. It ensures legitimate access to restricted resources by the correct user. As online services are inexorably growing, the significance of identity management is increasing as well. The identity provider is the central actor to realise identity management processes. During the development of the different identity management models from isolated, over centralised to identity federations, the identity provider became a powerful actor that reflects a trusted third party. The user and the service provider must trust the identity provider in the domains privacy, authentication, credential management and attribute management. Therefore, the service provider and the user have a clear dependency on the identity provider. Actually, the user is the most discriminated party in this setting. To turn over a new leaf in identity management, Allen [11] postulated the principles of self-sovereign identity. Generally, these axioms cover the areas security, controllability and portability. Thus, the user should be brought back in control about its digital self. However, the identity provider, as a central authority stands in the way to achieve this novel sovereignty. With the parallel development of blockchain as a general execution platform for decentralised applications, a suitable implementation approach emerged. A dedicated blockchain or smart contracts in either the permissioned or unpermissioned model do not require a central authority to execute programs. As a result, a decentralised identity provider supports the self-sovereign identity paradigm. Accordingly, a major implementation barrier is removed.

Initially, we investigated the structure of self-sovereign identity and the respective decentralisation capability of blockchain. We concluded that identification, authentication, attributes and storage are the major components of a solution in this space. For the area of identification, blockchain can provide non-central identifier and claims registries. Self-authenticating schemes based on public key crypto-

graphy are usually applied. Besides that, verifiable claims represent attributes that can be attested by several entities. The flexibility to select user-defined storage solutions drives decentralisation as well. In addition to that, the execution environment and the supervising organisation can be decentralised with the support of blockchain. Program code is transparently executed by a peer-to-peer network. A central organisation does not exist at all in the unpermissioned case or is superseded by a diverse committee in the permissioned setting. In fact, the scope of a traditional identity provider is limited to the functions of a mere attribute provider.

The dissolution of the identity provider also leads to a shift in the required mutual trust between the user, service provider and the remaining attribute provider. We analysed the trust domains privacy, authentication, credential and attribute management with the support of patterns in the self-sovereign identity context. Thus, we could deduce that demanded trust in proper authentication and credential management is not required anymore. Additionally, trust in the attribute management can be limited by applying the trust-enhancing attribute aggregation methodology. Comparing with the traditional identity management models, the quantity and strength of trust requirements are reduced in the self-sovereign identity setting. Furthermore, the discriminated position of the user is improved, and therefore, the new paradigm delivers on its promise.

Subsequently, we examined trust models in attribute assurance to work towards a trust-enhancing attribute aggregation. In particular, we scrutinised the structure and assessed evaluation approaches. An attribute assurance trust model encompasses the attestation and trust network as well as a trust decision process where the last one includes but is not limited to a trust function and acceptance rules. Starting from security objectives and attack vectors, the security posture of the model depends on the shape of the trust function. We inferred that the trust function should consider high trust values and should not depend on a single or restricted group of attribute providers. Furthermore, we presented and evaluated the assessment approaches classification, conceptual analysis, practical study and simulation. The different strategies have their distinct virtues and limitations in investigating trust models. We described their impact on representation on the components and their analysis of characteristics.

Ensuing the foray in the trust model study, we built a theoretical concept for trust-enhancing attribute aggregation. Thereby, we construct a trust function based on an attribute's correctness and validity. The probability for both factors is combined to represent one attribute provider. Furthermore, the joint probability of several attribute providers results in an overall score. The previously investigated security properties hold true for the trust function. Additionally, we embedded this trust function into the self-sovereign identity context to build a complete

trust model that represents an intermediate approach between a web and chain of trust. As the last point, we classified the trust model in the taxonomy besides popular other sample schemes and devised its practical application as trust module in an identity broker.

A myriad of solutions, existing application landscape, non-adherence to established protocols and the new concept of verifiable claims threatens the usage of self-sovereign identity at the side of the service provider. We investigated a solution to apply the previously proposed trust function practically and to overcome these adoption challenges simultaneously. Thus, we devised an attribute trust-enhancing identity broker that abstracts from a single self-sovereign identity solution and mediates the communication with the OIDC and SAML2 protocol for integration towards other applications. The identity broker is constituted of several components, whereas the trust engine allows the implementation of various trust modules. A trust module specifies the conditions for the acceptance of verifiable claims as attributes. A verifiable claim issuance facility enables the service provider to attest its own claims. Furthermore, we analysed the most suitable location of the broker at the side of the service provider. This position also does not compromise the objectives of the self-sovereign identity principles. A security analysis based on attack trees showed the broker-specific attack vectors for illegal service consumption.

Eventually, we presented a case study for self-sovereign identity authentication and attribute-based access control with the support of the identity broker. Thereby, we described the representation of required attributes as trust rules and the integration into the broker. The selection of applications encompasses the broker user interface, tele-TASK and OpenHPI for demonstration purposes. The set of applications demands distinct properties of a user.

Overall, the blockchain-based self-sovereign identity concept can become a breakthrough in identity management. We have clearly devised the valuable change in trust requirements compared to the traditional models and therefore, the benefits for the user. On the contrary, established identity providers are reduced to mere attribute providers that can still issue verifiable claims. With the use of trust-enhancing attribute aggregation, a new approach emanates that enables a further trust reduction into the authenticity of properties and increased flexibility for choosing respective issuers. Even the trust in a single attribute provider is allocated to a larger group. The self-sovereign identity ecosystem is predestined for its implementation. Originating from the service provider, a scheme is possible that combines the advantages of the web and chain of trust model. The practical applicability and adoption are supported by our proposed identity broker without compromising self-sovereign principles for the user.

8.2 Future Work

Future research work encompasses manifold topics to investigate or to enable the self-sovereign identity paradigm as well as to drive the merging of this new concept and the existing application environment.

- **Bootstrapping of Trust:** The self-sovereign identity ecosystem is based on verifiable claims as attributes and decentralised identifiers as a designator of an identity. Attribute providers can attest a property under their decentralised identifier. To derive a trust score for the asserted claim, the respective ownership of the designator by an entity must be known. The bootstrapping of trust in the decentralised identifiers and therefore, in the complete ecosystem is a field of research. In the PKI based on X.509, the root certificates of trusted certificate authorities are pre-shipped with browsers and operating systems. Therefore, an individual trust decision of the user and even the knowledge about the trusted entities is not necessary. A browser simply shows a green or red icon to indicate trust for a user. For instance, a proposed solution to start trust in self-sovereign identity is the use of PKI certificates. However, the use of a PKI as the basis for a decentralised scheme may contradict its vision.

- **Practical Determination of Probabilities:** As the foundation for our proposed trust-enhancing attribute aggregation function serves the probabilities for validity and correctness as well as a provider-specific dependency factor. We see these values as subjective opinions towards the attribute provider and assumed the individual configuration by a relying party in the identity broker. However, the development of practical experiments and the gathering of realistic data to determine the probabilities can be researched. Additionally, methodologies for the definition of the dependency factor can be investigated.

- **Integration of Adverse Claims:** One of the foundations of the user's self-sovereignty and the trust modelling are verifiable claims. The user obtains, manages and provide the claims to the service provider. In case that claims are positive for the user, they enable it to interact with relying parties. Thus, the user is naturally incentivised to carry out its claim-related activities. On the contrary, an adverse claim attests a negative characteristic to a user. In this situation, the regular motivation of the user to obtain and provide this claim does not hold true. Additionally, the potential of whitewashing exist. The user drops it's current identifier and creates a new designator on the decentralised identity provider. Thus, the management, transfer and use of adverse claims in the entire setting require research.

- **Interoperability Concepts:** The rise of blockchain technology lead to the creation of plenty of self-sovereign identity solutions. These implementations compete for the grace of the user community and the service providers. It is unlikely that there will be a single winner to offer the preferred identity wallet, blockchain and tools. Additionally, there might be a specific rationale to establish several dedicated solutions. For instance, blockchain networks for identity management might be community-specific, owned by nations or treaty organisations. Nonetheless, interoperability of the different solutions is a desirable objective to achieve a breakthrough of the entire self-sovereign identity ecosystem. Our proposed identity broker is one approach for interoperability. Additionally, protocols and standards but also blockchain exchange methods, for instance, notary schemes and hashlocking [173], exist. The analysis and further development of interoperability approaches with regard to identity management impose an interesting research field.

- **Verifiable Claim Data Markets:** With the separation of the identifier from the attributes of an identity and the establishment of a blockchain network, a verifiable claim market can be established. Attributes of a user and its attestations become a tradable good. The service provider may start acting as an attribute provider. Relying parties may choose new trusted issuers, for instance, due to higher trustworthiness or lower costs. Research is required to determine implementation, motivations, incentives and practicability to foster such a market.

- **Trust Model Simulator:** As outlined in the Section 4.3 the simulation approach combines favourite characteristics compared to the conceptual and practical analysis. To systematically apply the simulation concept, a tool-based simulator demands research.

Glossary and Acronyms

AfA The Attribute for Acceptance (AfA) metric is a trust model characteristic.

AP The Attribute Provider (AP) verifies and issues properties of a user.

API An Application Programming Interface (API) is a communication interface to an application.

ATIB The Attribute Trust-Enhancing Identity Broker (ATIB) mediates communication between SSI solutions and web applications.

DID The Decentralised Identifier (DID) standard defines designators of identities in the SSI paradigm.

HL The Hyperledger (HL) project is a collection of blockchain frameworks.

IdP The Identity Provider (IdP) implements identity management processes.

OIDC OpenID Connect (OIDC) is an identity management protocol for authentication that is built on top of OAuth 2.0.

PGP Pretty Good Privacy (PGP) is a peer-to-peer email attestation scheme.

PKI A Public Key Infrastructure (PKI) encompasses hierarchical structured certificate authorities that issue certificates about principals.

QR The Quick Response (QR) code is a graphical encoding standard for information.

SAML2 The Security Assertion Markup Language in Version 2 (SAML2) is an identity management standard for authentication based on XML messages.

SP The Service Provider (SP) offers a service to user that requires identity management.

SSI The novel Self-Sovereign Identity (SSI) paradigm is based on principles that are postulated by Allen [11].

TfA The Trust for Acceptance (TfA) metric is a trust model characteristic.

UID A Unique Identifier (UID) unambiguously identifies an object in a realm.

URL The Uniform Resource Locator (URL) uniquely addresses an resource in the Internet.

VC A Verifiable Claim (VC) represents an attribute in the SSI paradigm.

W3C The World Wide Web Consortium (W3C) is a standardisation body.

X.509 X.509 is the Internet Public Key Infrastructure standard defined in RfC 5280.

Bibliography

[1] McKinsey. (2020) How covid-19 has pushed companies over the technology tipping point - and transformed business forever. [Online]. Available: https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and/-transformed-business-forever (accessed on 2022-01-05).

[2] Strategy&. Digitization for economic growth and job creation. regional and industry perspectives. [Online]. Available: https://www.strategyand.pwc.com/m1/en/reports/digitization-for-economic-growth-and-job-creation.pdf (accessed on 2022-01-05).

[3] M. E. Whitman and H. J. Mattord, *Principles of Information Security*, 6th ed. Boston, Massachusetts, USA: Cengage Learning, Inc., 03 2017.

[4] S. Samonas and D. Coss, "The cia strikes back: Redefining confidentiality, integrity and availability in security," *Journal of Information Systems Security*, vol. 10, no. 3, pp. 21–45, 2014.

[5] A. Preukschat and D. Reed, *Self-sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*, 1st ed. Shelter Island, New York, USA: Manning Publications, 06 2021.

[6] P. Steiner. (1993) On the internet, nobody knows you're a dog. [Online]. Available: https://www.plsteiner.com/cartoons#/newyorker (accessed on 2022-01-05).

[7] T. Hunt. (2019) Pwned passwords, version 5. [Online]. Available: https://www.troyhunt.com/pwned-passwords-version-5/ (accessed on 2022-01-05).

[8] J. Turner. (2020) Study reveals average person has 100 passwords. [Online]. Available: https://tech.co/news/average-person-100-passwords (accessed on 2022-01-05).

[9] G. Kontaxis, M. Polychronakis, and E. P. Markatos, "Minimizing information disclosure to third parties in social login platforms," *International Journal of Information Security*, vol. 11, no. 5, pp. 321–332, 2012.

[10] European Parliament and Council. (2016) EU General Data Protection Regulation (GDPR): Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ 2016 L 119/1. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj (accessed on 2022-01-05).

[11] C. Allen. (2016) The path to self-sovereign identity. [Online]. Available: http://www.lifewithalacrity.com/previous/2016/04/the-path-to-self-soverereign-identity.html (accessed on 2022-01-05).

[12] Deutsche Bundesbank. Finanzsanktionen. Allgemeine Informationen. [Online]. Available: https://www.bundesbank.de/de/service/finanzsanktionen/finanzsanktionen-609138 (accessed on 2022-01-05).

[13] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Transactions on Information and System Security*, vol. 18, no. 1, 2015.

[14] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf (accessed on 2022-01-05).

[15] A. Grüner, A. Mühle, T. Gayvoronskaya, and C. Meinel, "A comparative analysis of trust requirements in decentralized identity management," in *Proceedings of the 33rd International Conference on Advanced Information Networking and Applications (AINA). Advances in Intelligent Systems and Computing*, vol. 926. Cham, Germany: Springer, 2019, pp. 200–213.

[16] J. Sabater and C. Sierra, "Review on computational trust and reputation models," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 33–60, 2005.

[17] A. Jøsang, J. Fabre, B. Hay, J. Dalziel, and S. Pope, "Trust requirements in identity management," in *Proceedings of the 2005 Australasian Workshop on Grid Computing and e-Research (AusGrid)*, vol. 44. Darlinghurst, Australia: Australian Computer Society, 2005, pp. 99–108.

[18] U. Kylau, I. Thomas, M. Menzel, and C. Meinel, "Trust requirements in identity federation topologies," in *Proceedings of the IEEE 27th International*

Conference on Advanced Information Networking and Applications (AINA). Advances in Intelligent Systems and Computing, vol. 1. Los Alamitos, California, USA: IEEE Computer Society, 2009, pp. 137–145.

[19] N. Klingenstein, "Attribute aggregation and federated identity," in *Proceedings of the 2007 International Symposium on Applications and the Internet Workshops (SAINT)*. Los Alamitos, California, USA: IEEE Computer Society, 2007, pp. 26–26.

[20] M. Kuperberg, "Blockchain-based identity management: A survey from the enterprise and ecosystem perspective," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1008–1027, 2019.

[21] A. Grüner, A. Mühle, and C. Meinel, "Analyzing interoperability and portability concepts for self-sovereign identity," in *Proceedings of the 20th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trustcom)*. Shenyang, China: IEEE Computer Society, 2021, pp. 587–597.

[22] ——, "ATIB: Design and evaluation of an architecture for brokered self-sovereign identity integration and trust-enhancing attribute aggregation for service provider," *IEEE Access*, vol. 9, pp. 138 553–138 570, 2021.

[23] A. Grüner and C. Meinel, "On the structure and assessment of trust models in attribute assurance," in *Proceedings of the 35th International Conference on Advanced Information Networking and Applications (AINA). Advances in Intelligent Systems and Computing*. Cham, Germany: Springer, 2021, pp. 447–458.

[24] A. Grüner, A. Mühle, and C. Meinel, "An integration architecture to enable service providers for self-sovereign identity," in *Proceedings of the 18th IEEE International Symposium on Network Computing and Applications (NCA)*. Cambridge, Massachusetts, USA: IEEE Computer Society, 2019, pp. 1–5.

[25] ——, "Using probabilistic attribute aggregation for increasing trust in attribute assurance," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. Xiamen, China: IEEE Computer Society, 2019, pp. 633–640.

[26] A. Grüner, A. Mühle, M. Meinig, and C. Meinel, "A taxonomy of trust models for attribute assurance in identity management," in *Proceedings of the Workshops of the International 34th Conference on Advanced Information Networking and Applications (WAINA). Web, Artificial Intelligence and*

Network Applications, vol. 1150. Cham, Germany: Springer, 2020, pp. 65–76.

[27] A. Grüner, A. Mühle, T. Gayvoronskaya, and C. Meinel, "Towards a blockchain-based identity provider," in *Proceedings of the 12th International Conference on Emerging Security Information, Systems and Technologies (Secureware)*. Wilmington, Delaware, USA: IARIA, 2018, pp. 73–78.

[28] A. Grüner, A. Mühle, T. Gayvoronskaya, and C. Meinel, "A quantifiable trust model for blockchain-based identity management," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. Los Alamitos, California, USA: IEEE Computer Society, 2018, pp. 1475–1482.

[29] A. Mühle, A. Grüner, and C. Meinel, "Characterising proxy usage in the bitcoin peer-to-peer network," in *International Conference on Distributed Computing and Networking 2021*. New York, New York, USA: Association for Computing Machinery, 2021, pp. 176–185.

[30] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Computer Science Review*, vol. 30, pp. 80–86, 2018.

[31] P. Windley, *Digital Identity. Unmasking Identity Management Architecture*, 1st ed. Sebastopol, California, USA: O'Reilly Media, Inc., 08 2005.

[32] J. Lewis, "Enterprise identity management - it's about the business," *Datenschutz und Datensicherheit*, vol. 27, no. 9, 2003.

[33] International Telecommunication Union (ITU). Y.2720. ngn identity management framework. [Online]. Available: https://www.itu.int/rec/T-REC-Y.2720-200901-I (accessed on 2022-01-05).

[34] C. Tietz, C. Pelchen, C. Meinel, and M. Schnjakin, "Management Digitaler Identitäten: Aktueller Status und zukünftige Trends," Hasso Plattner Institute for Digital Engineering at the University of Potsdam, Potsdam, Germany, Tech. Rep. 114, 2017.

[35] E. Bertino and K. Takahashi, *Identity Management: Concepts, Technologies, and Systems*, 1st ed. Norwood, Massachussets, USA: Artech House, 12 2010.

[36] A. Jøsang and S. Pope, "User centric identity management," in *Proceedings of the 2005 Asia Pacific Information Technology Security Conference (AusCERT)*, 2005, pp. 77–89.

[37] D. W. Chadwick, *Federated Identity Management*. Berlin, Heidelberg, Germany: Springer, 2009, pp. 96–120.

[38] A. Bhargav-Spantzel, A. Squicciarini, and E. Bertino, "Trust negotiation in identity management," *IEEE Security & Privacy*, vol. 5, no. 2, pp. 55–63, 04 2007.

[39] Y. Cao and L. Yang, "A survey of identity management technology," in *Proceedings of the 2010 IEEE International Conference on Information Theory and Information Security (ICITIS)*. Beijing, China: IEEE Computer Society, 2010, pp. 287–293.

[40] C. Castelfranchi and R. Falcone, *Trust Theory: A Socio-Cognitive and Computational Model*, 1st ed. Chichester, UK: Wiley Publishing, 04 2010.

[41] T. Grandison and M. Sloman, "A survey of trust in internet applications," *IEEE Communications Surveys and Tutorials*, vol. 3, no. 4, pp. 2–16, 2000.

[42] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 03 2007.

[43] S. P. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, University of Stirling, 04 1994.

[44] D. H. McKnight and N. L. Chervany, "The meanings of trust," University of Minnesota, Carlson School of Management, Minneapolis, Minnesota, USA, Tech. Rep. MISRC 9604, 1996.

[45] I. Thomas, "A logic-based framework to enable attribute assurance for digital identities in servive-oriented architectures and the web," Ph.D. dissertation, Hasso Plattner Institute for Digital Engineering at the University of Potsdam, 07 2012.

[46] I. Thomas and C. Meinel, "An attribute assurance framework to define and match trust in identity attributes," in *Proceedings of the 2011 IEEE International Conference on Web Services (ICWS)*. Los Alamitos, California, USA: IEEE Computer Society, 2011, pp. 580–587.

[47] Kantara Initiative. Identity assurance framework. [Online]. Available: https://kantarainitiative.org/identity-assurance-framework/ (accessed on 2022-01-05).

[48] Office of the Management and Budget. E-authentication guidance for federal agencies. [Online]. Available: https://georgewbush-whitehouse.archives.gov/omb/memoranda/fy04/m04-04.pdf (accessed on 2022-01-05).

[49] Office of the e-Envoy, UK. (2002) Registration and authentication - e-government strategy framework policy and guidelines. [Online]. Available: https://ntouk.files.wordpress.com/2015/06/registration-authenticationv3.pdf (accessed on 2022-01-05).

[50] Government of Canada. Guideline on identity assurance. [Online]. Available: https://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=30678 (accessed on 2022-01-05).

[51] European Parliament and Council. Regulation (EU) No 910/2014 of the European Parliament and of the Council 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014R0910&from=DE (accessed on 2022-01-05).

[52] A. Narayanan and J. Clark, "Bitcoin's academic pedigree," *Communications of the ACM*, vol. 60, no. 4, pp. 36–45, 2017.

[53] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[54] B. Pillai, K. Biswas, and V. Muthukkumarasamy, "Blockchain interoperable digital objects," in *Proceedings of the 2nd International Conference on Blockchain (ICBC)*. Berlin, Heidelberg, Germany: Springer, 2019, pp. 80–94.

[55] A. Tobin and D. Reed. (2016) The inevitable rise of self-sovereign identity. the sovrin foundation. [Online]. Available: https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf (accessed on 2022-01-05).

[56] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Pro-*

ceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS). New York, NY, USA: Association for Computing Machinery, 2016, pp. 3–16.

[57] J. R. Douceur, "The sybil attack," in *Proceedings of the 2002 International Workshop on Peer-to-Peer Systems (IPTPS)*, vol. 2429. Berlin, Heidelberg, Germany: Springer, 2002, pp. 251–260.

[58] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. [Online]. Available: https://gavwood.com/paper.pdf (accessed on 2022-01-05).

[59] S. K. Lo, X. Xu, M. Staples, and L. Yao, "Reliability analysis for blockchain oracles," *Computers & Electrical Engineering*, vol. 83, p. 106582, 2020.

[60] Z. Wilcox-O'Hearn. (2001) Names: Decentralized, secure, human-meaningful: Choose two. [Online]. Available: http://www.zooko.com/distnames.html (accessed on 2019-05-01).

[61] Internet Engineering Task Force. (2008) Rfc 5280. internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. [Online]. Available: https://tools.ietf.org/html/rfc5280 (accessed on 2022-01-05).

[62] ——. (2004) Rfc 4122. a universally unique identifier (uuid) urn namespace. [Online]. Available: https://tools.ietf.org/html/rfc4122 (accessed on 2022-01-05).

[63] Namecoin. [Online]. Available: https://www.namecoin.org (accessed on 2022-01-05).

[64] Ethereum name service. [Online]. Available: https://ens.domains (accessed on 2022-01-05).

[65] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. (2016) uport: A platform for self-sovereign identity. [Online]. Available: http://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf (accessed on 2022-01-05).

[66] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *Proceedings of the 2016 USENIX Annual Technical Conference (USENIX ATC)*. Denver, Colorado, USA: USENIX Association, 2016, pp. 181–194.

[67] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, and M. Sabadello. Decentralized identifiers (dids) v1.0. core data model and syntaxes. [Online]. Available: https://www.w3.org/TR/did-core/ (accessed on 2022-01-05).

[68] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[69] M. Sporny, D. Longley, and D. Chadwick. (2019) Verifiable credentials data model 1.0. expressing verifiable information on the web. [Online]. Available: https://www.w3.org/TR/vc-data-model/ (accessed on 2022-01-05).

[70] Jolocom. (2018) Jolocom whitepaper. self-sovereign and decentralised identity by design. [Online]. Available: https://github.com/jolocom/jolocom-lib/wiki/Jolocom-Whitepaper (accessed on 2022-01-05).

[71] J. Benet. Ipfs - content addressed, versioned, p2p file system. [Online]. Available: https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf (accessed on 2022-01-05).

[72] The Linux Foundation. Hyperledger Indy. [Online]. Available: https://www.hyperledger.org/use/hyperledger-indy (accessed on 2022-01-05).

[73] The Sovrin Foundation. (2019) Sovrin governance framework v2. master document v2. [Online]. Available: https://sovrin.org/wp-content/uploads/Sovrin-Governance-Framework-V2-Master-Document-V2.pdf (accessed on 2022-01-05).

[74] M. S. Ferdous and R. Poet, "Analysing attribute aggregation models in federated identity management," in *Proceedings of the 6th International Conference on Security of Information and Networks (SIN)*. New York, New York, USA: Association for Computing Machinery, 2013, pp. 181–188.

[75] S. Ruohomaa and L. Kutvonen, "Trust management survey," in *Proceedings of the 3rd International Conference on Trust Management (iTrust)*. Berlin, Heidelberg, Germany: Springer, 2005, pp. 77–92.

[76] Z. Yan, P. Zhang, and A. Vasilakos, "A survey on trust management for internet of things," *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014.

[77] J.-H. Cho, K. Chan, and S. Adali, "A survey on trust modeling," *ACM Computing Surveys*, vol. 48, pp. 1–40, 2015.

[78] A. Jøsang, "An algebra for assessing trust in certification chains," in *Proceedings of the 1999 Network and Distributed Systems Symposium (NDSS)*. San Diego, California, USA: The Internet Society, 1999.

[79] W. Yang, C. Huang, B. Wang, T. Wang, and Z. Zhang, "A general trust model based on trust algebra," in *Proceedings of the 2009 International Conference on Multimedia Information Networking and Security (MINES)*, vol. 1. Los Alamitos, California, USA: IEEE Computer Society, 2009, pp. 125–129.

[80] J. Huang and D. Nicol, "A formal-semantics-based calculus of trust," *IEEE Internet Computing*, vol. 14, pp. 38–46, 2010.

[81] S. Ries, S. M. Habib, M. Mühlhäuser, and V. Varadharajan, "Certainlogic: A logic for modeling trust and uncertainty," in *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (Trust)*. Berlin, Heidelberg, Germany: Springer, 2011, pp. 254–261.

[82] A. Aldini, "A calculus for trust and reputation systems," in *Proceedings of the 2014 IFIP International Conference on Trust Management (IFIP TM)*. Berlin Heidelberg, Germany: Springer, 2014, pp. 173–188.

[83] M. Carbone, M. Nielsen, and V. Sassone, "A formal model for trust in dynamic networks," in *Proceedings of the First International Conference on Software Engineering and Formal Methods (SEFM)*, vol. 10. Los Alamitos, California, Germany: IEEE Computer Society, 2003, pp. 54–61.

[84] M. Kinateder, E. Baschny, and K. Rothermel, "Towards a generic trust model - comparison of various trust update algorithms," in *Proceedings of the Third International Conference on Trust Management (iTrust)*. Berlin, Heidelberg, Germany: Springer, 2005, pp. 177–192.

[85] M. Fragkakis and N. Alexandris, "Comparing the trust and security models of mobile agents," in *Proceedings of the Third International Symposium on Information Assurance and Security (IAS)*. Los Alamitos, California, USA: IEEE Computer Society, 2007, pp. 363–368.

[86] F. Moyano, C. Fernandez-Gago, and J. Lopez, "A conceptual framework for trust models," in *Proceedings of the 2012 Trust, Privacy and Security in Digital Business (TrustBus)*. Berlin, Heidelberg, Germany: Springer, 2012, pp. 93–104.

[87] C. Haydar, A. Roussanaly, and A. Boyer, "Comparing local, collective, and global trust models," *International Journal on Advances in Life Sciences*, vol. 6, 2014.

[88] D. Jelenc, R. Hermoso, J. Sabater-Mir, and D. Trček, "Decision making matters: A better way to evaluate trust models," *Knowledge-Based Systems*, vol. 52, pp. 147–164, 2013.

[89] M. Sel, "A comparison of trust models," in *Proceedings of the 2015 IEEE International Symposium on Systems Engineering (ISSE)*. Wiesbaden, Germany: Springer Fachmedien, 2015, pp. 206–215.

[90] F. Gomez Marmol and G. Martinez Perez, "Trust and reputation models comparison," *Internet Research*, vol. 21, 2011.

[91] M. Youssef, E. Abdeslam, and D. Mohamed, "A jade based testbed for evaluating computational trust models," in *Proceedings of the 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*. Rabat, Morocco: IEEE computer Society, 2015, pp. 1–7.

[92] K. K. Fullam, T. Klos, G. Muller, J. Sabater-Mir, K. S. Barber, and L. Vercouter, "The agent reputation and trust (art) testbed," in *Proceedings of the 2006 International Conference on Trust Management (iTrust)*. Berlin, Heidelberg, Germany: Springer, 2006, pp. 439–442.

[93] H. El Bakkali and B. I. Kaitouni, "A logic-based reasoning about pki trust model," in *Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC)*. Hammamet, Tunisia: IEEE Computer Society, 2001, pp. 42–48.

[94] ——, "A predicate calculus logic for the pki trust model analysis," in *Proceedings of the 2001 IEEE International Symposium on Network Computing and Applications (NCA)*. Los Alamitos, California, USA: IEEE Computer Society, 2001, pp. 368–371.

[95] Haibo Yu, Chunzhao Jin, and Haiyan Che, "A description logic for pki trust domain modeling," in *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA)*. Los Alamitos, California, USA: IEEE Computer Society, 2005, pp. 524–528.

[96] J. Huang and D. Nicol, "A calculus of trust and its application to pki and identity management," in *Proceedings of the 8th Symposium on Identity and Trust on the Internet (IDtrust)*. New York, New York, USA: Association for Computing Machinery, 2009, pp. 23–37.

[97] U. Maurer, "Modelling a public-key infrastructure," in *Proceedings of the 1996 European Symposium on Research in Computer Security (ESORICS)*. Berlin, Heidelberg, Germany: Springer, 1996, pp. 325–350.

[98] J. Marchesini and S. Smith, "Modeling public key infrastructures in the real world," in *Proceedings of the 2005 European Public Key Infrastructure Workshop (EuroPKI)*. Berlin, Heidelberg, Germany: Springer, 2005, pp. 118–134.

[99] M. Henderson, R. Coulter, E. Dawson, and E. Okamoto, "Modelling trust structures for public key infrastructures," in *Proceedings of the 2002 Australasian Conference on Information Security and Privacy (ACISP)*. Berlin, Heidelberg, Germany: Springer, 2002, pp. 56–70.

[100] R. Perlman, "An overview of pki trust models," *IEEE Network: The Magazine of Global Internetworking*, vol. 13, no. 6, pp. 38–43, 1999.

[101] Z. E. Uahhabi and H. E. Bakkali, "A comparative study of pki trust models," in *Proceedings of the 2014 IEEE International Conference on Next Generation Networks and Services (NGNS)*. Los Alamitos, California, USA: IEEE Computer Society, 2014, pp. 255–261.

[102] A. Ulrich, R. Holz, P. Hauck, and G. Carle, "Investigating the openpgp web of trust," in *Proceedings of the 2011 European Symposium on Research in Computer Security (ESORICS)*. Berlin, Heidelberg, Germany: Springer, 2011, pp. 489–507.

[103] Internet Engineering Task Force. (2007) Rfc 4880. openpgp message format profile. [Online]. Available: https://tools.ietf.org/html/rfc4880 (accessed on 2022-01-05).

[104] N. Alexopoulos, J. Daubert, M. Mühlhäuser, and S. M. Habib, "Beyond the hype: On using blockchains in trust management for authentication," in *Proceedings of the 2017 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trustcom)*. Los Alamitos, California, USA: IEEE Computer Society, 2017, pp. 546–553.

[105] A. Mohan and D. M. Blough, "AttributeTrust - a framework for evaluating trust in aggregated attributes via a reputation system," in *Proceedings of the 6th Annual Conference on Privacy, Security and Trust (PST)*. Los Alamitos, California, USA: IEEE Computer Society, 2008, pp. 201–212.

[106] H. Gomi, "Authentication trust metric and assessment for federated identity management systems," *IEICE Transactions on Information and Systems*, vol. 95-D, no. 1, pp. 29–37, 2012.

[107] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th International Conference on World Wide Web (WWW)*. New York, New York, USA: Association for Computing Machinery, 2003, pp. 640–651.

[108] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, California, USA, Tech. Rep. 1999-66, 11 1999.

[109] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[110] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," in *Proceedings Third International Conference on Peer-to-Peer Computing (P2P)*. Los Alamitos, California, USA: IEEE Computer Society, 2003, pp. 150–157.

[111] R. Alnemr and C. Meinel, "Why rating is not enough: A study on online reputation systems," in *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. Los Alamitos, California, USA: IEEE Computer Society, 2011, pp. 415–421.

[112] Amazon. About comments, feedback, & ratings. [Online]. Available: https://www.amazon.com/gp/help/customer/display.html?nodeId=201889150 (accessed on 2022-01-05).

[113] B. Hulsebosch, M. Wegdam, B. Zoetekouw, N. van Dijk, and R. P. van Wijnen. (2011) Virtual collaboration attribute management. [Online]. Available: https://www.surf.nl/binaries/content/assets/surf/en/knowledgebase/2012/EDS+11-06+Attribute+Management+v1.0.pdf (accessed on 2022-01-05).

[114] P. R. Zimmermann, *The Official PGP User's Guide*. Cambridge, MA, USA: MIT Press, 1995.

[115] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers & Security*, vol. 24, no. 1, pp. 31–43, 2005.

[116] A. Abdul-Rahman, "The pgp trust model," *Journal of Electronic Commerce*, 1997.

[117] M.-W. Dictionary. Definition of taxonomy. [Online]. Available: https://www.merriam-webster.com/dictionary/taxonomy (accessed on 2022-01-05).

[118] Apple Inc. (2019) List of available trusted root certificates in ios 13, ipados 13, macos 10.15, watchos 6, and tvos 13. [Online]. Available: https://support.apple.com/en-us/HT210770 (accessed on 2022-01-05).

[119] S. Čapkun, L. Buttyán, and J.-P. Hubaux, "Small worlds in security systems: An analysis of the pgp certificate graph," in *Proceedings of the 2002 Workshop on New Security Paradigms*. New York, New York, USA: Association for Computing Machinery, 2002, pp. 28–35.

[120] K. Fiskerstrand. Sks keyservers. [Online]. Available: https://sks-keyservers.net (accessed on 2022-01-05).

[121] M.-W. Dictionary. Definition of simulation. [Online]. Available: https://www.merriam-webster.com/dictionary/simulation (accessed on 2022-01-05).

[122] Internet Engineering Task Force. (2000) Rfc 2818. http over tls. [Online]. Available: https://tools.ietf.org/html/rfc2818 (accessed on 2022-01-05).

[123] A. Whitten and J. D. Tygar, "Why johnny can't encrypt: A usability evaluation of pgp 5.0," in *Proceedings of the 8th Conference on USENIX Security Symposium (SSYM)*, vol. Volume 8. Berkely, California, USA: USENIX Association, 1999, p. 14.

[124] S. Sheng, L. Broderick, C. A. Koranda, and J. J. Hyland. (2006) Why johnny still can't encrypt: Evaluating the usability of email encryption software. [Online]. Available: https://cups.cs.cmu.edu/soups/2006/posters/sheng-poster_abstract.pdf (accessed on 2022-01-05).

[125] D. Chadwick, G. Inman, and N. Klingenstein, "A conceptual model for attribute aggregation," *Future Generation Computer Systems*, vol. 26, no. 7, pp. 1043–1052, 2010.

[126] D. W. Chadwick and G. Inman, "The trusted attribute aggregation service (TAAS) - providing an attribute aggregation layer for federated identity management," in *Proceedings of the 2013 International Conference on Availability, Reliability and Security (ARES)*. Los Alamitos, California, USA: IEEE Computer Society, 2013, pp. 285–290.

[127] M. S. Ferdous, F. Chowdhury, and R. Poet, "A hybrid model of attribute aggregation in federated identity management," in *Proceedings of the 2nd International Workshop on Enterprise Security (ES)*. Cham, Germany: Springer, 2017, pp. 120–154.

[128] OASIS. Saml version 2.0. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf (accessed on 2022-01-05).

[129] H. Ouechtati and N. Ben Azzouna, "Trust-ABAC towards an access control system for the internet of things," in *Proceedings of the 12th International Conference on Green, Pervasive, and Cloud Computing (GPC)*. Berlin, Heidelberg, Germany: Springer, 2017, pp. 75–89.

[130] K. Yamaji, T. Kataoka, M. Nakamura, T. Orawiwattanakul, and N. Sonehara, "Attribute aggregating system for shibboleth based access management federation," in *Proceedings of the 10th IEEE Annual International Symposium on Applications and the Internet Workshops (SAINT)*. Los Alamitos, California, USA: IEEE Computer Society, 2010, pp. 281–284.

[131] J. Gemmill, J.-P. Robinson, T. Scavo, and P. Bangalore, "Cross-domain authorization for federated virtual organizations using the myVocs collaboration environment," *Concurrency and Computation: Practice and Experience*, vol. 21, no. 4, pp. 509–532, 2009.

[132] J. Watt, R. Sinnott, G. Inman, and D. Chadwick, "Federated authentication and authorisation in the social science domain," in *Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES)*. Los Alamitos, California, USA: IEEE Computer Society, 2011, pp. 541–548.

[133] A. B. Augusto and M. E. Correia, "OFELIA - a secure mobile attribute aggregation infrastructure for user-centric identity management," in *Proceedings of the 2012 IFIP International Information Security Conference (SEC)*. Berlin, Heidelberg, Germany: Springer, 2012, pp. 61–74.

[134] J. Jonczy, M. Wüthrich, and R. Haenni, "A probabilistic trust model for GnuPG," in *Proceedings of the 23rd Chaos Communication Congress (CCC)*, 2006, pp. 61–66.

[135] g10 code GmbH. Gnupg. [Online]. Available: https://gnupg.org/download/ (accessed on 2022-01-05).

[136] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[137] L. Fahrmeir, R. Künstler, I. Pigeot, and G. Tutz, *Statistik. Der Weg zur Datenanalyse*, 4th ed. Berlin, Heidelberg, Germany: Springer, 2003.

[138] I. Thomas, M. Menzel, and C. Meinel, "Using quantified trust levels to describe authentication requirements in federated identity management," in *Proceedings of the 2008 ACM Workshop on Secure Web Services (SWS)*. New York, New York, USA: Association for Computing Machinery, 2008, pp. 71–80.

[139] Tornado. [Online]. Available: https://www.tornadoweb.org/en/stable/ (accessed on 2022-01-05).

[140] C. Fisch, "Initial coin offerings (icos) to finance new ventures," *Journal of Business Venturing*, vol. 34, no. 1, pp. 1 – 22, 2019.

[141] OpenID Foundation. Openid connect core 1.0. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html (accessed on 2022-01-05).

[142] M. Sabadello, K. D. Hartog, C. Lundkvist, C. Franz, A. Elias, A. Hughes, J. Jordan, and D. Zagidulin. Introduction to did auth. [Online]. Available: https://github.com/WebOfTrustInfo/rwot6-santabarbara/blob/master/final-documents/did-auth.md (accessed on 2022-01-05).

[143] Universal Resolver. [Online]. Available: https://github.com/decentralized-identity/universal-resolver (accessed on 2022-01-05).

[144] The Linux Foundation. Hyperledger Aries. [Online]. Available: https://www.hyperledger.org/projects/aries (accessed on 2022-01-05).

[145] NIST. (2021) Glossary. Interoperability. [Online]. Available: https://csrc.nist.gov/glossary/term/Interoperability (accessed on 2022-01-05).

[146] R. Rezaei, T. Chiew, S. Lee, and Z. Shams Aliee, "Interoperability evaluation models: A systematic review," *Computers in Industry*, pp. 1–23, 2014.

[147] S. Koussouris, F. Lampathaki, S. Mouzakitis, Y. Charalabidis, and J. Psarras, "Digging into the real-life enterprise interoperability areas definition and overview of the main research areas," in *Proceedings of the 2011 World Multi-Conf. on Systemics, Cybernetics and Informatics (WMSCI)*, 2011, pp. 254–259.

[148] Kim Cameron. (2009) Seven Laws of Identity. [Online]. Available: https://www.identityblog.com/?p=1065 (accessed on 2022-01-05).

[149] M. Sabadello. (2020) Blockchain and identity. [Online]. Available: https://github.com/peacekeeper/blockchain-identity (accessed on 2022-01-05).

[150] OpenID Foundation. (2007) Openid authentication 2.0 - final. [Online]. Available: https://openid.net/specs/openid-authentication-2_0.html (accessed on 2022-01-05).

[151] B. Ingerson, C. C. Evans, and O. Ben-Kiki. Yet another markup language (yaml) 1.0. [Online]. Available: https://yaml.org/spec/history/2001-05-26.html (accessed on 2022-01-05).

[152] Internet Engineering Task Force. (2008) Rfc 5321. simple mail transfer protocol. [Online]. Available: https://tools.ietf.org/html/rfc5321 (accessed on 2022-01-05).

[153] ——. (2006) Rfc 4511. lightweight directory access protocol (ldap): The protocol. [Online]. Available: https://tools.ietf.org/html/rfc4511 (accessed on 2022-01-05).

[154] R. Hedberg. (2014) Pyoidc. [Online]. Available: https://pyoidc.readthedocs.io/en/latest/ (accessed on 2022-01-05).

[155] ——. (2011) Pysaml2. [Online]. Available: https://pysaml2.readthedocs.io/en/latest/ (accessed on 2022-01-05).

[156] OpenJS Foundation. Node.js. [Online]. Available: https://nodejs.org/en/ (accessed on 2022-01-05).

[157] Rinkeby. [Online]. Available: https://www.rinkeby.io (accessed on 2022-01-05).

[158] Internet Engineering Task Force (IETF). (2015) Rfc 7519. json web token (jwt). [Online]. Available: https://tools.ietf.org/html/rfc7519 (accessed on 2022-01-05).

[159] The Linux Foundation. Hyperledger aries. cloud agent python. [Online]. Available: https://github.com/hyperledger/aries-cloudagent-python (accessed on 2022-01-05).

[160] VON Community. Von. verifiable organizations network. [Online]. Available: https://vonx.io (accessed on 2022-01-05).

[161] International Standardization Organization. (2015) Iso/iec 18004:2015 information technology - automatic identification and data capture techniques - qr code bar code symbology specification. [Online]. Available: https://www.iso.org/standard/62021.html (accessed on 2022-01-05).

[162] Locust. a modern load testing framework. [Online]. Available: https://locust.io (accessed on 2022-01-05).

[163] Dan Kegel. The c10k problem. [Online]. Available: http://www.kegel.com/c10k.html (accessed on 2022-01-05).

[164] B. Schneier, "Attack trees," *Dr. Dobb's Journal of Software Tools*, no. 24, pp. 21–29, 1999.

[165] W. Burgers, R. Verdult, and M. van Eekelen, "Prevent session hijacking by binding the session to the cryptographic network credentials," in *Proceedings of 2013 Nordic Conference on Secure IT Systems (NordSec)*. Berlin, Heidelberg, Germany: Springer, 2013, pp. 33–50.

[166] M. Dalton, C. Kozyrakis, and N. Zeldovich, "Nemesis: Preventing authentication & access control vulnerabilities in web applications," in *Proceedings of the 18th Conference on USENIX Security Symposium (SSYM)*. Denver, Colorado, USA: USENIX Association, 2009, pp. 267–282.

[167] K. D. Mitnick and W. L. Simon, *The Art of Deception: Controlling the Human Element of Security*, 1st ed. Indianapolis, Indiana, USA: Wiley Publishing Inc., 10 2002.

[168] I. A. Tondel, M. Jaatun, and P. H. Meland, "Security requirements for the rest of us: A survey," *Software, IEEE*, vol. 25, pp. 20–27, 2008.

[169] Hasso Plattner Institute. Hpi identity provider. [Online]. Available: https://oidc.hpi.de (accessed on 2022-01-05).

[170] ——. tele-task. [Online]. Available: https://www.tele-task.de (accessed on 2022-01-05).

[171] ——. Openhpi. [Online]. Available: https://open.hpi.de (accessed on 2022-01-05).

[172] Veramo. Veramo. [Online]. Available: http://veramo.io (accessed on 2022-01-05).

[173] T. Koens and E. Poll, "Assessing interoperability solutions for distributed ledgers," *Pervasive and Mobile Computing*, vol. 59, pp. 1574–1192, 2019.

A Appendix

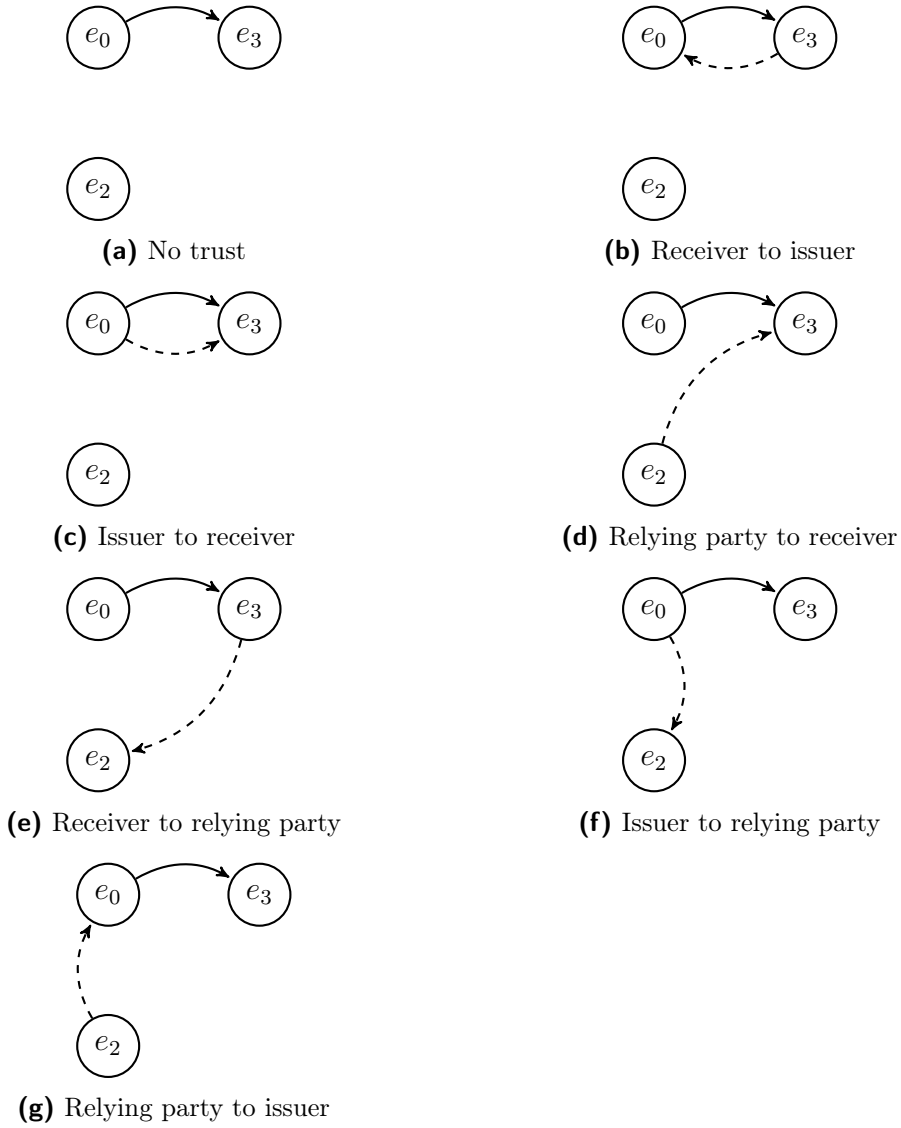
In this appendix, we present comprehensive information about potential trust formations, a component view of attribute assurance trust models, a proof concerning joining the probability of multiple attribute providers, the attribute aggregation algorithm and the code structure of ATIB.

A.1 Potential Trust Relationships

We investigated potential transformations from the attestation network to the trust network (cf. 4.2.4.2) and came to the conclusion that no obvious relations exist. Fig. A.1 highlights the potential transformations for a specific attestation. The node e_0 issues an attestation to entity e_3. Thereby, we also call the node e_3 receiver. The individual e_2 is the relying party in this scenario. Furthermore, we omit the attribute class and value for the assertion as well as the trust rating within the trust relationship. Additionally, we limit the visualisation to a single trust association to reduce complexity.

Under described conditions, the figure encompasses seven potential trust constellations for a single attestation. Despite an attestation from issuer to receiver, there might be no trust at all between the entities (a). Illustrated in (b), an inverse trust relation that is opposite to the attestation can exist. The receiver trusts the issuer. Besides that, a parallel trust affiliation along the assertion may exist (c). Therefore, the issuer also trusts the receiver. Furthermore, in (d), the relying party may trust the receiver of the attestation. Another potential trust relationship is shown in (e). The receiver of the attestation trusts the relying party. Subfigure (f) visualises a possible trust association that originates from the issuer to the relying party. Finally, the diagram in (g) depicts the trust of the relying party in the attestation issuer. The last relationship reflects the most commonly known and referred to association when investigating an attribute assertion.

Overall, this analysis of reduced potential trust constellations already delineates 7 different associations. Considering that a single assertion may result in multiple trust dependencies that can originate from various entities or the same node, the resulting quantity of associations would be multiplied. This situation illustrates the complexity of deriving a reasonable trust network from the attestation mesh.



(a) No trust

(b) Receiver to issuer

(c) Issuer to receiver

(d) Relying party to receiver

(e) Receiver to relying party

(f) Issuer to relying party

(g) Relying party to issuer

Figure A.1: Potential trust relationships

A.2 Component View of Trust Models

Trust models are composed of different elements. We described these components as attestation network, trust network and the trust decision process. The composed framework can be used to derive characteristics of a trust model. Besides that, it provides additionally a classification based on a definition view. Commonly proposed trust models may only define certain components and rely on other existing components. For instance, Jonczy et al. [134] proposes a trust function and relies on the other components, e.g. attestation network, on the PGP web of trust.

Trust Model Component		**PKI based on X.509**	**PGP**	**Logic-based Ass. Framew.**	**AttributeTrust**
AN	Nodes	√	√		
	Attestation	√	√		
TN	Nodes	√	√	√	√
	Relation	√	√	√	√
D	Function	√	√	√	√
	Trust Base	√	√	√	√
	Att. Base	√	√		√
	Acc. Rules	√	√	√	√

Table A.1: Component view of sample trust models

The elements of the sample trust models (cf. Chapter 4.3.1) are depicted in Table A.1. The PKI based on X.509 [61] and PGP [114] propose a complete trust model consisting of the attestation and trust mesh as well as a trust decision process. In contrast, Mohan and Blough's AttributeTrust extensively describes actors and the propagation of trust between the entities. The trust network relations are confidence paths. The trust function applies concatenation and aggregation of confidence values to determine an overall score for a specific attribute provider. Additionally, the existence of thresholds for acceptance is mentioned without a detailed elaboration. However, there is no specification of an attestation network. The authors assume the existence solely. Likewise, Thomas and Meinel's Logic-based Assurance Framework [45] comprises logical clauses to specify trust. Thus, a trust function and acceptance rules are determined. Indirectly, a trust network is delineated that is composed of the locally stored knowledge base. Nonetheless, the proposal completely omits the definition of an attestation network or the identification of the nodes.

A.3 Proof of Joining Multiple Providers

In Chapter 5.2.4, we described the composition of the trust function Θ by combining the probabilities of correctness and validity from several attribute providers. Thereby, we stated the final formula for n providers in Definition 5.8. To complete these considerations, we proof by mathematical induction the correctness of the theorem.

$$\forall n \geq 1 : \Theta(\mathscr{P}_{p_1} \vee \ldots \vee \mathscr{P}_{p_n}) = \sum_{i=1}^{n} \sum_{j=1}^{n-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} \tag{A.1}$$

- **Induction basis:** We show that the formula holds true for $n = 1$.

$$\Theta(\mathscr{P}_{p_1}) = \sum_{i=1}^{1} \sum_{j=1}^{1-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} = \sum_{j=1}^{1} (-1) \cdot \prod_{k=1}^{j} -\mathscr{P}_{p_k} = \mathscr{P}_{p_1} \tag{A.2}$$

- **Induction hypothesis:** We assume the correctness of the theorem for $n = m$.

$$\Theta(\mathscr{P}_{p_1} \vee \ldots \vee \mathscr{P}_{p_m}) = \sum_{i=1}^{m} \sum_{j=1}^{m-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} \tag{A.3}$$

- **Induction step:** Given the hypothesis, we show that the theorem holds true for $n = m + 1$.

$$\begin{aligned} \Theta(\mathscr{P}_{p_1} \vee \ldots \vee \mathscr{P}_{p_m} \vee \mathscr{P}_{p_{m+1}}) &= \sum_{i=1}^{m} \sum_{j=1}^{m-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} \\ &\quad + \sum_{i=m+1}^{m+1} \sum_{j=1}^{(m+1)-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} \\ &= \sum_{i=1}^{m+1} \sum_{j=1}^{(m+1)-i+1} (-1) \cdot \prod_{k=i}^{j} -\mathscr{P}_{p_k} \quad \square \end{aligned} \tag{A.4}$$

A.4 Attribute Aggregation Algorithm

In Chapter 5, we have built the theoretical foundation for trust-enhancing attribute aggregation and measured the performance of the algorithm. Fig. A.2 presents the implemented algorithm in pseudo code. The transferred attributes and the providers that attest their correctness serve as input parameters. The set of trusted and untrusted attributes are delivered as output of the computation. After providing the input, the algorithm iterates over each group of attribute and associated providers. The attribute is verified to be required in the application. Additionally, the providers are validated against the entities of the local trust store. Subsequently, the trust function Θ aggregates the probabilities. The result is compared against the threshold of the acceptance rule. If the outcome is above the threshold, the attribute belongs to the trusted set. Otherwise, the property is assigned to the untrusted class. Finally, the two result sets are returned and the application can decide on the usage.

Input: $C = \{\langle a_1; p_{1,1}, \ldots, p_{1,n}\rangle, \ldots, \langle a_m; p_{m,1}, \ldots, p_{1,k}\rangle\}$ with $a_i \in A^*, p_i \in P^*$

```
 1: function EvaluateTrust(C)
 2:     T ← ∅                                  ▷ Initialize set of trusted attributes
 3:     U ← ∅                                  ▷ Initialize set of untrusted attributes
 4:     for c = ⟨a_i; p_{i,1}, ..., p_{i,n}⟩ ∈ C do
 5:         if a_i ∈ A then
 6:             P' ← ∅
 7:             for i ← 1 to n do
 8:                 if p_i ∈ P then
 9:                     P' ← p_i
10:                 end if
11:             end for
12:             if Θ(P_{p_1}, ..., P_{p_o}) with p_i ∈ P' > t_{a_i} then
13:                 T ← a_i
14:             else
15:                 U ← a_i
16:             end if
17:         end if
18:     end for
19:     return T, U
20: end function
```

Output: T, U

Figure A.2: Attribute aggregation algorithm pseudo code

A.5 Code Structure of ATIB

We use PyCharm as the Python development environment for creating the code base of ATIB. Fig. A.3 shows the package structure. The base package comprises all code artefacts for the database (database) store, the self-sovereign identity wrappers (ssi). Additionally, the subpackage verification encompasses the verifier implementations. Despite that, the base package includes the routines for the SAML2 and the OIDC provider, including the Blockchain authentication method as well.

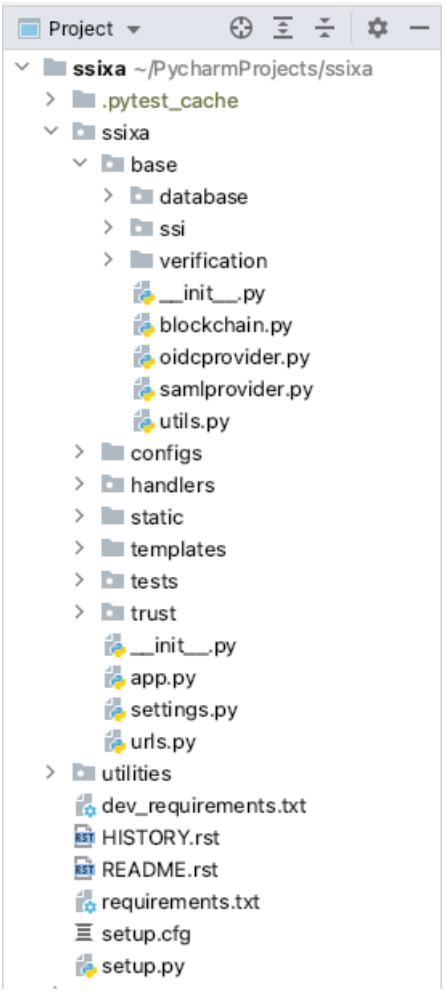


Figure A.3: ATIB code structure

The config folder collects all standard configuration files for the test environments. Furthermore, they serve as templates for newly created configuration files. The handler package consists of the classes for web request handling and repre-

sents, therefore, the entry point for accessing an URL of ATIB. Besides that, the static package comprises all static web content, and the folder templates encompass the dynamically filled web templates. The folder trust contains all classes for the trust engine and the trust modules. Furthermore, the test and utilities package encompasses test scripts and support tools for setting up ATIB.