Using Probabilistic Attribute Aggregation for Increasing Trust in Attribute Assurance

Andreas Grüner, Alexander Mühle and Christoph Meinel
Hasso Plattner Institute (HPI)
University of Potsdam, 14482, Potsdam, Germany
Email: {andreas.gruener, alexander.muehle, christoph.meinel}@hpi.uni-potsdam.de

***Abstract*—Identity management is an essential cornerstone of securing online services. Service provisioning relies on correct and valid attributes of a digital identity. Therefore, the identity provider is a trusted third party with a specific trust requirement towards a verified attribute supply. This trust demand implies a significant dependency on users and service providers. We propose a novel attribute aggregation method to reduce the reliance on one identity provider. Trust in an attribute is modelled as a combined assurance of several identity providers based on probability distributions. We formally describe the proposed aggregation model. The resulting trust model is implemented in a gateway that is used for authentication with self-sovereign identity solutions. Thereby, we devise a service provider specific web of trust that constitutes an intermediate approach bridging a global hierarchical model and a locally decentralized peer to peer scheme.**

***Index Terms*—Identity assurance, attribute assurance, digital identity, trust, identity management, attribute aggregation**

I. Introduction

Online services are an integral component of everyday life. These services are used pervasively in the business as well as the private sphere and provide personalized functionality. Additionally, the application of strong security measures is an inevitable necessity to build a trustful service for the user. An essential cornerstone for personalization and security is identity management. Digital identities and management processes enable the unique and reliable recognition of a person. Furthermore, the properties of a digital identity support service provisioning and attribute-based access control techniques.

In the beginning, identity management was distinct to each application and therefore one application was isolated from from the other. The advance of identity management leads to the development of centralized and federated models [1]. Within these schemes, the identity provider is a trusted third party in relation to the other entities for delivering accurate attributes [2]. Beyond that, blockchain technology enables the implementation of a decentralized identity provider that is generally referred to as a self-sovereign identity. Attributes are modelled as verifiable claims [3] that are comprised of claims and attestations from different attribute providers. Validity and correctness of these verifiable claims are of significant importance due to the impact on service provisioning.

Attribute aggregation approaches target the combination of attributes from different identity providers. A joint set of properties may completely fulfill requirements of a certain service provider where a single source does not comprise all demanded information. Despite the composition, the origin of each characteristic is a dedicated identity provider. Specifically, for each required attribute a contributing source is chosen. Therefore, each provider must be trusted for attribute management in a comparable manner to use a single source. Trust is required in the processes for attribute verification and issuance of the provider. Consequently, users and service providers significantly depend on the applied attribute providers.

A desirable development is the reduction of the described dependency and therefore, a limitation of the actual trust requirement for users and service providers [4]. We propose an attribute aggregation model that combines the same attribute offered by distinct attribute providers. Our approach is based on probabilities to increase trust in attribute assurance and to reduce the dependency towards one attribute provider. The proposed trust model can be applied to create a service provider specific web of trust where each participant considers different trust levels for a particular attribute provider and combinations thereof. Additionally, we implemented the trust model in a gateway for authentication with self-sovereign identity solutions by using the verifiable claim paradigm.

The remainder of the paper is organized as follows. In Section II, we present related research work in this domain. Subsequently, in Section III, we outline a motivating scenario and the objective of our work. Our novel attribute aggregation model is formally defined in Section IV. Furthermore, we describe the implementation of the trust model in an authentication gateway in Section V and evaluate it in Section VI. Finally, we discuss our results in Section VII and conclude our paper in Section VIII.

II. Related Work

Related research work is divided into two areas. The first field focuses on attribute aggregation from different identity providers. The second domain comprises definitions and models for identity as well as the attribute assurance of identity providers.

Attribute aggregation schemes denote the accumulation of attributes from different identity providers to complete a required set of attributes [5]. The inability of a single identity provider to deliver all required attributes is listed as the main rationale. Ferdous and Poet [6] provide a taxonomy of

attribute aggregation models in identity management. These patterns differentiate the location of the aggregation at either the identity provider, service provider or at the user's client. Chadwick et al. [7] describe in-depth a conceptual model for attribute aggregation through a linking service. The linking service is a component under control of the user that holds the credentials for digital identities at different identity providers. Additionally, the retrieval of the attributes and the service provider communication is managed.

Chadwick and Inman [8] outline a Trusted Attribute Aggregation Service (TAAS) that acts as an additional trusted third party and mediates the communication flow between the other actors. Further work considers hybrid aggregation services [9], focuses on privacy-preserving implementations [10], targets the Internet of Things [11] and considers specific protocols or use cases [12] [13] [14] [15] [16] [17]. Similarly, our approach aggregates attributes from different identity providers. However, we do not aggregate different attributes from distinct providers. On the contrary, our proposal collects the same attribute from different providers to increase trust.

Identity and attribute assurance determines trust in digital identities and their attributes. Thomas et al. [18] [19] [20] define an attribute assurance framework with a local trust database. Within the database, an identity provider is marked as trustworthy for specific attributes. On the service provider side, the database and a logical language are used to conclude on valid combinations of attributes and originating providers. Basically, a binary decision is made if an attribute is accepted from a certain provider.

The AttributeTrust [21] framework models the relationship between attribute consumers and providers as a graph-based network. Nodes represent users, service providers and attribute providers. Edges between the nodes represent confidence paths about attribute usage. Accumulated confidence paths enable a service provider to determine trust in an attribute of a specific provider. Besides that, governmental [22] [23] and non-governmental [24] organizations publish guidelines for identity and attribute assurance. Requirements for the level of assurance, for instance, attribute verification methodologies, are outlined to provide a common ground for trust.

In contrast to our approach, previous research specifies trust in an attribute of a particular attribute or identity provider. We aggregate trust of different providers in the same attribute to increase assurance about correctness and validity.

III. Motivating Scenario and Objective

The main actors in identity management comprise users, identity providers and service providers. Usually, the identity provider acts additionally as an attribute provider. The user and service provider rely on the attribute management of the identity provider. As a consequence, attribute verification and revocation processes must be effective to enable accurate service provisioning.

A service provider offers an online shop for ordering books. The online shop uses an external identity provider for identity and authentication services. Therefore, the user needs to register with the identity provider. During the registration process information about attributes, e.g. name, address and credit card is indicated by the user. At the same time, the identity provider has the obligation to verify the stated characteristics. After completing the enrollment process successfully, the digital identity can be used in the online book store. The user opens the corresponding web application and needs to sign in to finally place a book order. For authentication, the online book store redirects the customer to the identity provider. At the identity provider, the customer proves control of a credential that belongs to a specific digital identity. Subsequently, the user automatically returns to the online book store and is authenticated. In the act of redirecting, the identity provider also delivers the required attributes, e.g. name, address and credit card information, to the online shop of the service provider. Name and address support the correct delivery of the ordered goods to the customer. The credit card information enables billing. Hence, the attributes are essential for the service provider and the user to successfully accomplish the business transaction.

The outlined trust relationship implies a strong dependency of the user and the service provider towards the identity provider. Both parties require correct and valid attributes. A wrong name and address lead to false delivery of the book. Hence, the user and the service provider have to cope with the resolution. Erroneously entered credit card information leads to failed payments to the service provider. Additionally, wrong credit card information that is deliberately stated and insufficiently verified by the identity provider may lead to fraudulent activities. Furthermore, the user and the service provider are bound to a specific identity provider or to a small set of identity providers. In case the identity provider with the required set of attributes is not available or willingly rejects the service, the user and the service provider are prevented from conducting a business transaction. Selecting a group of attribute providers that are trusted in different combinations for specific attributes addresses this situation.

An identity provider, more specifically an attribute provider, can be realized in a strict hierarchical model or as a decentralized peer to peer scheme. In the hierarchical model, the identity provider is a trusted third party and solely responsible for properties. Such a scheme is applied by using a Certificate Authority (CA) for X.509 [25] certificates. A global renowned list of CAs is by default stored in browsers to evaluate trust. Nonetheless, a trust decision is binary to trust or not to trust a certain CA. This trust decision is individual but absolute to a specific actor. Pretty Good Privacy (PGP) [26] is a decentralized peer to peer approach that uses attestations of peers to prove the correctness of attributes, predominantly the ownership of email addresses. However, trust in the peers' confirmations are subjective to the evaluator and do not differentiate between different trust levels. The confirmation through a state agency implies the same trust as the confirmation of an arbitrary neighbor next to a user.

Overall, globally trusting one identity provider implies an

absolute dependency towards this entity. Additionally, the level of trust given by the trustor is subjective. Having confirmations by locally defined peers may not be enough if the service provider does not trust them. Therefore, to address either circumstance the main contribution of our paper targets the following points:

1) Reducing the dependency towards one identity provider with regard to attribute management by the user and service provider.
2) Enabling a service provider specific web of trust where service providers can trust varyingly strong different attribute providers and combinations of them with regard to a specific attribute.

IV. Attribute Aggregation for Increasing Trust in Attribute Assurance

In the following section, we outline a definition of trust that drives the understanding of assurance and connects the previously described dependency and negative consequences. Subsequently, we define a probabilistic model for trust in attributes and the aggregation of them. We will embed this model in a formal description of the context and provide samples for better understanding.

A. Definition of Trust

Trust is a widespread social and economic phenomenon that is a significant factor in decision making and a characteristic of personal relationships. Trust is very subjective in nature [27] and specific to a certain context. Therefore, a large variety of definitions in numerous disciplines were framed [28].

In our opinion, the definition of decision trust is most applicable. Josang et al. [28], based on previous work by McKnight and Chervany [29], formulate decision trust 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 clearly outlines that trust implies a dependency between different entities. A misuse of this dependency results in an adverse impact. In the previous sections, we described in detail the dependency between users, service providers and the identity provider in the context of attribute management. Additionally, we summarized potential negative consequences if the trust is misused in the motivating scenario.

B. Probabilistic Modelling of Trust in Attributes

Users and service providers greatly depend on attributes and therefore, trust the attribute provider. In our opinion, the dependency on attributes refers to correctness and validity [4]. Correctness specifies the representation of a true fact. For instance, if a digital identity of a user has the attribute *address*, the content of the attribute must reflect the true address where the user lives. Validity concerns whether the value of the provided attribute is already and remains valid. The point in time of user authentication, when the attribute is provided, must be within the validity period. Additionally, the attribute value must occur updated respectively and completely revoked in a timely manner if the underlying fact changes.

We model the correctness and validity of an attribute as random variables that have a binary outcome space. The outcome is *true* or *false* stating that the received property is (in)correct or (in)valid.

Definition 1 *Let $\boldsymbol{C}$ be a binary random variable reflecting the correctness of an attribute. $\mathbf{C}_{\mathbf{P}}^{\mathbf{A}}$ denotes the correctness of a particular attribute $\boldsymbol{A}$ from attribute provider $\boldsymbol{P}$. The outcome $\boldsymbol{c}$ of $\boldsymbol{C}$ (or $\mathbf{C}_{\mathbf{P}}^{\mathbf{A}}$) specifies a single attribute usage at a service provider with the following possible values:*

$\boldsymbol{c}$ = 1 implies that the attribute is correct
$\boldsymbol{c}$ = 0 indicates that the attribute is not correct

After defining correctness, we model the validity of an attribute in an analogue way.

Definition 2 *Let $\boldsymbol{V}$ be a binary random variable reflecting the validity of an attribute. $\mathbf{V}_{\mathbf{P}}^{\mathbf{A}}$ denotes the validity of a particular attribute $\boldsymbol{A}$ from attribute provider $\boldsymbol{P}$. The outcome $\boldsymbol{v}$ of $\boldsymbol{V}$ (or $\mathbf{V}_{\mathbf{P}}^{\mathbf{A}}$) specifies a single attribute usage at a service provider with the following possible values:*

$\boldsymbol{v}$ = 1 implies that the attribute is valid
$\boldsymbol{v}$ = 0 indicates that the attribute is not valid

Within the definitions, the usage of attributes at the service provider is referenced. During authentication, the identity provider asserts the digital identity of the user and acts additionally as attribute provider. We consider the transmitted assertion of the attributes and the subsequent usage as a random lottery. Each authentication and therefore conveyance of attributes is a new random event for the service provider. For each instance of the event, the correctness and validity of the transmitted attribute can change. We can express the quality of an attribute provider to assert attributes as respective probabilities and their distribution.

Definition 3 *Let $\mathbf{Pr}_{\mathbf{P}}^{\mathbf{A}}(\mathbf{C})$ the probability for correctness $\boldsymbol{C}$ of attribute $\boldsymbol{A}$ from attribute provider $\boldsymbol{P}$ with*

$\mathbf{Pr}_{\mathbf{P}}^{\mathbf{A}}(\mathbf{C}=\mathbf{1})$ = *probability that attribute is correct*

$\mathbf{Pr}_{\mathbf{P}}^{\mathbf{A}}(\mathbf{C}=\mathbf{0})$ = *probability that attribute is not correct*

Similarly, we define the probability for the validity of an attribute that is delivered by an attribute provider.

Definition 4 *Let $\mathbf{Pr}_{\mathbf{P}}^{\mathbf{A}}(\mathbf{V})$ the probability for validity $\boldsymbol{V}$ of attribute $\boldsymbol{A}$ from attribute provider $\boldsymbol{P}$ with*

$\mathbf{Pr}_{\mathbf{P}}^{\mathbf{A}}(\mathbf{V}=\mathbf{1})$ = *probability that attribute is valid*

$\mathbf{Pr}_{\mathbf{P}}^{\mathbf{A}}(\mathbf{V}=\mathbf{0})$ = *probability that attribute is not valid*

Using the probability of correctness and validity, we can evaluate an attribute provider for delivering a certain attribute. The combination of these probabilities contributes to an overall

likelihood for an attribute provider in the context of a particular characteristic. To describe the joined probability, we need to determine if the events for correctness and validity are independent or dependent.

The same organization or entity ensures both correctness and validity for one attribute at one provider. In case there are weaknesses in established processes or with the personnel one or the other characteristic is impacted. Additionally, if an attribute is correct, the validity of the property can be reasonably evaluated. In contrast, in case an attribute is false an adequate assessment with regard to validity is not rational. In our opinion, the events for correctness and validity of a property delivered by one provider are related and therefore, the probabilities are dependent. We can calculate the joint probability based on the following formula [30].

$$Pr_P^A(C \wedge V) = Pr_P^A(C) \cdot Pr_P^A(V|C)$$

Under the assumption that $Pr_P^A(C)$ and $Pr_P^A(V)$ is known, the conditional probability $Pr_P^A(V|C)$ needs to be determined to calculate the combined likelihood of both events. The conditional probability has a lower and upper bound [31].

$$Pr_P^A(C) \cdot Pr_P^A(V) < Pr_P^A(C \wedge V) < min(Pr_P^A(C), Pr_P^A(V))$$

The lower bound reflects the probability in case the events were independent. The upper bound is the minimum of either the probability of correctness or validity, in case the conditional event is given. We approximate the actual conditional probability by a function with a dependency factor. The approximation function is derived from [31]. The dependency factor reflects the density relation between correctness and validity at an attribute provider. It adjusts the conditional probability either closer to the lower bound or to the upper border.

Definition 5 *Let* $\mathbf{f_{d_P}}$ *be a conditional probability approximation function that is parametrized by* $\mathbf{d_P} \in [0,1]$ *for an attribute provider* ***P****. The factor* $\mathbf{d_P}$ *reflects the relation between correctness and validity at the attribute provider* ***P****.*

$f_{d_P}(Pr_P^A(C), Pr_P^A(V)) = Pr_P^A(C) + d_P(min(1, \frac{Pr_P^A(C)}{Pr_P^A(V)} - Pr_P^A(C))$ [31]

By using the function f_{d_P} the probability of receiving a valid and correct attribute A from an attribute provider P can be approximately calculated with the following formula.

$$Pr_P^A(C \wedge V) \approx Pr_P^A(C) \cdot f_{d_P}(Pr_P^A(C), Pr_P^A(V))$$

In a general setting, a service provider can interact with several attribute providers for the delivery of different attributes. Therefore, we interpret A as a set of attributes and P as a set of attribute providers. We can use the joint probability of correctness and validity for an attribute $a \in A$ of each provider $p_i \in P$ to increase trust in the overall delivery. In case a service provider can receive the same attribute a from a first attribute provider $p_1 \in P$ and a second provider $p_2 \in P$, we need to determine the overall probability of the occurrence of either event of both suppliers. The following function calculates the joined probability by using the approximation function.

$$Pr^a_{[p_1,p_2]}((C^a_{p_1} \wedge V^a_{p_1}) \vee (C^a_{p_2} \wedge V^a_{p_2})) \approx Pr^a_{p_1}(C^a_{p_1} \wedge V^a_{p_1}) + Pr^a_{p_2}(C^a_{p_2} \wedge V^a_{p_2}) - Pr^a_{p_1}(C^a_{p_1} \wedge V^a_{p_1}) \cdot Pr^a_{p_2}(C^a_{p_2} \wedge V^a_{p_2})$$

We can generalize the setting to an amount of n attribute providers $p_1, \ldots, p_n \in P$. The combined probability of all events is denoted as the following and is calculated in a pairwise sequential order that is comparable to two providers.

$$Pr^a_{[p_1,\ldots,p_n]}((C^a_{p_1} \wedge V^a_{p_1}) \vee \ldots \vee (C^a_{p_n} \wedge V^a_{p_n}))$$

In the subsequent sections we refer solely with $Pr^a_{p_1}$ and $Pr^a_{[p_1,\ldots,p_n]}$ to the probability of validity and correctness for delivering an attribute from a specific attribute provider respectively from several attribute providers.

C. Formally Modelling of Attribute Assurance Environment

For a service provider, an attribute assurance environment is a setting for accepting an attribute from an attribute provider or from combinations of attribute providers. We formally model an attribute assurance environment E_s for a service provider s as a set that consists of attributes A, attribute providers P, relations R between attributes and providers, a configuration set C and a function f. The function f is used to determine the level of trust in a specific attribute during an authentication process as described earlier.

$$E_s = \{A, P, R, C, f\}$$

The set of attributes A represents all attributes $a_1, \ldots, a_n$ that are required from a user by the service provider and an attribute specific threshold $t_{a_1}, \ldots, t_{a_1} \in [0,1]$. The threshold reflects a trust barrier. If the threshold is exceeded the trust in the attribute is enough and can be accepted. The higher the threshold the higher the required assurance in the attribute from the perspective of the service provider.

$$A = \{(a_1, t_{a_1}), \ldots, (a_n, t_{a_n})\}$$

The class of attribute providers P comprises all entities $p_1, \ldots, p_m$ that are trusted as an attribute provider by the service provider. Additionally, a correctness and validity relation factor $d_{p_1}, \ldots, d_{p_m} \in [0,1]$ belongs to each attribute provider.

$$P = \{(p_1, d_{p_1}), \ldots, (p_m, d_{p_m})\}$$

Usually, a service provider accepts attributes from a specific attribute provider. The class of relations R contains these dependencies. It depicts which provider can deliver a particular attribute.

$$R = \{(a, p) | a \in A \wedge p \in P\}$$

Besides that, the configuration C reflects a set of relations between the attribute provider its delivered properties and the probability of correctness and validity that is assumed by the service provider for this combination. The higher the assumed probability, the higher is the actual trust in the attribute provider for the respective attribute by the service provider.

$$C = \{((a,p), Pr_p^a(C), Pr_p^a(V)) | (a,p) \in R \wedge Pr_p^a(C) \in [0,1] \wedge Pr_p^a(V) \in [0,1]\}$$

Besides that, the service provider applies function f when evaluating the overall trustworthiness of a provided attribute. We define function f as the following:

$$f(a^*, [p_1^*, \ldots, p_n^*]) := Pr^{a^*}_{[p_1^*,\ldots,p_n^*]} \text{ with}$$
$$a^* \in A \wedge p_1^*, \ldots, p_n^* \in P \wedge \forall_i (a^*, p_i^*) \in R$$

The attribute a^* and provider $p_1^* \ldots p_n^*$ denotes the actually supplied attribute and respective provider verifications. The attribute must be a match to the required attribute set of the service provider. Moreover, the provider must be part of the accepted providers for the attribute. During authentication, an attribute is considered as trusted if the result of trust function f exceeds the threshold for the attribute.

$$\textit{Attribute a is trusted} \Leftrightarrow f(a^*, [p_1^*, \ldots, p_n^*]) > t_{a^*}$$

The environment E_s provides a local view on the trust setting of a specific service provider. A comparable environment can be specified for all service providers. By combining the environments of all service providers $s_1, \ldots, s_k$, we can formally specify a global view W as an overall trust model that is locally specific to each service provider.

$$W = \{E_{s_1}, \ldots, E_{s_k}\}$$

Each service provider can use different attribute providers for its required properties of the user. Furthermore, a particular service provider may assume different probabilities for correctness and validity as well as the dependency factor for an attribute provider. The values indicate the trust of the service provider towards the attribute provider. In addition to that, the threshold to accept a property can be chosen by a service provider depending on the criticality of its service. These characteristics enable a novel web of trust that originates from the trust demand of the service providers. However, this web of trust impacts the user as well. The user has more freedom to choose different attribute providers because of the service provider's flexibility in accepting them.

D. Sample Environment and Calculation

Having outlined a formal model for attribute assurance environments that are specific to each service provider and a trust evaluation function, we present an environment and calculation for a sample service provider s to illustrate functionality.

In Table I, the attribute assurance environment is listed for service provider s. The service provider requires three attributes a_1, a_2 and a_3 that are accepted by different thresholds. The attribute a_1 is most critical for the service provider and therefore, solely accepted by a value above 0.9. Further properties are admitted with a lower threshold. Additionally, the service provider maintains relationships to four attribute providers p_1, p_2, p_3 and p_4. For each attribute provider, the relationship factor between correctness and validity is specified with 0.8. Provider p_1 supplies the attributes a_1, a_2 and a_3. Furthermore, characteristic a_1 is additionally communicated by provider p_2. Attribute provider p_3 and p_4 solely deliver one attribute, a_2 and a_3 respectively. Finally, the row that specifies the configuration comprises the probabilities for correctness and validity from the view of the service provider towards the attribute providers.

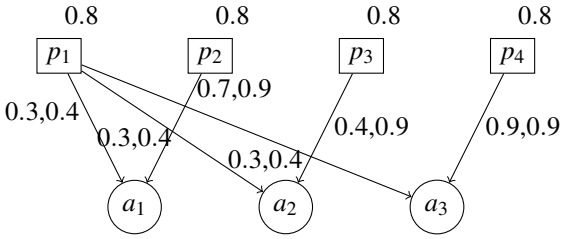

Fig. 1. Sample environment graph for service provider s

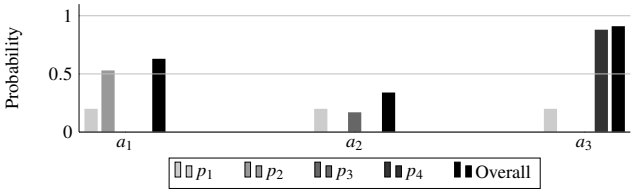

Fig. 2. Attribute assurance graph for service provider s

Class	Members
Attributes (A)	$(a_1,0.9)$, $(a_2,0.7)$, $(a_3,0.5)$
Attribute Providers (P)	$(p_1,0.8)$, $(p_2,0.8)$, $(p_3,0.8)$, $(p_4,0.8)$
Relations (R)	(p_1,a_1), (p_1,a_2), (p_1,a_3), (p_2,a_1), (p_3,a_2), (p_4,a_3)
Configuration (C)	$((p_1,a_1),0.3,0.4)$, $((p_1,a_2),0.3,0.4)$, $((p_1,a_3),0.3,0.4)$, $(p_2,a_1),0.7,0.9)$, $((p_3,a_2),0.4,0.9)$, $((p_4,a_3),0.9,0.9)$

TABLE I
SAMPLE ENVIRONMENT FOR SERVICE PROVIDER s

The attributes and their respective providers can be visualized as a graph-based network. In Fig. 1, the graph of the sample environment for service provider s is presented. Two different types of nodes exist. Rectangles denote an attribute provider and circles define an attribute. Connecting edges depict that a provider delivers a characteristic with the assumed probabilities.

A different viewpoint on the service provider and its trust in the attributes and respective attribute provider is shown in Fig. 2 as an attribute assurance graph. For each attribute, several bar charts are presented. An individual bar represents the probability of correctness and validity for an attribute if it is delivered by the corresponding attribute provider. The far right bar reflects the overall probability for the respective attribute in case all accepted attribute providers state the property.

E. Attribute Assurance Behavior

After presenting an attribute assurance environment, we describe further sample calculations to drive the understanding of how the overall probability for an attribute a adapts based on

the underlying probabilities of the different attribute providers. We calculate with a relation factor of 0.8 for all attribute providers. We consider two categories of providers. A small probability provider supplies an attribute with $Pr_p^a(C) = 0.1$ and $Pr_p^a(V) = 0.1$. A large probability provider delivers a property with $Pr_p^a(C) = 0.8$ and $Pr_p^a(V) = 0.8$. We define different scenarios where a number of small and large probability providers deliver the same attribute.

In Table II, the different scenarios are outlined and the overall probability is shown. In case 14 providers with small probability deliver an attribute, the overall trustworthiness is about 0.7. The value is lower if compared to 2 providers with large probabilities that total approximately 0.95. 14 large providers result in the same high probability as 7 large and 7 small providers. 2 small probability providers deliver the lowest overall trust of about 0.16.

Scenario	Small provider	Large provider	Overall
Sev. small	14	0	≈ 0.7
Sev. large	0	14	≈ 1
Sev. small/ large	7	7	≈ 1
Few small	2	0	≈ 0.16
Few large	0	2	≈ 0.95

TABLE II
SAMPLE CALCULATIONS

V. IMPLEMENTATION

In this section, we describe the implementation of the defined attribute assurance environment as a trust model into a self-sovereign identity (SSI) gateway. Therefore, we outline first background on the gateway and present afterwards present details of the trust model implementation.

A. Background on Self-sovereign Identity Gateway

The SSI gateway[1] connects blockchain-based identity management solutions e.g. uPort [32], with the protocol OpenID Connect (OIDC) [33]. It abstracts from a single SSI solution and its proprietary integration methods to enable service providers to use a standard protocol. An overview of the involved components of the gateway and the surrounding actors are provided in Fig. 3. The gateway is comprised of the SSI broker, trust engine with trust modules and a component to manage the OIDC authentication protocol.

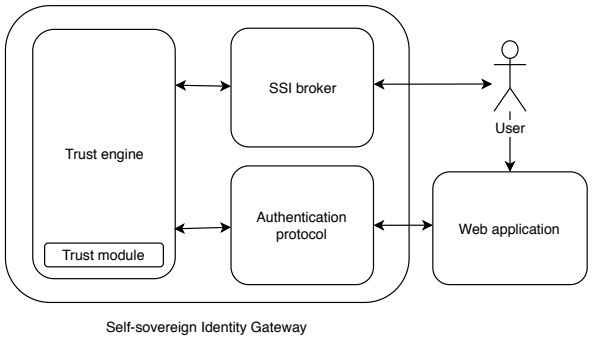

Fig. 3. Overview self-sovereign identity gateway integration

[1]The gateway is available under https://ssixa.de

The user starts the authentication process at the service provider's web application by selecting the SSI gateway as identity provider. Thus, the user gets redirected to the gateway following the OIDC protocol. Subsequently, the gateway manages the authentication with the SSI client of the user. This communication is controlled by the SSI broker component. Usually, the SSI client is implemented as a mobile application. The gateway provides an authentication challenge containing a random value and additionally requests the required attributes of the user. With the support of the SSI client, the user sends a response that includes the properties as verifiable claims. For one attribute, several verifiable claims from different providers can be returned. By using the trust engine, the attribute or a set of properties are evaluated for trustworthiness according to their issuers. The evaluation is conducted based on an implemented trust module. In case, the trust engine considers the attributes as trusted, these characteristics are returned within the authentication flow to the web application. If the process is successful, the user is authenticated at the web application.

B. Attribute Assurance Implementation as Trust Module

As described in the previous section, the trust module is the core element that evaluates the trustworthiness of attributes provided as verifiable claims. Within this subsection, we present the underlying database schema to reflect the attribute assurance environment and the algorithm for determining the trust in the attribute.

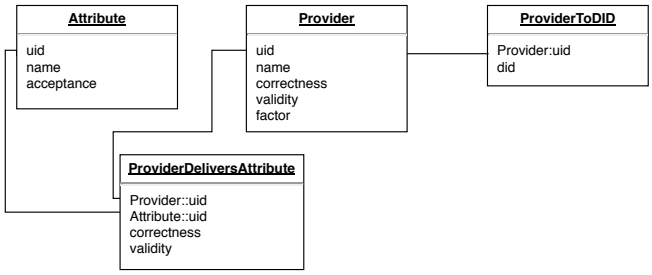

Fig. 4. Entity relationship diagram of database schema

The entity-relationship model of the database schema is presented in Fig. 4. An Attribute table acquires all relevant attributes that are needed by the service provider. Additionally, an acceptance level per attribute can be defined which serves as a threshold. The Provider table lists all attribute providers and the relationship table ProviderDeliversAttribute reflects the accepted attributes from the particular provider. Both tables contain fields to specify assumed correctness and validity values either being provider or property-specific. The table ProviderToDID stores the decentralized identifier (DID) [34] for the attribute providers. The DID standard defines a schema to address digital identities within the different SSI solutions. Basically, the SSI client provides the verifiable claims of the user for the requested attribute. The verifiable claims are issued by a digital identity that belongs to an attribute provider and that is referenced by a DID. This reference is resolved with the

support of the stored data to obtain the respectively assumed probabilities and further configuration.

In algorithm 1, the procedure for evaluating the assurance in the attribute and making the final trust decision is shown. The attribute assurance environment of a service provider is required as input parameter. Additionally, the actual attribute and its provider attestations that are supplied during an authentication process serve as further input parameters. During the evaluation, it is verified, that the provided attribute lies within the set of required attributes of the service provider. Besides that, a validation occurs that the attribute provider of the given attestations is trusted for the attribute. Subsequently, the probability of the property is calculated and compared against the attribute-specific threshold. In case the threshold is exceeded, the attribute is considered as trustworthy. Otherwise, the property is not trusted.

Algorithm 1 Evaluate attribute assurance at sp s

Input: a^*, $p_i^* \in P^*$ ▷ Attribute and its providers
Input: $E_s = \{A, P, R, C, f\}$ ▷ Attribute assurance environment
1: **procedure** EVALUATEASSURANCE(a^*,P^*,E_s)
2: **if** $a^* \in A$ **then**
3: $P^{**} \leftarrow \mathcal{O}$
4: **for** $p_i^* \in P^*$ **do**
5: **if** $p_i^* \in P$ and $(p_i^*, a^*) \in R$ **then**
6: $P^{**} \leftarrow p_i^*$
7: **end if**
8: **end for**
9: **if** $f(a^*, P^{**}) > t_a$ **then**
10: **return** a^* is trusted
11: **else**
12: **return** a^* is not trusted
13: **end if**
14: **end if**
15: **end procedure**
Output: a^* is trusted/ a^* is not trusted

VI. Evaluation

The SSI gateway and, therefore the trust module, is implemented in the Python[2] programming language by using the Tornado[3] web application framework. A virtual machine with 1024 MB main memory and one CPU having 2.4 Ghz clock rate serve as a test environment. The operating system of the virtual machine is Ubuntu 18.04[4]. The database model of the attribute assurance environment is implemented in a PostgreSQL[5] database that is co-located on the virtual machine.

We run several test scenarios to evaluate the execution time of the trust module under an increasing number of attribute providers with a different count of properties. The results are shown in Fig. 5. The solid line reflects the verification of one attribute that is delivered by up to ten attribute providers. Average execution time of 7 milliseconds (ms) is captured. The dotted line shows the evaluation of two attributes with a varying number of attribute providers. The mean execution time amounts to 15 ms. The last scenario covered the calculation for three attributes. The average processing time is 22 ms. We can see an increase in the processing time that strongly depends on the number of attributes that are verified. In contrast, the number of providers that deliver an attribute has no impact or an impact that is below the accuracy of the measurement. The execution time of the initial run for all scenarios is minimally higher than the next rounds. That seems to be caused by caching strategies of the database system.

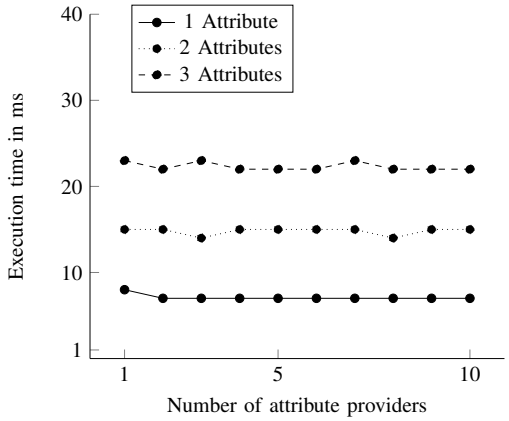

Fig. 5. Execution times

VII. Discussion

With the outlined methodology and the practical implementation of a trust module in the SSI gateway, we have shown an approach to remediate the dependency to one attribute provider and to enable a new web of trust originating from the service provider. However, there are some challenges within our approach.

From a theoretical point of view, the modelling of correctness and validity for an attribute largely depends on the underlying probabilities for the attribute providers. These probabilities need to be individually assumed and set per service provider. Besides that, the dependency factor needs to be determined per attribute provider in an analogue way. The determination of these numbers are subjective tasks at the side of the service provider.

In addition to that, from a practical perspective, our approach significantly relies on the pervasive adoption of SSI solutions and our gateway. The usage of verifiable claims decouples the identity from the actual attribute supply. Attribute providers need to foster the SSI ecosystem by providing verifiable claims for the different SSI solutions. Our SSI gateway with the corresponding trust module is an easy way to utilize an SSI solution for authentication and to apply the respective trust module. Without the adoption of these solutions, the described web of trust can hardly be realized.

[2]Python Software Foundation: https://www.python.org
[3]Tornado Web Framework: https://www.tornadoweb.org/en/stable/
[4]Ubuntu: https://www.ubuntu.com
[5]PostgreSQL: https://www.postgresql.org

VIII. Conclusion

In identity management, a strong dependency of the user and service provider towards the identity provider with regard to attribute management exists. We modelled the trust in an attribute as the probability of correctness and validity that is specific to a provider. The joint probability of several providers that are varyingly trusted increases the overall assurance in the attribute. We use this theoretic foundation to implement a trust module in an SSI gateway to reduce the dependency towards one specific attribute provider. At the same time, we enable a web of trust, originating from the service provider side, by applying the SSI gateway with differently configured trust modules for authentication at web applications.

References

[1] G. Williamson, D. Yip, I. Sharoni, and K. Spaulding, *Identity Management: A Primer*. MC Press Online, LP., 2009.

[2] 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 - Volume 44*, ser. ACSW Frontiers '05. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2005, pp. 99–108. [Online]. Available: http://dl.acm.org/citation.cfm?id=1082290.1082305

[3] M. Sporny and D. Longley. (2018) W3c community group draft report. verifiable claims data model and representations 1.0. [Online]. Available: https://www.w3.org/2017/05/vc-data-model/CGFR/2017-05-01/ [Accessed: 2019-05-18]

[4] A. Grüner, A. Mühle, T. Gayvoronskaya, and C. Meinel, "A comparative analysis of trust requirements in decentralized identity management," in *Advanced Information Networking and Applications*, L. Barolli, M. Takizawa, F. Xhafa, and T. Enokido, Eds. Springer International Publishing, 2020.

[5] 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: 2018-07-19]

[6] 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*, ser. SIN '13. ACM, 2013, pp. 181–188.

[7] D. Chadwick, G. Inman, and N. Klingenstein, "A conceptual model for attribute aggregation," *Future Generation Comp. Syst.*, vol. 26, pp. 1043–1052, 07 2010.

[8] D. W. Chadwick and G. Inman, "The trusted attribute aggregation service (TAAS) - providing an attribute aggregation layer for federated identity management," in *2013 International Conference on Availability, Reliability and Security, ARES 2013, Regensburg, Germany, September 2-6, 2013*, 2013, pp. 285–290.

[9] M. S. Ferdous, F. Chowdhury, and R. Poet, "A hybrid model of attribute aggregation in federated identity management," in *Enterprise Security*, V. Chang, M. Ramachandran, R. J. Walters, and G. Wills, Eds. Cham: Springer International Publishing, 2017, pp. 120–154.

[10] 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*, 11 2013.

[11] 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*, 05 2017.

[12] 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 Annual International Symposium on Applications and the Internet Workshops*, 07 2010, pp. 281–284.

[13] N. Klingenstein, "Attribute aggregation and federated identity," in *Proceedings of the 2007 International Symposium on Applications and the Internet Workshops*, 2007, pp. 26–.

[14] J. Gemmill, J.-P. Robinson, T. Scavo, and P. Bangalore, "Cross-domain authorization for federated virtual organizations using the myvocs collaboration environment," *Concurr. Comput. : Pract. Exper.*, vol. 21, no. 4, pp. 509–532, Mar. 2009.

[15] 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*, 08 2011, pp. 541–548.

[16] A. B. Augusto and M. E. Correia, "Ofelia - a secure mobile attribute aggregation infrastructure for user-centric identity management," in *Information Security and Privacy Research*, D. Gritzalis, S. Furnell, and M. Theoharidou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 61–74.

[17] D. W. Chadwick and M. Hibbert, "Towards automated trust establishment in federated identity management," in *Trust Management VII*, C. Fernández-Gago, F. Martinelli, S. Pearson, and I. Agudo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 33–48.

[18] I. Thomas and C. Meinel, "An attribute assurance framework to define and match trust in identity attributes," in *Proceedings of the 9th International Conference on Web Services*, 08 2011, pp. 580 – 587.

[19] ——, "An identity provider to manage reliable digital identities for soa and the web," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, ser. IDTRUST '10, 2010, pp. 26–36.

[20] ——, "Enhancing claim-based identity management by adding a credibility level to the notion of claims," in *Proceedings of the International Conference on Services Computing*, 2009.

[21] 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 2008*, 2008, pp. 201–212.

[22] P. A. Grassi, M. E. Garcia, and J. L. Fenton. (2017) Nist special publication 800-63-3. digital identity guidelines. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-63-3 [Accessed: 2019-05-02]

[23] T. B. of Canada Secretariat. (2016) Guideline on identity assurance. [Online]. Available: https://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=30678&section=HTML [Accessed: 2019-05-02]

[24] L. A. Project. (2007) Liberty identity assurance framework. version 1.1. [Online]. Available: http://www.projectliberty.org/content/download/4315/28869/file/liberty-identity-assurance-framework-v1.1.pdf [Accessed: 2019-05-02]

[25] Internet Engineering Task Force (IETF). Rfc 5280. internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. [Online]. Available: https://tools.ietf.org/html/rfc5280 [Accessed: 2019-05-18]

[26] A. Abdul-Rahman, "The pgp trust model," *EDI-Forum: the Journal of Electronic Commerce*, vol. 10, pp. 27–31, 1997.

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

[28] 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, Mar. 2007.

[29] D. H. McKnight and N. L. Chervany, "The meanings of trust," University of Minnesota, Tech. Rep., 1996.

[30] M. N. Das, *Statistical methods and concepts*. New age international publishers, 1989.

[31] 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*, ser. SWS '08. ACM, 2008, pp. 71–80.

[32] 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: 2018-07-19]

[33] OpenID Foundation. Openid connect core 1.0. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html [Accessed: 2019-06-05]

[34] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, and M. Sabadello. (2018) Decentralized identifiers (dids). data model and syntaxes for decentralized identifiers (dids). [Online]. Available: https://w3c-ccg.github.io/did-spec/ [Accessed: 2019-06-05]