An Identity Provider to manage Reliable Digital Identities for SOA and the Web

Ivonne Thomas
Hasso-Plattner-Institute for IT-Systems
Engineering
Prof.-Dr.-Helmert-Str. 2-3
D-14482 Potsdam
ivonne.thomas@hpi.uni-potsdam.de

Christoph Meinel
Hasso-Plattner-Institute for IT-Systems
Engineering
Prof.-Dr.-Helmert-Str. 2-3
D-14482 Potsdam
meinel@hpi.uni-potsdam.de

ABSTRACT
In this paper, we describe the implementation of our identity provider, based on open web service standards, which has been extended to distinguish between different qualities of identity attributes; therefore enabling a relying party to distinguish between verified and unverified digital identities.
Our contribution is the definition and representation of identity meta information for identity attributes on the identity provider side and the conveyance of this information as Identity Attribute Context Classes to a relying party. As a main result, we propose a format and semantic to include identity attribute meta information into security tokens which are sent from the identity provider to a relying party in addition to the attribute value itself.

Categories and SubjectDescriptors
K.6.5 [Management of Computing and Information Systems]: Security and Protection—Authentication

General Terms
Security

Keywords
SOA Security, Identity Management, Identity Provider, Attribute Management

1. INTRODUCTION
Digital Identity Management broadly refers to the establishment and controlled use of a persons "real-life" identity as digital identities in computer networks. Looking at the current online world, performing transactions as online banking, online shopping or communicating in social networks has become an inherent part of life. Hereby, personal, identity-related data plays a major role, since for many activities a service provider requires details about the identity of a user, be it to offer personalized services or to hold it liable in case anything bad happens. Examples include: the purchase of a good, that requires payment and delivery, or the provision of tailored recommendations based on the history of past purchases.
A digital identity usually comprises a limited set of attributes of a "real-life identity" that characterizes this entity (cf. also [23] or [7]). Unfortunately, managing numerous digital identities and associated authentication credentials is cumbersome for most computer users. Users do not only have difficulties to remember their passwords, they also bear a great burden to keep their account information up-to-date.
To overcome the limitations of the closed domain, open identity management models emerged as a way of sharing identity information across several trust domains in a controlled manner. The basic idea is having several places to manage a user’s identity data (so called identity providers) and to exchange identity attributes between entities holding identity information (the identity providers) and those consuming it (the relying parties). Open protocols and standards exists to exchange identity attributes as security tokens between identity providers and relying parties (cf. e.g. OASIS Identity Metasystem Interoperability specification 1.0 [19]).
Nevertheless, when we look at the Internet today, we still find an environment of mostly isolated domains. The reasons for the pre-dominance of the isolated model are comprehensible. Isolation allows organizations to retain control over their identity management systems. As organizations usually have different legal and technical requirements for identity management, they find it difficult to give up this control.
However, with regard to the Internet, we can find many identity attributes which do not require strong verification. Often the user can enter information into his account which does not require any verification. It really depends on what a digital identity is used for. If the user logs on to a site to prove on repeat visits that it is the same user, it does not matter whether his digital identity matches with his "real-life identity" as long as it is always the same digital identity he uses to log on. Only if critical transactions are performed, as ordering an item or paying for a service, the integrity of provided user data is required to hold the user liable in case anything bad happens. Current approaches for sharing identity data between domains as proposed by the open identity management models mainly considers the attribute value itself, but hardly how this value was collected or whether any
verification process took place.

In order to enable service providers to rely on information from a foreign source, an identity management for the Internet should be able to deal with attributes with a strong verification besides attributes without any verification which are managed by the users themselves. Moreover, it should allow a relying party (such as a service) to assess the value of received identity information in terms of correctness and integrity.

In Thomas et al. [22], we argued that this assessment should be done on the granularity level of the identity data – meaning, that the decision to trust should not only be made between the issuing and the relying party on a all-comprising level, but for each identity attribute, which is exchanged, separately. To give an example, we could consider a university which is trusted to make right assertions about whether a user is a student, but not about whether this user pays its telephone bills.

In this paper, we concentrate on the information required in addition to the attribute value itself to make right assertion about the credibility of an identity attribute. This meta identity information is all information additionally to the attribute value itself which enables a relying party to decide whether it trusts the received value with regard to an intended transaction. To be specific, we provide an identity provider which

- is based on open web service standards, such as WS-Trust, SAML and WS-Metadata-Exchange
- allows the definition of identity meta data and
- conveys identity meta data as so called Attribute Context Classes in SAML security tokens to a relying party

The rest of this paper is structured as follows. Section 2 shows how a scenario could look like which opens up current identity islands by using identity information from many sources across the Internet. In Section 3 we lay some foundations by giving a short introduction to claim-based identity management and the Identity Metasystem. It follows an overview of related work in the area of assurance frameworks as well as a discussion of their limitations in Section 4. After this, Section 5 introduces the trust model, that we use to identify and classify identity meta data that a relying party requires to assess identity information from a foreign source. Section 6 describes the implementation of our identity provider with regard to the definition and exchange of meta data between independent trust domains. In the centre of this section is our extension to the SAML 2.0 token format to convey meta information as part of the security token. Finally, Section 7 concludes the paper and highlights future work.

2. MOTIVATING EXAMPLE

Basically, we can make two observations with regard to the storage and administration of identity information on the Internet, today. The first observation is that basically every service provider on the Internet manages information which is specific to its domain, namely the information which was created during the interaction with a customer and the system, such as a customer number. A second observation is that information stored in independent domains is often redundant, because certain pieces of a subject’s identity are required by every service or web site provider. Examples include: the name and address of a person or its birthday. Hence, basically every service or web site provider has identity information, i.e. information about its user’s digital identities, which he could provide to other participants (given the user’s consent) and basically every service or web site provider also consumes certain information which it requests from the user and which it does not necessarily need to manage itself. A possible solution towards a more effective management of identity information is demonstrated in Figure 1. Instead of entering the same information into different user accounts, the user could reference to another account which already contains this information. For example, the newspaper publisher would receive the assertion that its customer is a student directly from the users university and the information about the user’s banking account information directly from the bank.

![Figure 1: Usecase showing independent identity domains and potential reliance on other Identity Providers](image)

3. BACKGROUND

3.1 Claim-based Identity Management

In order to implement a scenario such as introduced in Section 2, identity management concepts are required that take the decentralized nature of the Internet into account. Open identity management models evolved to address exactly this requirement. Instead of having isolated identity silos as with the traditional approaches, open identity management models are based on the idea of having several places to manage a users identity data (so called identity providers) and to share the identity information between these places and the places where this information is needed.

A concrete implementation of such an open identity management model offers the claim-based identity management. Claim-based identity management uses the notion of claims
to describe identity attributes. A claim is an identity attribute named with an abstract identifier (e.g. a URI), which applications and services can use to specify the attributes they need as for example a name or a user’s address. Given as a URI, claims provide a platform-independent way to present identity information and are well integrated into the open web service standards such as SAML [8], WS-Trust [15] or WS-Policy [6] which can be used to request and exchange identity information as claims.

3.2 The Identity Metasystem

As claim-based identity management provides interoperability among different identity systems, it is also used as one possibility to implement a related concept, the concept of an Identity Metasystem. Identity Metasystems provide an identity layer on top of existing identity systems and promise an easier management of digital identities among the Internet. This layer abstracts from concrete technologies and provides the necessary mechanisms to describe, exchange and distribute identity information across identity management solutions.

![Diagram](image)

Figure 2: Participants involved in the Identity Metasystem (FMC Block Diagram [12])

To do so, the Identity Metasystem distinguishes three different types of participants as denoted in Figure 2: the consumer of identity information (relying parties), authorities which manage and provide users’ digital identities (identity provider) as well as a component to choose a digital identity, called identity selector, and the user. In fact, putting the user in the center of all decision processes regarding his identity and creating a consistent and justifiable user experience belongs to the main principles of Identity Metasystems. These principles which explain successes and failures of identity management systems have been written down by Kim Cameron in the Laws of Identity [7].

The relying party is a service or Web site, which requires a certain set of user attributes / claims to perform a certain action. Instead of managing this information itself, it allows users to authenticate themselves at a federated identity provider and then relies on the assertion issued by this identity provider.

An identity provider (IdP) holds digital identities of registered users for the purpose of provisioning these identities, or portions of them, to a party willing to rely on this information (the relying party). Upon successful registration the identity provider issues a so-called Information Card, which holds all necessary meta data about the interaction between the user and the identity provider, including the URI to contact the IdP, the authentication to the IdP, the claims the IdP can assert as well as supported token types. It is important to note, that Information Cards do not contain any claim values, only the information how to connect to an identity provider to obtain asserted claims as security tokens.

Finally, the identity selector is a piece of software on the user’s system which handles the communication between the relying party and the identity provider and provides a consistent user interface to manage Information Cards. Upon request, the identity selector retrieves the policy of the relying party, matches the requirements with the Information Cards of the user and presents the user with a selection of suitable identity providers, from which he can choose. The identity selector takes care of performing the authentication procedure between the user and IdP (e.g. by requesting a password or digital signature) and sends an request for a security token to the identity provider. Upon successful authentication, the identity provider answers with a security token, which the user can use to prove his identity to the relying party.

4. RELATED WORK

The need to trust on information received from a foreign party is inherent to open identity management systems. If a relying party has to rely on identity information received from a foreign party, the need for assurance that the information is reliable is a natural requirement prior to using it. In order to address this need, several initiatives around the world have defined assurance frameworks which cluster trust requirements into different levels of trust. A level of trust or level of assurance (LoA) reflects the degree of confidence that a relying party can assign to the assertions made by another identity provider with respect to a user’s identity information.

4.1 Assurance Frameworks

In the area of authentication trust level, the UK Office of the e-Envoy has published a document called “Registration and Authentication – E-Government Strategy Framework Policy and Guideline” [20]. In this document the initial registration process of a person with the system as well as the authentication process for a user’s engagement in an e-government transaction are defined. Depending on the severity of consequences that might arise from unauthorized access, four authentication trust levels are defined, reaching from Level 0 for minimal damage up to Level 3 for substantial damage. The more severe the likely consequences, the more confidence in an asserted identity will be required when engaging in a transaction. For example, for filing an income tax return electronically, an authentication trust level of two is needed, which is reached when the client can present a credential (preferable a digital certificate) and can proof his right to that credential, e.g. by signing it with his private key.

The e-Authentication Initiative, another approach, is a major project of the e-government program of the US. The core concept is a federated architecture with multiple e-government applications and credential providers. In order to assist agencies in determining the appropriate level of identity assurance for electronic transactions, the initiative
has published a policy called "E-Authentication Guidance for Federal Agencies" (OMB M-04-04) [10]. The document defines four assurance levels, which are based on the risks associated with an authentication error. The four assurance levels reach from "little or no confidence in the asserted identity" to "very high confidence in the asserted identity".

In order to determine the required level of assurance, a risk assessment is accomplished for each transaction. Hereby, the potential harm and its likelihood of occurrence are identified. The technical requirements that apply for each assurance level are described in a recommendation of the National Institute of Standards and Technology (NIST), which is called "Electronic Authentication Guideline" (NIST 800-63) [17]. This document states specific technical requirements for each of the four levels for the token type, the authentication protocol as well as the types of attacks which need to be prevented.

A quite comprehensive approach that extends the OMB/NIST levels has been proposed by InCommon, a federation of more than 100 members from industry, government and the higher education sector [11]. InCommon uses the Shibboleth specifications and defines an Identity Assurance Assessment Framework. Aspects covered are Business, Policy and Operational Factors, Registration and Identity Proofing, Digital Electronic Credential Technology, Credential Issuance and Management, Security and Management of Authentication Events, Identity Information Management, the Identity Assertion Content as well as the Technical Environment.

Further approaches have been developed as part of the Liberty Alliance project's Identity Assurance Framework [1] as well as in the context of the European Stork project [5].

4.2 Limitations

Current approaches for assurance frameworks as described in the previous section provide a comprehensive assessment for identity providers by defining(gathering) trust requirements with regard to all the processes, technologies, technical infrastructure and further protection in place that have an influence on the degree of confidence into the assertion's contents made by an identity provider. The result is a global trust semantics, which allows a classification of identity providers with respect to different levels of trust. Such a classification can serve as the input to policy frameworks as well as a base for contracts and inter-organizational agreements.

Although current approaches provide a quite comprehensive assessment, a number of limitations exists. Existing assurance frameworks mostly refer to the identity as a whole, but do not refer to trust requirements of specific attributes. It is for example not possible to distinguish between self-asserted attributes an identity provider might manage besides attributes that were verified. Especially with regard to platforms of non-institutional providers such as Facebook, users often prefer using pseudonyms when acting in these communities. In fact, in blogs and forum discussions, anonymity of users is a frequent requirement. Also for over-18-services, anonymity of the users often is in favor while at the same time a verified assertion of a user's age is required. For these purposes, an identity provider could manage self-asserted attributes besides verified attributes. When doing so, reflecting these differences in the assertions is a major requirement.

Also, using existing assurance frameworks, it is hard to reflect possible changes of a user's identity trust level over time. As identity proofing processes are cost-intensive and time-consuming due to the effort required to verify a user's identity attributes, a verification of an attribute might not be desired as long as a user is not involved in transactions that demand a higher trust level. Therefore a user might decide to register with an identity provider without proper identity proofing, having for example his/her name self-asserted and getting involved in the identity proofing only upon concrete requirement. This requires a different trust level per user and does not allow to rate an identity provider as a whole.

Furthermore, identity providers are inherently different due to their affiliation with an organization or institution and might be suitable for asserting certain identity attributes only to a limited extent. For example, a banking identity provider will be in particular suitable to assert that a user can pay for a certain service, but might have weak records of the user's status as a student while for a university's identity provider it would probably be the opposite. In fact, such a diversity of identity provisioning sources is intended in the user-centric model which aims at reflecting the way identities are managed in the real world.

Taking all these facts into account, current approaches are likely to work for federations in which members have similar trust requirements, but are less likely to work when applied to the open market and user-centric models.

In our approach we aim at providing identity meta information for identity attributes in order to allow an identity provider to manage a mix of verified and not-verified attributes and more importantly in order to enable a relying party to distinguish between these different qualities of trust.

4.3 Levels of Assurance for Attributes

Work regarding trust levels for attributes has been conducted by Chadwick et al. in [9], Chadwick et al. build on NIST's concept of assurance levels. Similar to our work, they propose to have separate metrics for identity proofing processes (expressed in the Registration LOA) and the authentication of a subject (expressed in the Authentication LOA). Authentication LOA and Registration LOA are combined to a Session LOA and sent in each assertion from an identity provider to a service provider. Compared to this, our work is targeted more towards the relying party site. In our work, we aim at providing more choices for a relying party's access control decisions by conveying not only a trust level, but also trust-related information to be evaluated during access control. For this purpose, we propose to extend existing protocols by so called Attribute Context Classes that contain, besides a basic trust level, further meta data to enable the relying party to assess the trustworthiness of the received information.

5. A LAYERED TRUST MODEL

This section presents our trust model used by a relying party such as a service provider to accept identity information from a foreign partner and to perform access control decisions based on the received information. In this model we basically distinguish between two types of trust. First, a trust relationship is required between the service provider and the identity provider in order to trust the correctness of the assertions and second, for a concrete transaction, the
service provider has to decide whether the identity-based information in the assertions are sufficient to reach a certain trust level which is required to perform the request. While in the first case, the trust relationship is of a long-running kind, the trust establishment in the second case is part of identity-based access control mechanisms. We call the first kind of trust, organizational trust and the second kind identity trust. The following section gives a detailed characterization and comparison.

5.1 The Concept of Organizational Trust

Organizational trust refers to the quality of the trust relationship between the participants of a SOA or web-based scenario. When service consumers and service providers are located within the same trust domain, registration, authentication and management of participants happen under the same administrative control and are, therefore, usually fully trusted. However, with regard to cross-organizational scenarios involving services from different organizations, trust between the participants of a SOA is not given per default. Models for identity management as federated identity management establish cross-organizational trust by setting up federation agreements and contracts to extend the trust domain of an organization to the federation. Having a federation or not, whenever organizational borders are crossed, the question of whether the partner is trusted arises. Factors as past experience, the minimum trust settings for, for example, registration and authentication of users or the reputation of a company are important properties to assess the trustworthiness of the potential business partner. Also, the kind of business relationship is an important factor. A B2B relationship is usually more trustworthy than a B2C relationship due to contracts which manifest certain obligations and procedures of the business partners. In order to classify different qualities of trust relationships, assurance frameworks exist to help business partners to assess their identity management services. (cf. Section 4). However, a detailed assessment is not always feasible. Sometimes the decision to trust is founded on much fewer assessments. Especially in the user-centric model, a relying party such as an online store might decide to trust an identity provider based on soft criteria as the reputation or global image of the company running the identity service rather than on verifiable facts.

In our trust model, we assume that any kind of assessment has been done by the relying party and led to a classification of identity providers into two (trusted, not trusted) or more levels of trust. It is important to note that this decision is specific to a relying party and can be based on strong contracts, the certification of an identity provider by a trusted authority, past experiences just as any other trust criteria that the service provider regards as appropriate.

5.1.1 Formalism

On an abstract level, we can express the quality of any trust relationship as a mapping from a set of Trust Criteria (TC) to a level of trust or level of assurance (LoA):

\[ is\text{Trusted}_{ini} : (TC_1, ..., TC_n) \rightarrow LoA \]

This is exactly what assurance frameworks do. Assurance frameworks define a mapping from certain trust criteria to a level of trust, which in almost all frameworks is one of \{1, 2, 3, 4\}.

In the trust model underlying our implementation, we use a simplified variant of this function with two trust levels \{trusted, untrusted\}. Our trust criteria is the identity provider as a whole (Issuer):

\[ is\text{Trusted} : \text{Issuer} \rightarrow \{\text{trusted, untrusted}\} \]

5.2 The Concept of Identity Trust

Identity trust refers to the trust an entity such as a service provider has into the identity of a subject and its behavior. While the organizational trust level indicates the credibility of the issuer of assertions, the identity trust level indicates the trustworthiness of the subject about which assertions are made. Identity trust is established by credentials that verify properties of the subject. In the claim-based identity management model, these required properties to build up trust (trust requirements) are expressed as claims and exchanged in security tokens. In order to assess the trust into the identity of a subject, such as a user, a relying party needs to assess the received tokens. Hereby, several factors influence the trustworthiness. In order to identify these factors, we use our model of a digital identity.

Figure 3: Model of a digital identity based on [13]

Figure 3 shows our model of a digital identity which we extended from Menzel et al.[13]. This model shows the major relationships between the identity provider, the concept of a digital identity, accounts as well as token and authentication credentials. As can be seen from the picture, a digital identity consists of several Subject Attributes and is hold in an
Account. Each Account can comprise several Digital Identities. Using this model, we can identify the aspects that have an influence on the overall trust into an identity. These are (as marked in green):

**TC-1 Trust into the authentication process and the subject-to-account mapping.**

TC-1 refers to the trust that an identity provider associates a specific subject with the correct record in the identity provider database during an authentication event.

**TC-2 Trust into the subject’s attributes.**

TC-2 refers to the process of identity / attribute proofing and the mechanisms used to verify a specific attribute.

**TC-3 Trust into the token.**

TC-3 refers to the characteristics of the data transfer between the identity provider and a service provider, e.g. the nature of the token and mechanisms used to protect the token from being forged, replayed or altered.

All these factors are subject to vary between different digital identities of the same or different users within an identity provider. In this case, a relying party needs to check on them per transaction. For example, if an identity provider offers various ways of authentication, the relying party requires to know whether the user typed in a password or presented a signed certificate. The same holds for the subject’s attributes. If the process of identity proofing varies between different attributes or different users, a relying party requires to know whether the user presented her/his ID card upon registration or whether the name was self-assigned. Of course, if these factors are static, it is reasonable to consider them as part of the organizational trust relationship as it is usually done in current frameworks.

As we aim in our identity provider to provide digital identities with varying qualities of user attributes, we focus on TC-2 and define a metric on the subject attributes.

### 5.3 Formalisms

We define AttributeTrust to be a function which returns the strength of the attribute proofing process in dependence of the issuer and a certain attribute.

$$\text{AttributeTrust} : (\text{Issuer}, \text{Attribute}) \mapsto \text{AttributeLoA}$$

As with the isTrusted function defined in 5.1.1, it requires a common semantic of the AttributeLoA. Again, it is possible to cluster different trust requirements into levels of assurance. Caution has to be taken as trust requirements usually differ between attribute groups, for example processes to verify a name might be different from processes to verify a membership or the ownership of a specific email-address.

In the trust model underlying our implementation, we use a variant of this function which uses two trust levels with a common semantic {verified, unverified} and leave the specifics for each attribute to be checked separately. We define, isVerified to be a function which returns whether an identity attribute/claim was verified by the identity provider.

$$\text{isVerified} : (\text{Issuer}, \text{Attribute}) \mapsto \{\text{verified}, \text{unverified}\}$$

Depending on the needs, we plan to extend this function in future implementations.

To derive the overall credibility, we combine the results of the functions isTrusted and isVerified. The way, in which both results are combined shall be defined by a function $h$, which can be application-specific or globally defined. The function $h$ describes, in which way the fact whether an identity attribute has been verified is combined with the fact whether this has been done by a trusted identity provider. To follow our observation, we would define the credibility of a claim to be 1 only if the claim was verified and issued by a trusted issuer. In all other cases, it is 0. A mathematical definition for $h$ is given below.

$$\text{credibility(issuer, claim)} \mapsto h(\text{isTrusted(issuer),}$$

$$\text{isVerified(issuer, claim)})$$

with $h$ e.g. defined as

$$h : \{\text{trusted, untrusted}\} \times \{\text{verified, unverified}\} \mapsto \{1, 0\}$$

$$h : (b_1, b_2) \mapsto \begin{cases} 1, & \text{if } b_1 = \text{trusted} \text{ and } b_2 = \text{verified} \\ 0, & \text{otherwise} \end{cases}$$

Of course, alternative definitions of $h$ are possible to model other trust behavior. In [22], we give for example the following definition of $h$ which distinguishes three different levels of trust.

$$h : \{\text{trusted, untrusted}\} \times \{\text{verified, unverified}\} \mapsto \{2, 1, 0\}$$

$$h : (b_1, b_2) \mapsto \begin{cases} 2, & \text{if } b_1 = \text{trusted} \text{ and } b_2 = \text{verified} \\ 1, & \text{if } b_1 = \text{untrusted} \text{ and } b_2 = \text{verified} \\ 0, & \text{otherwise} \end{cases}$$

Please refer to [22] for further details.

### 5.4 Comparison

Table 1 summarizes the concepts of organizational trust and identity trust and compares them. As Organizational Trust refers to the quality of the trust relationship between organization, it implicitly answers the question: "Can we trust the issuer of a token?". The decision to trust another entity as an identity provider in a SOA or web-based infrastructure, is a decision which is drawn before any messages start flying around. Usually, federation agreements or similar contracts are negotiated and signed when setting up the federation. These decisions are then configured in the infrastructure. As compared to this, identity trust is the trust between the subject of the transaction and the service provider. It is service-call specific and therefore is negotiated each time, a call for a new transaction receives.

### 6. IMPLEMENTING AN IDENTITY PROVIDER FOR VERIFIED DIGITAL IDENTITIES

This section describes our implementation of a trust-aware claim-based identity provider. The section starts with a short description of the technical and functional characteristics of the existing identity provider. After this, Section 6.2 shows a use case which demonstrates the use of identity meta data in our identity provider. The next sections give
insights into our implementation. We describe, how we defined a data structure to express identity meta data as so-called Attribute Context Classes and how we extended the SAML 2.0 assertion specification to send identity meta data as part of security token.

6.1 Existing Identity Provider

This section gives a short overview about our implementation of an identity provider which is in the focus of this paper.

6.1.1 Functional Details

Our prototype is an implementation of an identity provider for service-oriented architectures as well as web applications which features

- a security token service in accordance to the WS-Trust specification 1.3 [16]
- an information card provider based on the specification of SAML 1.1, SAML 2.0 as well as Information Card
- an OpenID Provider according to the OpenID 2.0 Authentication specification [21]

It provides

- security token service functionality including
  - a WS-Meta Data Exchange endpoint to request meta data
  - requesting, issuing and signing of security tokens
  - support for authentication via username token or certificate
- information card provider functionality including
  - issuance of information cards for digital identities
  - creation, editing and deletion of claim types
  - support for various identity selectors
- general identity management system functionality including
  - the creation, editing and deletion of multiple digital identities per user
  - creation, editing and deletion of claims
  - assignment of attributes to digital identities

Table 1: Comparison of Identity Trust and Organizational Trust

<table>
<thead>
<tr>
<th>Organizational Trust</th>
<th>Identity Trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>refers to the quality of the trust relationship between organizations</td>
<td>refers to the identity associated with a transaction</td>
</tr>
<tr>
<td>Can we trust the issuer of a security token?</td>
<td>Can we trust the subject in the token?</td>
</tr>
<tr>
<td>determined out-of-band</td>
<td>determined during service call</td>
</tr>
<tr>
<td>configurable</td>
<td>negotiable</td>
</tr>
</tbody>
</table>

6.1.2 Technical Details

The prototype is developed in Java utilizing a number of open-source libraries. Most important are Sun’s web service stack Metro [4] for handling web services and supporting web service security mechanisms such as the security token service, openid4java to provide support of the OpenID 2.0 Authentication protocol [3] as well as maven [2] to provide configuration and deployment options.

A single web application makes up the prototype, which is deployed and run in Apache Tomcat. The web application offers a web interface as well as a web service-based interface.

6.2 Prototype Use Case

This section describes a small use case which demonstrates the use of identity attributes with different qualities in our identity provider. Figure 4 shows the attribute management page of our identity provider. On this page, a user can manage its identity attributes and assign them to digital identities, which are shown on the right-hand side. As can be seen in Figure 4, a user can have several attributes of the same type such as the E-Mail Address or Given Name. The type of the identity attribute is mapped to the protocol specific type defined by the protocol which is used to request attributes. In case of Information Card, the type is mapped to the global claim types and in case of OpenID the type refers to the attributes defined by the OpenID community that can be used with OpenID Attribute Exchange (cf. e.g. AXSchema.org). For each stored identity attribute the information whether this attribute has been verified during the collection of the data is shown. Moreover, additional information about the verification process is available as can be seen in Figure 5. Figure 5 shows all available identity meta data for a specific attribute type and for all attributes which have been verified. In this example, the user has registered three different email addresses - two of which are verified and one which is unverified. Looking at the verification details, we find additional information for the two which have been verified. One important information in the meta data is the source of the identity attribute. The source is the entity which provided the data. For example, it is possible that the verification process is the same, but has been performed by different identity providers. One use case that shows the relevance of this is the following: If, for example, an identity provider is federated with another partner and the user decides to link its accounts and to share a certain attribute, so that this attribute is available in both identity providers, the source would indicate the original identity provider that verified this attribute. In case this attribute is issued to another party, the information who verified the attribute will be of interests for the relying party to assess the trustworthiness of the information as the organizational trust might differ between the issuing and the verifying identity provider. As such a federation scenario still bears many open questions, it is due to future work. In our example in the current implementation, the source is in one case an authority (the company itself) and in the other case the user, who has provided the data. In addition to the source, the verification method is detailed in the identity meta data. Again in our example, this is for the first email address, the company itself who is acting as an email provider and in the second case, in which the user had entered the data, a verification email had been sent. Upon request, this information is sent as part of the service call.
curity token to a requesting party. Our demo application to show the use of the identity provider in a complete scenario is a classical web site for an online store selling music files which is shown in Figure 6. To complete the purchase, several personal attributes are requested from the user, such as his name, address and payment information. Furthermore, the music stores requires a valid email address to deliver the purchased mp3 files to. Therefore, once the store receives a security token from the identity provider, it will check whether the provided email address fulfills this requirement.

### 6.3 Using Identity Meta Data

This section describes selected implementation aspects with regard to the use of identity meta data in the identity provider.

#### 6.3.1 Identity and Organizational Trust

Given the classification into Organizational and Identity Trust as described in Section 5, this section shows its appliance in the identity meta system upon which our prototype implementation is based. As said before, there are three different types of participants in the identity meta system: the identity providers, the relying parties and the clients/users.

A relying party usually specifies a list of identity provider it trusts to make right assertions. Using the notion of claims, the relying party can express for a list of claims the issuer(s) it will accept tokens from. When receiving a security token, the relying party verifies the issuer of the token by checking whether the signature of the token matches the certificate of one of his trusted identity providers. This is in accordance to our notion of Organizational Trust. Only upon correct verification, the relying party will continue with the information in the token.

The information in the token is required to build up identity trust, that is the trust that the requesting user is in fact entitled to access the system. Therefore, the relying party lists the required identity information as claims in its policy. Upon retrieval of this information from a user’s identity provider, the relying party checks the value of the identity data with its access control policy and makes an entitlement decision. We store for each claim in accordance of the issuer certain attribute meta data information. This is on one hand the information whether a claim value has been verified (the verification status) and on the other side certain verification details. While the verification status is one of verified, unverified or unknown for all claims, the verification details can differ tremendously between different types of claims. Therefore, we keep the data structure at this point very general and easily adaptable and extensible. The next section goes into detail about this.

In order to model organizational trust, at the moment we simply store for each issuer of security tokens, whether we trust this issuer to make right assertions. As a possible refinement in future work, one could also store certain meta information, which is specific to this issuer, such as identity provider meta information. Such meta information could include, but is not limited to the authentication process supported by an identity provider as well as aspects concerning the storage and management of tokens.

### 6.3.2 Attribute Context Classes

We use so called Attribute Context Classes to define meta data for claims. The notion of Attribute Context Classes has been inspired by a former specification in the SAML community, that is the one of the so-called Authentication Context Classes [18]. Authentication Context Classes are a concept which was introduced in SAML 2.0 and which allows to specify meta data for the authentication used between two parties. As the security of an authentication mechanism depends highly on the values, which characterize such an authentication method, SAML Authentication Classes offer the possibility to describe the authentication process in much more detail. While with SAML 1.1 it was only possible to state that an authentication process was performed using a specific authentication method as, for example, a password, Kerberos or a hardware token, SAML 2.0 now allows to specify how the authentication was performed in addition to the fact that it was performed. This way it is possible to state whether a password with a length of two characters was used or a password with six characters, which was well-chosen and has a limit of three false attempts.

For the identity meta data, we adapted the idea and defined our own data model, which contains the following elements:

- **Attribute Context** This data element holds the attribute context, which is comprised of all additional information to the attribute value itself. This element is the upper container for all identity metadata.

- **Attribute Data Source** This data element indicates the source from which the attribute value was originally received and is part of the Attribute Context. This can be for example another identity provider, some authority as a certificate authority or the user himself who entered the data.

- **Verification Context** This data element holds the verification context, which comprises all information related to the verification of an identity attribute value. The Verification Context is one specific context within the Attribute Context.

- **Verification Status** This data element indicates the verification status of an identity attribute value, which should be one of verified, not verified or unknown. The verification status is part of the verification context.

- **Verification Context Declaration** The verification context declaration holds the verification process details. Such a detail could for example be the method.
Figure 4: Identity Provider prototype screenshot showing the management of verified and unverified identity attributes.

Figure 6: The MP3 Store Scenario: Acting as a relying party of the identity provider.
that has been used for verifying the correctness of the attribute. Further extensions are possible and should be added here. The verification context declaration besides the verification status make up the verification context.

### 6.4 SAML Attribute Statement Extentions

In order to exchange identity meta data as part of SAML assertions, we introduce extensions to the SAML 2.0 schema. These extensions allow to specify an attribute context to hold further information about an attribute value. The XML schema in Listing 1 presents our extensions, which are defined in a new namespace: http://de.hpi.ip/saml20/ext.

Listing 1: XML schema definition of identity metadata extensions

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://de.hpi.ip/saml20/ext"
  elementFormDefault="qualified">
  <!--
    Extension to
    saml:AttributeStatements to add
    attribute context information
  -->
  <xs:element
    name="AttributeContext"
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="AttributeDataSource"
          type="xs:string" />
        <xs:element
          ref="VerificationContext" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element
    name="VerificationContext"
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="VerificationStatus"
          type="xs:string" />
        <xs:element
          ref="VerificationContextDecl" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element
    name="VerificationContextDecl"
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="VerificationMethod"
          type="xs:anyType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2 gives an example that uses the introduced schema. According to our use case described in Section 6.2, the assertion states that the email address of the user is staff@company.de and has been verified. The method used for verification is a confirmation email which has been sent to the user.

```xml
<saml:Assertion
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlext="http://de.hpi.ip/saml20/ext">
  [...]</saml:Assertion>
```

### Context contains the data source of the attribute value as well as a verification context, which is meant to contain all information about the verification of the attribute. This includes the verification status besides further information about the verification process comprised in an element named VerificationContext as for example the verification method. The verification method is dependent on the attribute type. Therefore this element can encompass any element structure and is intended to be extended by a suitable data structure to describe an attributes verification. All additional elements are listed in the following with a brief explanation of their meanings:

- **AttributeContext** This element holds the attribute context. This element can be used within the SAML AttributeStatement element.
- **AttributeDataSource** This element holds the Attribute Data Source.
- **VerificationContext** This element holds the verification context.
- **VerificationStatus** This element holds the verification status. This element’s data type is intentionally defined as a general string to allow possibly extensions later on.
- **VerificationContextDecl** The element holds the verification context declaration.

The root element is the AttributeContext, which is added to the complex type Attribute of the SAML 2.0 namespace. It is meant to contain all meta information about the attribute value. The attribute value is defined on the same level in the SAML 2.0 type AttributeValue. The **AttributeValue**
7. CONCLUSION
Past experiences have shown that there would be no single center to the world of information. In order to get from the isolated model, in which each consumer of identity information manages this information himself to an identity management which takes the decentralized nature of the Internet into account, we argue that consumers of identity information need to be able to assess and distinguish the quality of the information they receive. In particular, with regard to the launch of electronic ID cards as fostered by several european governments, different sources of identity information will have a different quality in terms of correctness and integrity. To have this information integrated into current identity management models is the essence of this paper. Therefore, we defined a data structure to express identity meta data as so called Attribute Context Classes and extended the SAML 2.0 assertion specification to send identity meta data as part of security token. As a proof of concept, we presented an identity provider which is able to manage user-defined digital identities besides verified digital identities. Therefore, for each identity attribute, a so called claim, an attribute context is stored to hold information such as the method of verification that has been used to verify the claim.

As part of future work, we plan to extend the definition of required claims in web service policies by a policy reflecting the additional identity meta data required to assess a claim value.

8. REFERENCES