myIdP - The Personal Attribute Hub: Prototype and Quality of Claims

Annett Laube and Severin Hauser
Institute of ICT-based Management
Bern University of Applied Sciences
Biel/Bienne, Switzerland
Email: annett.laube@bfh.ch, severin.hauser@bfh.ch

***Abstract*—The myIdP service is an extension to the Swiss eID infrastructure with the aim to provide a service that handles personal attributes (like address, telephone number, email), which are neither part of the SuisseID identity providers nor of a Claim Assertion Service (CAS), because there is no official authority owning and certifying these data. The myIdP service is a CAS that can reuse data which a user has already given to an application via an Internet transaction. The data is thus validated by the web application before being transferred - as Security Assertion Markup Language (SAML) 2.0 attribute assertion - to the myIdP service. The myIdP service comes in two flavors with different trust relations: the attribute provider and the claim proxy. The attribute provider unites several claims for a given attribute and provides an optional quality assessment before sending it to a requesting web application. A trust relationship must consist between myIdP and the web application. The claim proxy only collects the received claims for a given attribute and transfers them with all details to the requesting application. The application can evaluate the confidence in the data based on the claim details. The model to assess the quality and trustworthiness of the data is based mainly on three factors: freshness of information, quality of the attribute issuer and recurrence of information. The myIdP service is evaluated in a scenario of prefilling e-forms in an eGovernment application.**

Keywords-electronic identity, SuisseID, attribute authority, e-form, quality assessment.

I. INTRODUCTION

In a previous paper [1], we presented myIdP, a service based on the SuisseID Infrastructure, an infrastructure for electronic proof of identities (eID) in Switzerland introduced in 2012. In this extended version of the paper we describe the prototype in more detail and discuss the quality model for claims.

The basis of the SuisseID Infrastructure is the SuisseID [2], which is available as USB stick or chip card and contains two digital certificates: (1) the SuisseID identification and authentication certificate (IAC) and (2) the SuisseID qualified digital certificate (QC). The SuisseID IAC can be used to identify the owner in Internet transactions. The SuisseID QC can be used to sign electronic documents in a forgery-proof manner and is not used in the context of myIdP. The SuisseIDs are issued by identity providers (IdP). Contrary to other European countries, where electronic identities are issued by the government together with offline identification (ID card), there are one governmental, but two commercial SuisseID IdPs [3][4] at present.

The SuisseID and its certificates contain only a minimum of personal data (SuisseID number, name or pseudonym and optional email address), due to stringent privacy and data protection requirements in Switzerland. Additionally, a subset of the personal data from the identification document (e.g., a passport) and a well-defined set of additional attributes gathered during the registration process (so called registration process data, RPD) are stored in the identity provider service (extended IdP). The only way to retrieve this data is by strong authentication with the IdP service, using the appropriate SuisseID IAC. The SuisseID Infrastructure is completed with a set of Claim Assertion Services (CAS) [5]. The purpose of a CAS is to provide and certify specific properties or attributes, which had been assigned to the SuisseID owner by some private or public authority. Examples are the membership of an organization or a company, and the proof of professional qualifications, like a notary or a doctor. Especially in the context of eGovernment there is a need for an extension of the beforehand described SuisseID Infrastructure.

More personal attributes (like invoice address, telephone number, email) used in web applications, e.g., online shops, or in electronic forms often used in the eGovernment, are neither subject of the SuisseID IdPs nor the CAS, because there is no official authority owning and certifying these data. The myIdP service fills that gap and allows a SuisseID owner to store and maintain personal attributes. The idea is to store information, which was at least entered (and thus used) once in a web application, for later reuse. The data is used and thus validated by the web application before being transferred as Security Assertion Markup Language (SAML) attribute statement [6] (the so-called attribute claim) to the myIdP service. After that, the user can reuse the attribute for other applications, which improves usability and reduces the error potential in the daily internet transactions.

This paper starts with the related work in Section II, then outlines the architecture, components and flavors of myIdP in Section III. Privacy and data security are subjects of Section IV. In Section V, the integration of myIdP in a scenario of prefilling e-forms is shown. The myIdP quality assessment and trustworthiness is discussed in Section VI. Section VII concludes the document and gives an outlook on further

improvements of myIdP.

II. Related Work

A service like myIdP or a SuisseID CAS technically corresponds to an Attribute Authority defined by SAML [6]: An Attribute Authority is a system entity that produces attribute assertions [7].

In general, most of the known SAML Identity Providers (IdPs) can act as an Attribute Authority and issue attribute assertions beside their usual authentication functionality. Examples are the government-issued electronic identities of the European Countries, like the German Identity Card [8], the beID from Belgium [9] or the Citizens Card from Austria [10]. Similar to the SuisseID, all these government-issued eIDs provide only a small number of personal attributes related to the identity document they belong to.

The national electronic identities of the European member states are made interoperable with the STORK European eID Interoperability Platform [11]. With six pilots, the STORK project offers several cross-border eGovernment identity services. In the follow-up project STORK 2.0 [12] also personal attributes related to eIDs are subject of investigation. E.g., in the banking pilot, public and private identity and attribute providers are included in the process of "Opening a bank account" in a foreign country, online, with a national eID, without physical presence. myIdP could be used in this context as attribute provider for personal attributes, like address, telephone number, email, etc.

In contrast to the central, government-regulated eID services, OpenID [13] is a decentralized authentification service for web based services. The user is free to choose his favourite OpenID identity provider to get an OpenID, which is an URL or XRI including an end-user chosen name (e.g., alice.openid.example.org). OpenID providers are, e.g., Clavid [14], CloudID [15], Google [16] etc. The OpenID providers themselves can support different authentification methods. For example, Clavid offers username/password, one time passwords, SuisseID authentication and the biometric AXSionics Internet Passport.

User attributes, like name, gender or favorite movies, can also be transferred from the OpenID identity providers to the relying party following the OpenID Attribute Exchange Specification [17]. The attributes can be defined (almost) freely, according to the requirements of the relying party. As many OpenID providers do not validate the information entered by the users, the provided attributes have a low level of assurance. There is a need for a validation by a trusted 3rd party, a so called attribute provider (AP). Google started the Open Attribute Exchange Network (Open AXN, also known as "street identity", see [18]) to include validated information from APs. myIdP has the potential to act as an OpenID attribute provider, but is currently only enabled to be used together with the SuisseID.

WebIDs [19] are especially common in social media (Facebook, LinkedIn, etc.) to allow users to identify themselves in order to publish information. Each user can make their own WebID or rely on an identity provider. The WebID is a URL with a #tag pointing to a FOAF file [20] that contains a cross-link to a (self-generated) certificate. Information that should be included but is not required in a WebID Profile are the name (foaf:name) of the individual or agent, the email address associated (foaf:mbox) and the agent's image (foaf:depiction). More attributes and links to all kind of web objects (other persons, groups, publications, account, etc.) can be included as well. WebIDs can be connected to OpenIDs and vice versa.

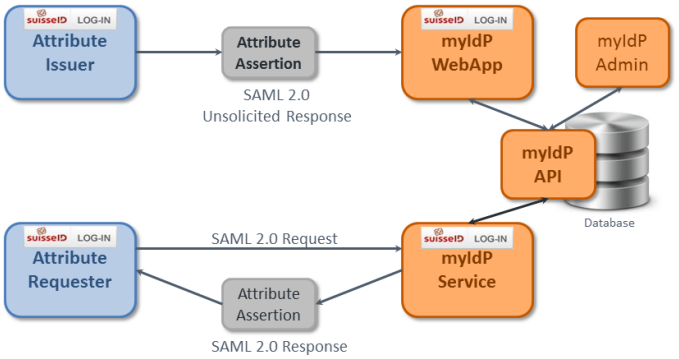

Figure 1. myIdP components and service provider roles

III. Architecture

myIdP consists of four components (see orange boxes in Figure 1): the myIdP Service, the myIdP WebApp, the myIdP Admin and the myIdP API.

A. myIdP Service

The **myIdP Service** is an attribute authority according to the SAML 2.0 standard [6] distributing assertions in response to identity attribute queries from an entity acting as an attribute requester. Like a typical SuisseID CAS [5], the users can – after a successful authentication with their SuisseID IAC – select and confirm attributes, which were formerly received from an attribute issuer, e.g., a web shop, and are stored in the myIdP database. The available attributes are not fixed. They depend on the application scenario and can be configured with the help of the myIdP Admin tool.

New to the concept of CAS is the provisioning of a quality (level of assurance, level of confidence) together within the attribute assertions. myIdP integrates a quality module that calculates the trustworthiness of the provided information on the basis of the age, number of affirmations and quality of the issuer of the received and stored attribute assertions. This assurance level or quality can be used by an attribute requester to insist on a certain level of assurance for the requested attributes. The different approaches to calculate the assurance level or quality are discussed in Section VI.

The myIdP Service is available in two flavors: the **Attribute Provider** and the **Claim Proxy**.

The Attribute Provider summarizes the attribute assertions available in the myIdP database for the given request. All details about the original attribute providers of the information are hidden. After the user has selected and confirmed the

attribute values, the newly built attribute assertion is signed by the myIdP Service. When requested, an assurance level is included in this assertion. For this myIdP flavor, a direct trust relationship is established between the myIdP Service and the web application in the attribute requester role.

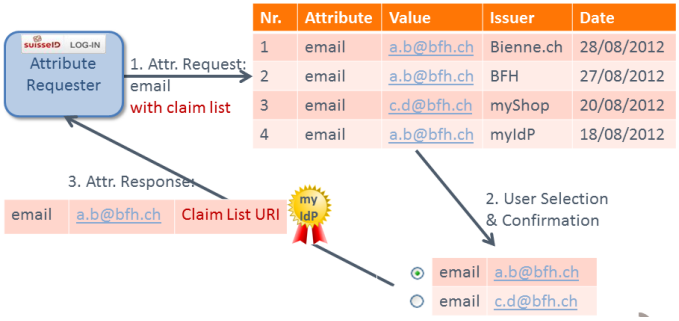

Figure 2. Example Processing Attribute Request

In Figure 2, an example of the process of constructing a response for an attribute request is shown. The attribute *email* is requested. In the myIdP database, four records with two different values are found. The two values are displayed to the user, who selects and confirms which one to use. The SAML attribute response then contains this value with myIdP as attribute issuer.

This is different in the second myIdP flavor. The Claim Proxy extracts the stored attribute assertions from the myIdP database for a given attribute request. After the selection of attribute values and the explicit confirmation by the user, an attribute assertion containing an URI and optionally the assurance level is returned to the requesting web application (the differences between the Attribute Provider and Proxy Mode are highlighted in red in Figure 2). This attribute request is also signed by myIdP but only to ensure integrity. The web application can use the URI from the attribute assertion to assess the originally received attribute assertions enveloped in an XML document. After downloading the XML document, the web application can access all details of the original assertions, including the issuers and timestamps, and perform its own quality assessment. The trust relationship has changed: the web application trusts the attribute issuers directly.

In order to support the provision of a quality assessment of an attribute value and of the claim list URI, the SAML attribute assertions was extended (see the XSD fragment shown in Figure 3).

The SAML attribute request contains an additional flag (attribute *ClaimList*), that indicates the use of claim proxy mode. The SAML attribute response contains as result an URI pointing to a list of claims (attribute assertions) extracted from the myIdP database (attribute *ClaimListURI*). See the shortened example of a SAML attribute response in Figure 4. The URI is only for a short time available to the service provider and enables the service provider to download all claim details.

An example for the provided claimlists is shown conceptionally in Figure 5. Corresponding to the example used in Figure

```
<complexType name="AttributeType">
  <sequence>
    <element ref="saml:AttributeValue"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Name" type="string"
    use="required" />
  <attribute name="NameFormat" type="anyURI"
    use="optional" />
  <attribute name="FriendlyName" type="string"
    use="optional" />
  <attribute name="myidp:Quality" type="decimal"
    use="optional" />
  <attribute name="myidp:ClaimListURI" type="anyURI"
    use="optional" />
  <attribute name="myidp:ClaimList" type="boolean"
    use="optional" />
  <anyAttribute namespace="##other"
    processContents="lax" />
</complexType>
```

Figure 3. Extended xsd `AttributeType`

```
<saml2p:Response>
 <saml2:Issuer>https://myidp.bfh.ch:8443
 </saml2:Issuer>
 <ds:Signature> ... </ds:Signature>
 <saml2p:Status>
  <saml2p:StatusCode Value=
    "urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </saml2p:Status>
 <saml2:Assertion>
  <saml2:Issuer>https://myidp.bfh.ch:8443
        </saml2:Issuer>
  <ds:Signature> ... </ds:Signature>
  <saml2:Subject>
   <saml2:NameID>1300-0000-0001-0001</saml2:NameID>
     ...
  </saml2:Subject>
  <saml2:Conditions
     NotBefore="2014-01-17T13:12:23.922Z"
     NotOnOrAfter="2014-01-17T13:22:25.922Z"/>
  <saml2:AttributeStatement>
   <saml2:Attribute
     Name="http://www.ech.ch/xmlns/
                 eCH-0046/2/emailAddress"
     NameFormat="urn:oasis:names:tc:SAML:2.0:
                      attrname-format:uri"
     myidp:issuerlisturi="https://myidp.bfh.ch:8443/
           myidp-service/requestXml?param=397332808"
     xmlns:myidp=
           "http://www.myidp.ch/xmlns/schema/v1">
    <saml2:AttributeValue>a.b@bfh.ch
    </saml2:AttributeValue>
   </saml2:Attribute>
  </saml2:AttributeStatement>
 </saml2:Assertion>
</saml2p:Response>
```

Figure 4. SAML Attribute response

2, the claim list contains three entries. The service provider can use the information to do its own quality assessment.

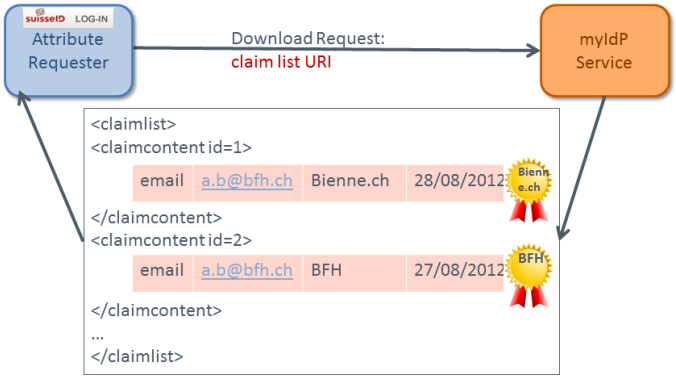

Figure 5. Example Claim Proxy - Claim List

B. myIdP WebApp

The **myIdP WebApp** is the end user front-end of myIdP, where users – after the successful authentication with their SuisseID IAC – can view and manage their attributes. Attributes cannot be entered directly into the myIdP WebApp, except the master data related to the myIdP account (billing address and contact email). Attributes always come from a service provider, e.g., a web application, acting as attribute issuer, which forwards – after confirmation by the user – the attribute assertions to myIdP. The attributes then arrive in the so-called **Inbox** (see Figure 7), where they can be viewed in detail and manually activated before they are exposed via the myIdP service. Corresponding to the user-centric approach of the SuisseID, the users are always in control of their data and can activate/deactivate and delete attributes at any time. As a side effect, the user gets a usage history of their attributes in the web.

C. myIdP Admin

The **myIdP Admin** is an administration tool for myIdP. It supports the maintenance of attribute definitions and the registration process of service providers, which is needed to set up secure connections and trust relationships. A service provider can register as attribute issuer or attribute requester (see Section III-E). The registration details contain the used URLs, certificates and optional SAML metadata.

The registration also allows myIdP to ensure that the service providers well behave. Otherwise one might for instance issue non-consented claims.

New attributes can be enabled for usage within the myIdP community simply by importing the related XML Schemata or by using the SAML Metadata Exchange [21].

In order to assist the research activities to develop an appropriate quality model, the myIdP Admin supports the substitution of the used quality model. Due to the use of the Strategy design pattern, it is possible to develop and integrate new quality calculation models at any time.

D. myIdP API

The **myIdP API** provides an interface to the central database commonly used by the other three myIdP components.

E. Service Providers

A service provider can interact with myIdP, incorporating two roles (see the blue boxes in Figure 1):

- **Attribute Requester:** The service provider electronically sends an attribute query to the myIdP Service in order to draw a confirmation statement - a SAML 2.0 attribute assertion - from the myIdP service and uses it in further actions, e.g., prefilling of web forms.
- **Attribute Issuer:** The service provider sends SAML 2.0 attribute statements (unsolicited SAML response) to myIdP. (Despite the possibility to group several attributes in one SAML statement, myIdP prefers single attribute statements, in order to expose a minimum of information in the claim proxy case.) The attribute values were entered either manually into the web application by the users or have been requested beforehand from the myIdP Service.
  A special attribute issuer is the myIdP WebApp, which uses the master data (address, email) entered during the myIdP registration process, to provide the first attribute statements to the users.

Figure 6. Screenshot myIdP Client - Attribute Issuer

To demonstrate and test the behaviour of a service provider, a demo web application, the **myIdP Client**, was developed. The myIdP Client offers two functionalities, corresponding to the two service provider roles: sending attributes to the myIdP Service and requesting attributes.

In Figure 6, the screen for acting as an attribute issuer is shown. The user can, like in a normal web application, enter some data, e.g., an address or the email. When the user hits the button ”next” and has checked the box beside, to confirm the disclosure of his data to myIdP, the myIdP client sends

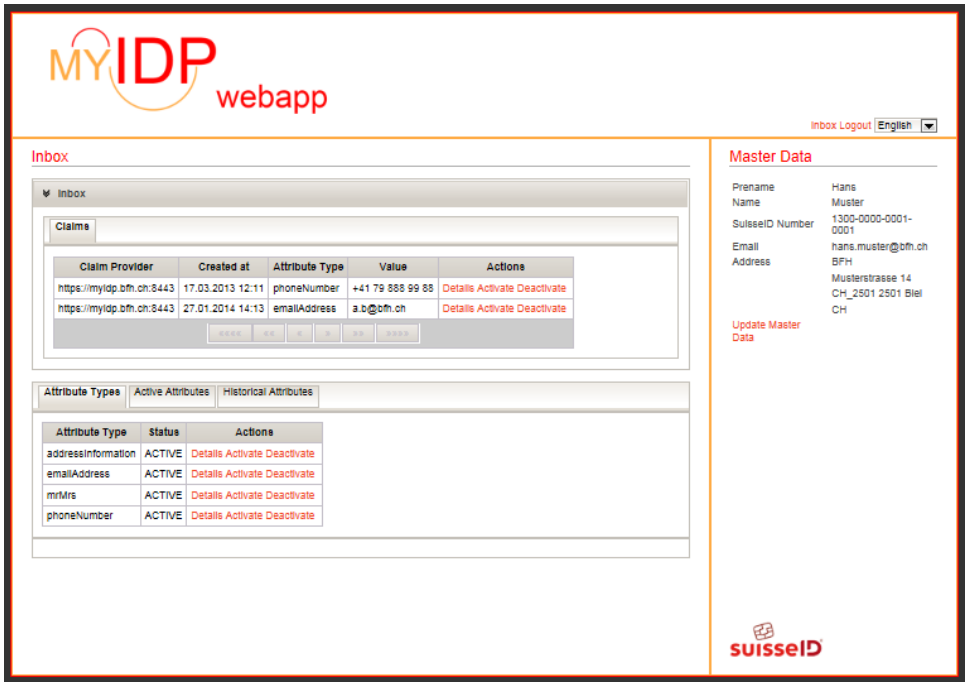

Figure 7. Screenshot myIdP WebApp - Inbox

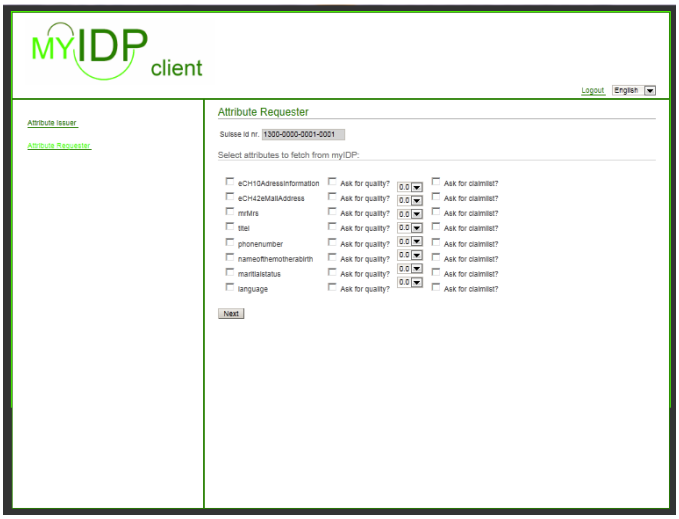

Figure 8. Screenshot myIdP Client - Attribute Requester

an unsolicited SAML response message to the myIdP Service. The myIdP service validates the message and if valid, stores it in the myIdP database. The user can now enter the myIdP WebApp to activate this attribute for further use.

Figure 8 shows the second role a service provider can incorporate. Before prompting the user for input, the service provider sends a request of a set of SAML attribute to the myIdP service. The user can now select and confirm the different values, instead of reentering them in the web application. In the myIdP client the normally hidden step is visualized. The user can select which attribute to request from myIdP. Additionally, a quality assurance value or the claim list activating the Claim Proxy mode of myIdP (see myIdP flavors in Section III-A) can be requested.

IV. Privacy

One important characteristic of myIdP (valid for both flavors) is the user-centric approach. The user is always aware which information is exchanged and has to confirm explicitly every single attribute that is sent out by myIdP.

myIdP implements multiple measures to ensure the privacy of the user:

First, every attribute issuer has to get a user consent before sending any attribute statement to myIdP. In the myIdP client, this is realized by actively requesting the user to check the check-box, that allows the application to send the information to myIdP (see Figure 6). Another option is to include the user consent prominent in the Terms of Usage of the application.

That the attribute issuers full fill these requirements is ensured by myIdP with a registration process for attribute issuers and a corresponding white list.

Secondly, in myIdP the incoming attributes are deactivated by default. The user has to confirm explicitly whether the attributes should be activated (for further use). At any time the user is free to delete attribute statements in the myIdP WebApp or to deactivate them.

Thirdly, the user is involved in every message exchange with an Attribute Requester and has to confirm all attribute values. In the claim proxy case, the disclosure of original attribute assertions needs to be approved as well. The attribute

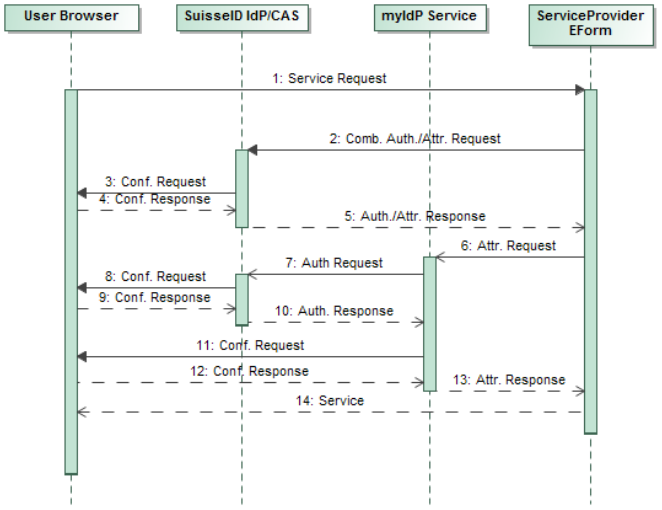

Figure 9. Sequence diagram ”Get e-form”

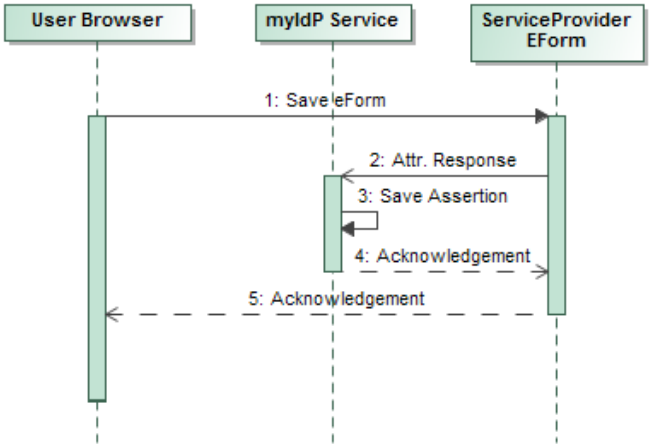

Figure 10. Sequence diagram ”Save e-form”

assertions contain information about visited web sites and could be used to track the user and to create user profiles.

In any case, myIdP only sends attribute statements to Attribute Requesters if a valid authentication with a SuisseID in addition to the user consent exists. These procedures ensure that the user exposes only the intended data and the privacy is protected.

V. Application Scenario

A scenario of completing electronic forms (e-forms) validates our approach. E-forms are commonly used in the Swiss eGovernment. With the help of the SuisseID, the citizen can be identified securely and the attributes stored in the core SuisseID components, like name, birthday, place of birth or nationality, can be used to prefill the e-forms. As the number of attributes available in the core SuisseID is quite limited, we propose the usage of myIdP to provide additional values for the e-forms.

For our proof of concept, we chose the form ”Proof of residence”, which had an integration with the core SuisseID infrastructure already. In Figure 11, an extract of the French version of this form is displayed. The data filled from the core SuisseID infrastructure and myIdP are marked differently (pink - core SuisseID, yellow - myIdP). The data from the SuisseID cannot be overwritten by the user, as they represent certified attributes validated by a trusted authority. In contrast, the myIdP data can be updated. When all data is up-to-date, the user only has to enter one number, which is the number of copies wished-for (the field is marked with a red box), before sending the e-form to the administration.

In Figure 9, the interactions between the user, the e-form provider, the core SuisseID components and myIdP are depicted (only the main scenario is depicted, exceptions and error cases are omitted for the sake of readability):

1) Service Request: the user requests an e-form from the e-form provider (e.g., by clicking on a link).
2) Authentication with SuisseID: the e-form provider issues a combined authentication and attribute request to the SuisseID IdP/CAS. The following attributes are requested: name, first name and birthday.
3) Confirmation request: the user has to identify themself, by entering his secret key (PIN) and in a second step to confirm the SuisseID attributes.
4) Confirmation response: the user’s decisions are sent back to the SuisseID IdP/CAS.
5) Authentication and Attribute response: the SuisseID IdP/CAS sends a combined authentication and attribute assertion back to the e-form provider.
6) myIdP attribute request: the e-form provider issues an attribute request to the myIdP service asking for the address and the email.
7) Authentication Request: the myIdP data are only accessible to the identified owners. That means, the myIdP Service forces a (second) SuisseID authentication.
8) Confirmation request: the user has to identify themself, by entering the secret key (PIN) and in a second step to confirm the disclosure of his identity.
9) Confirmation response: the user’s decisions are sent back to the SuisseID IdP.
10) Authentication response: the SuisseID IdP sends an authentication assertion back to the myIdP Service.
11) Confirmation request: the user has to select the attribute values, in case several emails or addresses are stored in myIdP, and to confirm the selection.
12) Confirmation response: the user’s decisions are sent back to myIdP.
13) Attribute response: myIdP sends an attribute assertion back to the e-form provider.
14) Service: the e-form is displayed to the user and contains the selected and confirmed values from the SuisseID IdP/CAS and myIdP.

The user now has to complete the e-form and to enter the number of copies desired. In case, the email or home address has changed, the data can be also manually corrected on the

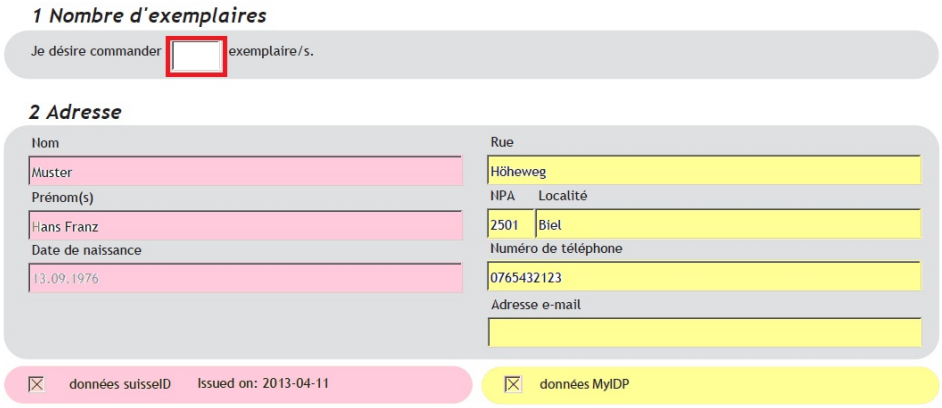

Figure 11. Prefilled e-form "Proof of residence"

e-form (the data from the SuisseID IdP/CAS are read-only and cannot be changed). When the document is saved the governmental process of providing the requested documents is started. However, the confirmed data from the e-form are also transferred – as new attribute assertions (unsolicited message) containing validated information – to myIdP (see Figure 10) where it is stored in the myIdP database.

Looking closely at the interactions between the different actors involved in the application scenario (see Figure 9), it becomes obvious that the user has to authenticate themself twice with the SuisseID IdP: the first time to access the data from the SuisseID CAS and the second time to access the data from the myIdP service. This is quite inconvenient for the user and hardly acceptable in an eGovernment scenario. A possible solution is to enhance the myIdP Service further to support the proxying of authentication requests to a subsequent identity provider, as described in [22]. With this functionality, the user would be requested to authenticate only once; but the two attribute confirmation requests would still be necessary.

A crucial point of using myIdP in eGovernment applications and also in other domains is the selection and standardization of attributes. In our scenario, we could reuse attributes defined and published as Swiss standards, e.g., the eCH-0010 [23] for the address and eCH-0042 [24] for the email. Relying on these standards, the service provider (in our use case, the e-forms provider) can define a stable mapping between the field names in the e-forms or the web application and the attributes supported by the myIdP Service.

VI. Quality Assessment and Trustworthiness

As already mentioned in Section III-A, the myIdP Service is enabled to offer a quality assessment for the provided attributes. This is important, because myIdP does not provide certified information about the SuisseID owner, like a normal CAS. The source of a normal CAS is typically a register belonging to a public or private authority. Examples are the Health Professional Index or the Notary Index of Switzerland, available in [25].

myIdP provides personal information typically without an official authority, which could validate and certify this data. To ensure a good data quality and to increase the trustworthiness of the myIdP data, the user is not allowed to enter the data directly in myIdP. Only a registered attribute issuer can send assertions to myIdP. This ensures, that all information available in myIdP is validated at least once by a service provider acting as attribute issuer.

A service provider acting as attribute requester needs to know how reliable the myIdP data are. In a closed myIdP community, where all service providers incorporate both roles (attribute requester as well as attribute issuer) and have built up trust relationships, the myIdP data would be evenly trustworthy. In a more open environment, where many service providers interact with myIdP, this is different. The service providers need a reliable mean to ensure the trustworthiness of the myIdP information.

The myIdP offers a quality assessment based on an open model. When a service provider acts as an attribute requester, it can ask for the optional quality assessment by myIdP (a value between 0 and 1), provided together with the requested attributes. It can even insist of a certain quality and include a minimum quality the attribute must fulfill in the attribute request (visible in the screenshot in Figure 8). In this case, myIdP will only select attribute values which match or exceed the requested quality.

If a service provider is not willing to rely on the quality assessment of myIdP, he can choose the myIdP Proxy mode (see Section III-A). In this myIdP flavor, the attribute requester gets all stored attribute assertions from the myIdP database belonging to the return attribute value. They can now perform their own quality assessment.

The quality assessment is a statement about the potential correctness of an attribute value. We identified three factors the quality assessment in myIdP should be based on:

1) Freshness of information
2) Quality of attribute issuer
3) Recurrence of information

A. Freshness of information

The freshness f can be calculated from the age a of an attribute assertion. This is the time between when the attribute

assertion was issued and the time when a service provider requests this information. The fresher the attribute assertion, the better the quality. The quality of an attribute decreases gradually. In [26], the Formula (1) was elaborated and tested. It calculates the freshness on the basis of a normalized age value (like the quality a value between 0 and 1). The normalization has to be determined in dependency of the attribute. The average validity of attributes can be quite different; some, like eye color or gender, do almost not change during lifetime, others, like address, do in certain periods. To determine this average validity, demographic information could be used.

$$f = 0.5 + \sqrt{1 - 2a^2} * 0.5 \quad \text{if } (0 \leq a \leq 0.5)$$
$$f = 0.5 - \sqrt{1 - 4 * (a-1)^2} * 0.5 \quad \text{if } (0.5 < a \leq 1) \tag{1}$$

B. Quality of attribute issuer

We propose to classify the attribute issuers according the the STORK Attribute Quality Authentication Assurance (AQAA) scheme [27], which is an extension of the STORK Quality Authentication Assurance (QAA) scheme published in [28]. The STORK QAA model permits quality levels to be assigned to various eID solutions, based on some of their main characteristics. The STORK AQAA model extends this model to be applied to attribute providers providing no directly related information to an eID solution.

The myIdP attribute issuers can be considered as attribute providers and therefore be classified into the four STORK QAA Levels (see Table I).

TABLE I. STORK QAA LEVELS [28]

STORK QAA level	Description
1	No or minimal assurance
2	Low assurance
3	Substantial assurance
4	High assurance

The AQAA level of the attribute issuers influences the quality of an attribute, like shown in Figure 12. In combination with the freshness of attribute (see Formula (1)), a low level of AQAA results in a displacement of the curve and therefor in a general decrease of quality.

The quality q of a single assertion can be calculated using the Formula (2), whereby coefficient k_{AQAA} indicates the decrease of quality assigned to the reached AQAA level of the attribute issuer.

$$q = \max\{f - k_{AQAA}, 0\} \tag{2}$$

C. Recurrence

An increasing number n of assertions containing the same value for an attribute should increase the quality of this value. We propose a formula (3) that shows a logarithmic behavior to calculate the recurrence r_{set} of a set of assertions. The rise coefficient k_{rise} is responsible to determine how steep the increase of quality should be at the beginning; it is normally

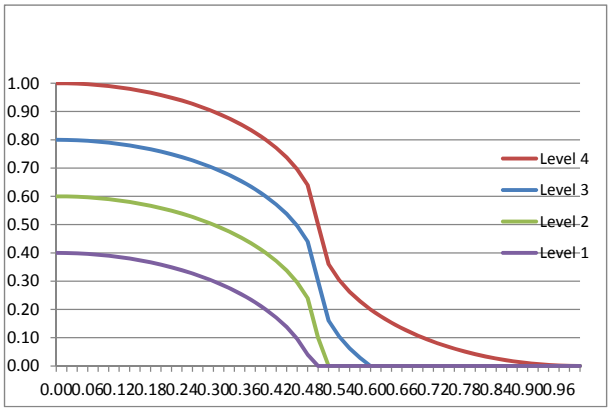

Figure 12. Freshness combined with AQAA level of attribute issuers

a value between 1 and 10. Like the influence of the age of the freshness, the rise coefficient can be different for each attribute.

$$r_{set} = \min\{(\log(n) + k_{rise})/(k_{rise} + 1) * 0.5, 1\} \quad \text{if } (0 \leq n) \tag{3}$$

D. Assertion sets

The quality of an assertion set can be calculated using different formulas. The simplest case would be to take only into account the recurrence and use Formula (3).

In Formula (4), the best freshness value is combined with size of the set (recurrence). The AQAA level of attribute issuers is not taken into account.

$$q_{set} = \min\{\max_i\{f_i\} + r_{set}, 1\} \tag{4}$$

Formula (5) is quite similar to Formula (4), but it uses the freshness in combination with the AQAA level.

$$q_{set} = \min\{\max_i\{q_i\} + r_{set}, 1\} \tag{5}$$

Both formulas tend to overrate the recurrence of the information and are prone to fraud scenarios, where many assertions from a low ranked attribute provider can significantly increase the quality.

$$q_{set} = \min\{\max_i\{q_i\} + r_{set}, \alpha_{MaxAQAA+1}\} \tag{6}$$

The third formula (6) reduces this risk of fraud by limiting the quality of the set of assertions to the maximum value allowed by the next highest AQAA level ($\alpha_{MaxAQAA+1}$). That means, when the set contains hundreds of assertions from a low-level attribute provider, the quality cannot reach a better value than the highest possible quality of an assertion from an attribute provider on the next higher level.

All three formulas were validated in several scenarios. Still, an assessment of a live-running scenario in a set-up myIdP

community is lacking. On the basis on real-world data, also other approaches to calculate the quality assessment, e.g., based on subjective logic or Bayesian networks would be possible.

VII. Conclusion

myIdP is an extension to the SuisseID infrastructure. It proposes a Claim Assertion Service (SAML attribute authority), which handles personal data used and validated beforehand in other internet transactions. The concept is extensible to other eID solutions and can also be integrated in the STORK European eID Interoperability Platform. In a next step, the possibility to use myIdP as OpenID attribute provider will be investigated. Also the combination with a WebID seems feasible.

The myIdP concept was validated with a prototypical implementation following the proposed architecture. The initial implementation on the basis of the SuisseID SDK [29] quickly showed some limitations. Especially the use of an flexible attribute set or structured attributes, like address, were only partly supported. This was also due to the SuisseID SDK, which was designed for a fixed attribute set. These limitations were addressed in a subsequent bachelor thesis [30]

As proof of concept, the prototype was integrated in an eGovernment scenario of prefilling an e-form in order to obtain a proof of residence. The integration of more e-forms is planned. As precondition the set of myIdP attributes has to be extended to have a standardized basis for the information exchange.

The promoting of the myIdP service showed that many applications are willing to act as Attribute Requester and to use the personal attributes available in myIdP. The functionality to act as Claim Provider and to provide validated information to myIdP and to confirm the reuse is often seen as burden. However, both roles equally have to be provided to create a network of validated personal attributes.

As soon as more service providers will use myIdP in a life scenario and provide regularly attribute claims, the model to calculate the assurance level (quality) can be validated on a real data basis and be improved further.

To strengthen the user-centric approach even more and to protect the private attribute, the central storage of claims in the myIdP database could be changed towards a pseudo-local approach that lets the user choose where to store the data: on a personal device or on a central place. The storage of SAML assertions on the user's device would also enable the usage of myIdP - in addition to the normal online scenario - in environments with limited or no connectivity.

Acknowledgment

Thanks to all students from the Bern University of Applied Sciences who helped with their student projects and bachelor theses [31][32][26] to realize this project.

References

[1] A. Laube and S. Hauser, "myIdP-The Personal Attribute Hub," in Proceedings of the Fifth International Conferences on Advanced Service Computing (SERVICE COMPUTATION 2013), Valencia, Spain, 2013, pp. 1–5.

[2] Arbeitsgruppe Spezifikation des Trägerschaftsverein SuisseID, "eCH0113 SuisseID Specification, Version 1.5," November 30, 2011.

[3] Quo Vadis - SuisseID - Website. [Online]. Available: http://www.quovadisglobal.ch/de/Zertifikate/SuisseID.aspx [retrieved: June, 2014]

[4] Swiss Post - SuisseID - Website. [Online]. Available: http://postsuisseid.ch/en [retrieved: June, 2014]

[5] Arbeitsgruppe SuisseID c/o Staatssekretariat für Wirtschaft SECO, "Claim Assertion Service (CAS), Technical Specification, Version 0.99.07," January 13, 2011.

[6] N. Ragouzis et al., "Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS Committee Draft," March 2008. [Online]. Available: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[7] J. Hodges, R. Philpott, and E. M. (Ed.), "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: https://www.oasis-open.org/committees/download.php/21111/saml-glossary-2.0-os.html

[8] M. Margraf, "The new German ID card," February 2011. [Online]. Available: http://www.personalausweisportal.de/SharedDocs/Downloads/EN/Paper_new_German_ID-card.pdf

[9] The official beID website. [Online]. Available: http://eid.belgium.be/en/ [retrieved: June, 2014]

[10] Austrian Citiziens Card - Official Website. [Online]. Available: http://www.buergerkarte.at/en/index.html [retrieved: June, 2014]

[11] STORK - Project Website. [Online]. Available: https://www.eid-stork.eu [retrieved: June, 2014]

[12] STORK 2.0 - Project Website. [Online]. Available: www.eid-stork2.eu [retrieved: June, 2014]

[13] specs@openid.net, "OpenID Authentication 2.0 - Final," December 2007. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html

[14] Clavid - Official Website. [Online]. Available: clavid.ch [retrieved: June, 2014]

[15] Cloudid.de - OpenIDentity Provider - Website. [Online]. Available: cloudid.de [retrieved: June, 2014]

[16] Google, "Federated Login for Google Account Users," June 2012, accessed June 2014. [Online]. Available: https://developers.google.com/accounts/docs/OpenID

[17] D. Hardt, J. Bufu, and J. Hoyt, "OpenID Attribute Exchange 1.0 - Final," December 2007. [Online]. Available: http://openid.net/specs/openid-attribute-exchange-1_0.html

[18] Open AXN group. Street Identity - Website. [Online]. Available: https://sites.google.com/site/streetidentitylmnop/ [retrieved: June, 2014]

[19] H. Story and S. Corlosquet (eds.), "WebID 1.0. Web Identification and Discovery. W3C Editor's Draft." January 2013. [Online]. Available: http://www.w3.org/2005/Incubator/webid/spec/

[20] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.98," August 2010. [Online]. Available: http://xmlns.com/foaf/spec/

[21] S. Cantor, J. Moreh, R. Philpott, and E. M. (Ed.), "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[22] S. Cantor, J. Kemp, R. Philpott, and E. M. (Ed.), "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[23] Verein eCH, "eCH-0010: Datenstandard Postadresse für natürliche Personen, Firmen, Organisationen und Behörden," October 2011. [Online]. Available: http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0010&documentVersion=5.00

[24] ——, "eCH-0042: Vorgehen zur Identifizierung von eGovernment-relevanten Geschäftsinhalten," June 2005. [Online]. Available: http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0042&documentVersion=1.00

[25] Notary Register of Switzerland - Website. [Online]. Available: https://reg.sdms.ch/eRoleRegister/SID-7dcb2c010099bd3ff8143569fdb840c3/Notary/Index [retrieved: June, 2014]

[26] A. Keller, "Qualitätsmodell im Kontext von myIdP. CASE Arbeit." Master's thesis, Bern University of Applied Sciences (BFH), June 2012. [Online]. Available: http://www.myidp.ch/acms/fileadmin/documents/case_Qualitaetsmodell_v1.0.pdf

[27] H. Graux, "D.3.2 - QAA status report," Apr 2013.

[28] B. Hulsebosch, G. Lenzini, and H. Eertink, "D2.3 - Quality authenticator scheme," Mar 2009.

[29] SuisseID SDK - website. [Online]. Available: https://www.e-service.admin.ch/wiki/display/suisseid/Home [retrieved: June, 2014]

[30] E. Jeannerat, C. Saner, and D. Schaeffer, "Bachelorthesis: SuisseID V2.0 SDKs," Master's thesis, Bern University of Applied Sciences (BFH), June 2013.

[31] R. Imwinkelried and D. Ehrler, "Bachelorthesis: Specification myIdP," Master's thesis, Bern University of Applied Sciences (BFH), January 2012.

[32] R. Bühlmann and M. Jeker, "Bachelorthesis: Specification myIdP Extensions," Master's thesis, Bern University of Applied Sciences (BFH), June 2012.