Automated User Information Conversion to improve Identity Federation Scalability

Daniela Pöhn[1] and Wolfgang Hommel[2]

[1]Leibniz Supercomputing Centre, Bavarian Academy of Sciences and Humanities, Boltzmannstraße 1, D-85748 Garching n. Munich, Germany, daniela.poehn@lrz.de

[2]Universität der Bundeswehr München, Computer Science Department, D-85577 Neubiberg, Germany, wolfgang.hommel@unibw.de

Keywords
Federated Identity Management, Inter-Federation, SAML, User information, Attribute conversion.

1. ABSTRACT

Many European higher education institutions (HEIs) have joined an identity federation, which is typically run by their national research and education network (NREN). For example, the German NREN DFN operates the so-called authentication and authorization infrastructure (AAI) DFN-AAI. By being part of a federation, each HEI makes use of federated identity management (FIM). FIM allows users to use the account at their home organization, also called identity provider (IDP), when accessing a service, run by a service provider (SP). By applying FIM, each user account is maintained at a single place, which reduces the required effort for identity & access management and improves the overall quality of user data.

Security Assertion Markup Language (SAML) is the predominant standard for FIM within the higher education. It introduces the closed-trust-boundaries paradigm, which is realized by mostly static national federations and, as research is not limited to geographic boundaries, inter-federations. The biggest inter-federation within the higher education environment is eduGAIN, operated by GÉANT (GÉANT, 2016). In order to be part of a federation or inter-federation, IDPs and SPs need to run a FIM software. Examples of widely deployed FIM software suites are Shibboleth and SimpleSAMLphp.

In order to let the user access a personalized or access-protected service, the SP requires certain user information, called attributes, from the IDP. These attributes can be, e.g., a unique ID, the user's real name, or an email address. The IDP sends such attributes in an assertion statement to the SP, while the SP indicates beforehand the required attributes in its so-called metadata. Metadata is one key element of SAML, which describes an entity (i.e., IDP or SP) with, among other information, the technical communication endpoints (URLs), contact information, and information about the attributes.

Before the attributes are sent to the SP by the IDP, they eventually need to be transformed into the data format understood by the SP. After resolving and filtering all the required attributes, this is done via attribute conversion rules. Federations have created so-called schemas to have a standardized vocabulary of attributes, i.e., they specify attributes with their names, unique identifiers, syntax, and semantics. For example, the German federation DFN-AAI uses the `dfnEduPerson` schema (DFN-AAI, 2009), which extends the international schema `eduPerson` (Internet2, 2013). Additionally, another international schema, `SCHAC` (REFEDS, 2015), exists, while projects and research communities can establish further individual schemas.

If no suitable conversion rule is available at the IDP, the IDP administrator has to implement and manually deploy a new conversion rule to the IDP software configuration. This workflow is typically triggered when a user contacts the IDP administrator complaining that a service does not work the

way it should. As different FIM software use different programming languages and different conversion methods, the situation gets even more complicated.

Therefore, in this article we present an approach based on a generic user attributes conversion rule repository, extending our previous work in (HMP, 2014). We first discuss the motivation for generic conversion rules and elicit the required functionality. We then design an architecture and workflows for this generic conversion rule repository, which is then explained based on a thorough example. We conclude this article with a summary and provide an outlook to our future work.

2. MOTIVATION FOR GENERIC CONVERSION RULES

Setting up an IDP software the first time requires a connector to the local identity and access management (I&AM) system, which is typically implemented using a leightweight directory access protocol (LDAP) server or relational database management system. The I&AM system authenticates the user and provides the user attributes in an IDP-internal format. Due to historical reasons and HEI-specific individual I&AM requirements, the IDP-internally used schema often differs from the schemas used nationally and internationally. Similarly, especially commercial SP services may require user data in proprietary formats that are not compatible with the schemas used in FIM federations. Therefore, attribute conversion rules are needed.

The German schema `dfnEduPerson` includes, for example, the attributes

- `dfnEduPersonCostCenter`,
- `dfnEduPersonStudyBranch1`,
- `dfnEduPersonFieldOfStudyString`, and
- `dfnEduPersonFinalDegree`.

Typical attributes for the schema `eduPerson` are the following:

- `eduPersonAffiliation`,
- `eduPersonScopedAffiliation`,
- `eduPersonEntitlement`, and
- `eduPersonPrincipalName`.

Other common attributes, according to `inetOrgPerson`, are, e.g.:

- `cn (CommonName)`,
- `displayName`,
- `givenName`,
- `sn (Surname)`,
- `o (OrganizationName)`, and
- `mail`.

`SCHAC` comprises further attributes, for example:

- `schacHomeOrganization`,
- `schacMotherTongue`,
- `schacDateOfBirth`, and
- `schacCountryOfCitizenship`.

Other federations may have yet again different schemas. In order to send the raw attributes from the I&AM to the SP, the IDP operator has to transform them by adding conversion rules. Within the FIM software Shibboleth, the conversion is done in a multi-step workflow:

- Fetch the raw user data into the ID
- P software (DataConnector)
- Define the attributes (AttributeDefinition)
- Filter the attributes (AttributeFilter)
- Send the attributes (AttributeRelease)

Other FIM software use similar steps, but with a different configuration format and step denomitations.

Typical conversion rules are:

- Renaming, e.g., from `DateofBirth` to `schacDateOfBirth`,
- Merging and Splitting, e.g., `sn` and `givenName` to `displayName`, and
- Transforming, e.g., different date formats.

For these conversion rules, Shibboleth and other FIM software have pre-defined conversion types, which still need to be applied manually by the IDP administrator.

Therefore, a new attribute at an SP requires the IDP administrator to add the definition, the conversion, and the filtering manually to the IDP configuration. As the user first has to inform the administrator, the administrator needs to gather all the required technical information and then adds the conversion rule; this results in waiting time for the user before she can actually make use of the desired service.

In order to enable automated conversion of attributes, a centralized conversion rule repository needs to be established. This repository stores the information for a conversion from format A to a format B using a generic syntax, which can then be transformed, e.g., to Shibboleth or SimpleSAMLphp conversion rule implementations. In contrast to the conversion rule repository described in (HMP, 2014), the generic conversion rule repository uses a generic format, enabling the re-use of conversion rules by different FIM software suites.

Therefore, an IDP administrator can upload her newly written conversion rule to the repository. If another IDP needs the same conversion rule, but uses another FIM software, it searches the repository based on the source and target attributes; the generic rule is transferred to the IDP and automatically integrated into its FIM software configuration by an IDP extension. This enables the full automation of IDP-SP setup procedures and enables the user to immediately make use of a service.

3. FUNCTIONALITY OF THE GENERIC CONVERSION RULES REPOSITORY

The centralized generic conversion rule repository can be seen as a trusted third party (TTP) for the re-use of conversion rules. The TTP can be operated, for example, by a federation or inter-federation. An extension of the IDP software downloads and integrates the generated conversion rules. This helps IDPs to reduce the workload, while, at the same time, reducing the waiting time for the users, and helping SPs to receive the required attributes.

Instead of the manually written and added conversion rule after a notification from the user, the extension searches for an appropriate conversion rule. The generic conversion rule is adapted for the FIM software and integrated locally. Although this approach is straight-forward, no such service is operated currently.

By making use of a generic conversion rule repository with a relational database for conversion rules, the setup is automated:

- First, the IDP software extension detects that the IDP does not have the necessary attributes for the SP available. The SP requirements are stated in the SAML metadata and required for the automation to work. The IDP attributes can be found within the configuration of the IDP software. By a simple comparison, the missing attributes can be identified.
- By having the required attributes and all the attributes the IDP can offer, the extension queries the TTP. The TTP uses an application programming interface (API) for the communication with the IDPs. The TTP searches a generic conversion rule suited for the IDP.
- If a generic conversion rule is found of a simple conversion, the rule is downloaded and a specific format generated by the IDP software. More complex conversion rules with scripts should be stored IDP software specific and manually downloaded by the IDP administrator.

- If no generic conversion rule is found, the IDP is returned an error. The IDP operator then writes a new conversion rule, which should optionally be uploaded to the TTP. The TTP extracts the generic rule by applying the templates. The generic rule is saved in the database.
- After downloading the conversion rule, the generated specific rule is integrated into the IDP's local configuration. As the conversion rule is now available, the IDP sends the required attributes to the SP and the user can make use of the service.

Therefore, a specific and generic conversion rule can be re-used, speeding up the setup of IDP-SP relationships by reducing the efforts needed for conversion rules. As the TTP is not needed for further communication, it cannot become the bottleneck of a federation or inter-federation and does not introduce any privacy issues. As scripts are not integrated automatically, this also increases the security, while the conversion rule management is lightweighted.

4. ARCHITECTURE AND WORKFLOWS

In order to explain the TTP with the generic conversion rule repository in more detail, the management workflow is shown first, before the generic conversion rule syntax is described.

a. Management Workflow

The basic management workflow for conversion rules, described in (HMP, 2014), can be re-used, as seen in Figure 1. After a new conversion rule is created, it is uploaded to the TTP. At a later date, an IDP operator can delete or update the conversion rule. When a conversion rule is updated or deleted, all IDPs, which re-use the conversion rule, must be notified.

An update of a conversion rule is, e.g., important, if the conversion rule includes a mistake. Even though the TTP validates the conversion rule after the upload, not all cases can be covered.

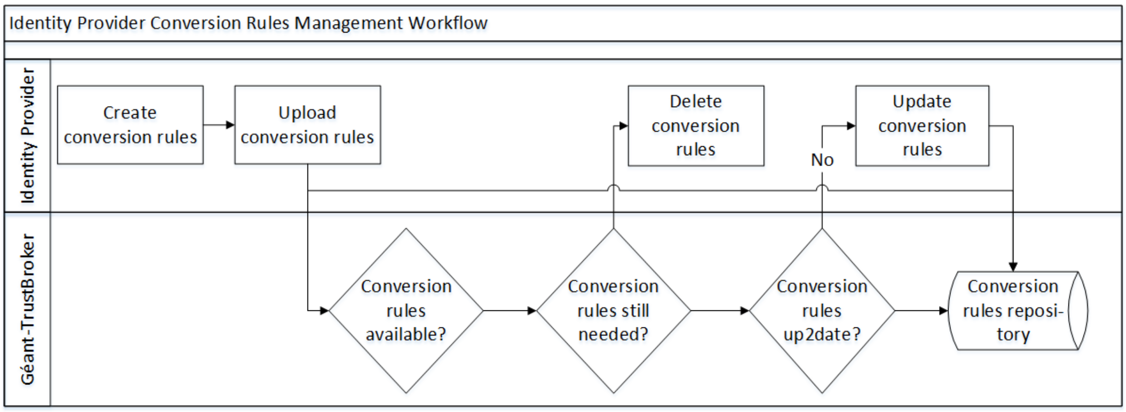
Identity Provider Conversion Rules Management Workflow
Identity Provider
Géant-TrustBroker
Create conversion rules
Upload conversion rules
Delete conversion rules
Update conversion rules
No
Conversion rules available?
Conversion rules still needed?
Conversion rules up2date?
Conversion rules repository

Figure 1: Conversion Rule Management (HMP, 2014)

All generic conversion rules are stored in the database.

b. Architecture

The centralized generic conversion rule repository has a web application, which enables the interactive management of conversion rules. Alternatively, the API can be used for the management of conversion rules. The backend consists of a simple logic of the application, while the necessary data for the conversion rule including permissions is stored in a simple database.

The database has the following tables:

- `ConversionRule`: Conversion from one or more attributes into another attribute.

- ConversionKeyword: Inserts keywords for specific conversion rules.
- ConversionAttribute: Information about source and target attributes for a conversion rule.

This minimalistic design allows the lightweighted storage of generic conversion rules, which is then translated into the FIM software specific format at the IDP. This basic architecture is shown in Figure 2. The TTP stores the generic conversion rules in a database. The generic rule is downloaded, converted and then inserted into the configuration. If a new conversion rule is written, it is translated into a generic format, if no scripts are included, and then uploaded to the TTP.

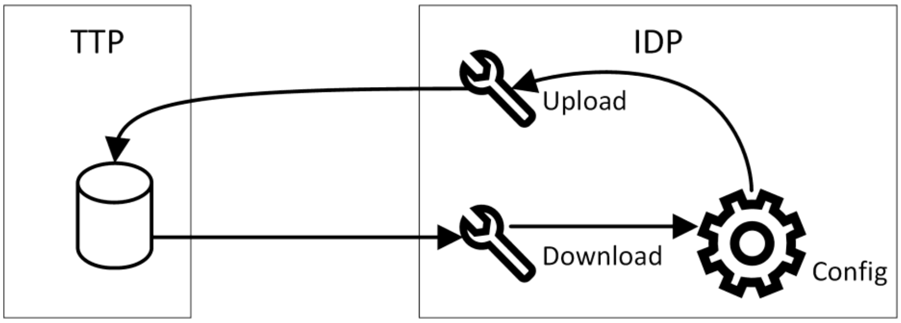

Figure 2: Architecture of Generic Conversion Rule Management

In order to design a generic format of simple conversion rules, the conversion within the FIM software needs to be reviewed in detail. Shibboleth uses different pre-defined operations, for example the following:

- Renaming by mapping of attributes.
- Splitting and other definitions with regular expressions.
- Merging can make use of the template attribute definition, which is written in the Velocity template language.
- Scoping by scoped attribute definition.
- Principal name by principal name attribute definition.
- Furthermore additional scripts can be used.

This shows that mapping can be re-used easily, while splitting and merging needs more information. More complex transformations can be implemented via scripts.

The following pre-defined conversions of SimpleSAMLphp are relevant for the generic conversion rule repository:

- core:AttributeAdd: adds an attribute to the response.
- core:AttributeAlter: searches and alters the value of an attribute.
- core:AttributeMap: renaming of attributes.
- core:PHP: modifies attributes by applying PHP code.
- core:ScopeAttribute: scoping of attributes.

This listing shows that renaming and scoping are relatively easy, while transformation, splitting, and merging are more complex to implement.

The open source FIM software PySAML2 uses a python dictionary to map attributes. Identifiers describe the supported name formats. To and fro statements then contain the actual mapping of the attributes. Therefore, renaming is possible out of the box. For all other conversions, Python language functionalities need to be used.

Active Directory Federation Services (ADFS) is becoming more popular. Although ADFS supports SAML, it uses terms from the WS-Federation world. Claim rules describe which claims (attributes) are sent to the relying party, which equals an SP. As claims are not available in the schemas described above, they need to be transformed by the ADFS 2.0 Custom Rule Language. This is done via an administration tool. An API is supposed to be available in the future. As a result, the generic conversion rule repository can only send the names of the attributes and the type of conversions in this specific case.

In order to fully automate the conversion, the following information is needed:

- sort of conversion,
- source attributes,
- target attribute, and
- additional information, like regex.

By mapping the pre-defined conversion rules, the following keywords are extracted, which are needed to allow the automated generation of FIM software specific conversion rules:

- `source`,
- `target`,
- `targeturn1`,
- `targeturn2` as well as the transformations
- `regex` respectively `pattern` and
- `conversion`.

The generic conversion rule is described as follows.

```
source={source1, source2, ...};
transformation = [renaming, merging, regex, conversion];
target={target, targeturn1, targeturn2};
source(transformation) => target;
```

A simple renaming can be described as the following generic code:

```
source;
transformation = renaming;
target={target, targeturn1, targeturn2};
```

In order to transform the generic conversion rule to a FIM-software-specific conversion rule, templates with the keywords are necessary at the IDP side. Shibboleth uses extensible markup language (XML) code to configure the software and, therefore, transform user attributes. The renaming template for the FIM software Shibboleth is the following:

```
<resolver:AttributeDefinition xsi:type="Simple"
  xmlns="urn:mace:shibboleth:2.0:resolver:ad" id="{{ target }}"
  sourceAttributeID="{{ source }}">
 <resolver:Dependency ref="{{ source|resource }}" />
  <resolver:AttributeEncoder xsi:type="SAML1String"
   xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
   name="{{ targeturn1 }}"/>
  <resolver:AttributeEncoder xsi:type="SAML2String"
   xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
   name="{{ targeturn2 }}"
   friendlyName="{{ target }}" />
</resolver:AttributeDefinition>
```

The FIM software SimpleSAMLphp is written, as the name indicates, in PHP. The renaming template is therefore different:

```
'authproc' => array(
      50 => array(
            'class' => 'core:AttributeMap',
            '{{ target }}' => '{{ source }}',
      ),
)
```

PySAML2 uses a python dictionary for the mapping of attributes, resulting in the following template:

```
MAP = {
    "identifier": "urn:oasis:names:tc:SAML:2.0:attrname-format:basic",
    "fro": {
        '{{ target }}': '{{ source }}',
        },
    "to": {
        '{{ source }}': '{{ target }}',
    }
}
```

These templates can be filled in with the values, when generating specific conversion rules. As long as a template exist, a specific simple conversion rule can be generated.

5. EXAMPLE

In order to illustrate the generic conversion rule repository, an example with the IDP Leibniz Supercomputing Centre (LRZ) is given. The IDP LRZ uses the FIM software Shibboleth, which enables simple conversion rules, like renaming and scoping; Transformation and other conversion rules including scripts need to be downloaded manually. Besides the national federation DFN-AAI, LRZ is also member of the international inter-federation eduGAIN.

Let us assume that the DFN-AAI operates such a conversion rule repository. Therefore, the IDP operator of the LRZ configured the IDP software extension in such a way that simple conversion rules of the DFN-AAI are automatically downloaded and integrated, while other conversion rules need be manually processed. Although this manual step results in waiting time, it is important for the trust in and acceptance of conversion rules. As additional feature, the user is sent an email automatically once the service is usable.

An employee of the LRZ wants to make use of a new service. In this step, the IDP software extension determines the required conversion rules. As the service specifies two attributes, which are not known by the LRZ yet, the IDP software extension queries the TTP.

For one attribute, `mailName`, a generic conversion rule can be found. The conversion rule consists of the target `mailName` and the source attributes `sn` and `givenName`.

```
source={sn, givenName};
transformation = merging;
target={mailName, targeturn1, targeturn2};
```

As the conversion rule includes a script, the conversion rule is manually downloaded after a brief check and then integrated.

```
<resolver:AttributeDefinition id="mailName" xsi:type="Script"
 xmlns="urn:mace:shibboleth:2.0:resolver:ad">
    <resolver:Dependency ref="sn" />
    <resolver:Dependency ref="givenName" />
    <resolver:AttributeEncoder xsi:type="SAML1String"
       xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
       name="{{ targeturn1 }}"/>
    <resolver:AttributeEncoder xsi:type="SAML2String"
       xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
       name="{{ targeturn2 }}" friendlyName="mailName" />
    <Script><![CDATA[
       importPackage(Packages.edu.internet2.middleware.shibboleth.common.
          attribute.provider);
       mailName = new BasicAttribute("mailName");
       merge = sn.getValues().get(0) + " " + givenName.getValues().get(0);
       mailName.getValues().add(merge);
    ]]></Script>
</resolver:AttributeDefinition>
```

For the second attribute, `personalDisplayName`, no conversion rule is found. Therefore, the IDP operator is informed via email about that problem. The administrator manually writes the needed conversion rule and uploades it to the TTP. The user is informed via email and can make use of the service right afterwards.

```
<resolver:AttributeDefinition xsi:type="Simple"
 xmlns="urn:mace:shibboleth:2.0:resolver:ad"
 id="personalDisplayName" sourceAttributeID="displayName">
```

```
  <resolver:Dependency ref="{{ source|resource }}" />
    <resolver:AttributeEncoder xsi:type="SAML1String"
     xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
     name="{{ targeturn1 }}"/>
    <resolver:AttributeEncoder xsi:type="SAML2String"
     xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
     name="{{ targeturn2 }}"
     friendlyName="personalDisplayName" />
</resolver:AttributeDefinition>
```

The specific conversion rule is uploaded to the TTP and stored in the database.

```
source={displayName};
transformation = renaming;
target={personalDisplayName, targeturn1, targeturn2};
```

The federation operator of the DFN-AAI receives an email about this new conversion rule, which can be validated so other IDPs can make use of it in a fully automated manner.

When another user of a different IDP wants to use the same service, the IDP can re-use the newly added conversion rule, reducing the waiting time for the user and the manual workload of the operator.

6. CONCLUSION

The generic conversion rule repository has been described in order to improve the conversion rule repository for Shibboleth (HMP, 2014). Therefore, it extends the proof of concept implementation of the GÉANT-TrustBroker by the on-the-fly attribute conversion between IDPs and SPs with different schemas and FIM software. This generic approach with the use of templates for the FIM software allows the re-use of shared conversion rules independend of the FIM software. Therefore, it speeds up the setup of IDP-SP relationships, helps SPs to receive the needed attributes, and reduces the waiting time for users. As scripts can be used within conversion rules and different FIM software uses different scripting language, a generic way to describe scripts should be found.

In the next step, this generic conversion rule repository is tested with different engaged parties within the GÉANT project. The experiences gained testing the generic conversion rule repository will be used to improve the repository, extend it to support further FIM software suites, and help to support IDPs in eduGAIN.

7. ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

The authors wish to thank the members of the Munich Network Management (MNM) Team for helpful comments on previous versions of this paper. The MNM-Team, directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering, is a group of researchers at Ludwig-Maximilians-Universität München, Technische Universität München, the University of the Federal Armed Forces, and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities.

8. REFERENCES

Hommel, W., Metzger, S., Pöhn, D. (2014). *Géant-TrustBroker: Simplifying Identity & Access Management for International Research Projects and Higher Education Communities*. 20th congress of the European University Information Systems Organisation (EUNIS 2014)

DFN-AAI (2009). *dfnEduPerson Schema 1.0*. Retrieved February 15, 2016, from: https://www.aai.dfn.de/fileadmin/documents/attributes/200811/dfneduperson-1.0.schema.txt

GÉANT (2016). *eduGAIN Homepage*. Retrieved February 15, 2016, from: http://services.geant.net/edugain/Pages/Home.aspx

Internet2 (2013). *eduPerson Object Class Specification (201310)*. Retrieved February 15, 2016, from: http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html

REFEDS (2015). *SCHAC Releases*. Retrieved February 15, 2016, from: https://wiki.refeds.org/display/STAN/SCHAC+Releases

9. AUTHORS' BIOGRAPHIES

Daniela Pöhn received a university diploma degree in Computer Science from the University of Hagen, Germany, in 2012.

She was engaged in the IT industry as a full-time software developer during her studies, before she joined LRZ as a Ph.D. candidate in September 2012.

She is involved in the identity management research activity of the GÉANT project since April, 2013, leading one task about the TrustBroker approach. The focus is mainly on interoperability within federated identity management.

Wolfgang Hommel has a Ph.D. as well as a postdoctoral lecture qualification from Ludwig-Maximilians-Universität in Munich, Germany, where he teaches information security lectures and labs.

His research focuses on information security and IT service management in complex large-scale and inter-organizational scenarios