JRA5Attributes

When a Home Bridging Element releases local attributes to a Remote Bridging Element, some attribute transformation and attribute filtering should take place. Similarly, Remote BE has to filter out unnecessary/unwanted attributes and transform the remaining according to its federation's rules.

Conversion

Regardless of attribute representation of each federation (eg. Attribute Certificates or simple string values), users want to transport the *information* included in attribute values. The aim of attribute conversion is to let BE administrators be able to define rules for including and extracting information into/from attributes that can be handled by both federations.

It is worth noting that inside the two federations they can embed

- the same information into different attribute-value pairs, eg. `fooFedHospitalWard: urn:foo:surgery` and `barFedAbteilung: urn:bar:Chirurgie` could carry the same information ("a medical institution has surgical ward"), while
- the same attribute-value pairs can carry different information, eg. `eduPersonAffiliation: student` could be used for higher educational students only in one federation but in the other this could mean everybody studying in education (from primary schools to PhD).

It is possible that in some cases the information that needs to be transferred is hold by a tuple of attributes. Sometimes the number of attributes representing the same information can differ in each federation. In these cases simple translation of attribute names and values is not enough.

Information needed for conversion

Naturally, federations need to agree on the vocabulary when exchanging these kinds of information. It can be achieved in the following ways:

- the confederation maintains a common vocabulary of allowed attribute values
- peers agree on attribute value interfaces (what Home sends and Remote accepts), independently of the confederation

In the former case, conversion is independent of the relying party, all what BEs need to do is to convert to and from the *common format*. However, maintaining a central vocabulary is not always easy and is quite hard to make it flexible enough. It should be possible to allow federations to define custom interfaces for talking to specific peers.

eduGAIN Credential Conversion Service (*eCCS*)

Gabriel López et al. describe a common 'eduGAIN Credential Conversions Service'.

Summary

They propose a central (although technically distributable) service for carrying out attribute conversion to and from a common representation (eduGain Common Credentials, *eCC*). Conversion polices for converting local attributes to *eCC* and vice versa are included in the metadata. Attributes can be transferred between BEs in *eCC* representation and in 'source format' (in unconverted form). In both cases an RBE needs to invoke *eCCS*, which sends back attributes in the required format of the remote federation.

eCCS operates by utilizing custom SAML extensions called *ConversionQuery* and *WrappedStatement* as well as conversion policy sets (XACML) published in metadata.

Probably the most important thing in eCCS is to **eliminate the NxN matrix problem** (where N is the number of BEs), because of conversion policies published in MDS.

Comments

- eCCS **MUST** be distributed and very close to the BE because of maintaining **privacy**. No central elements (outside the scope of the 2 federations) shall ever receive users' attributes.
- Representing every possible conversion scenarios in conversion policy might be quite difficult
- It would be quite hard if even possible to define custom 'common formats' between subsets of federations using the 'big' eduGAIN MDS. It would be necessary if some federations have different levels of cooperations requiring different attributes.

The Shibboleth Way

Shibboleth uses `*AttributeDefinition` elements to define conversion rules from data source (i.e. LDAP) attributes to "resolved" attributes. `AttributeDefinition`s can depend on `DataConnector`s or other `AttributeDefinition`s.

It can be used for attribute conversion in eduGAIN as well if we can define a `DataConnector` that can use attributes retrieved

- from the IdP (at the HBE), or
- from the HBE (at RBE).

Shibboleth2 has a number of built-in `AttributeDefinition`s:

- SimpleAttributeDefinition: pass through the retrieved value of the attribute

- ScopedAttributeDefinition: append a scope to the attribute value
- TemplateAttributeDefinition: sets value based on an arbitrary template of constant string and other attributes
- MappedAttributeDefinition: sets value according to conditions on (possibly other) attribute values
- ScriptedAttributeDefinition: execute a JSR-223 (Java) script to determine the attribute value. This script gets the context information in `requestContext` variable.
- ... and some others ...

Hooking into EduGAIN

It is still necessary to define an attribute vocabulary within a confederation, however it is technically easier to exchange additional attributes between relying parties. If 'EduGAIN' as a confederation can be one of the relying parties (and why not?), then the NxN matrix problem can be avoided, because only 'special' relying parties (peers) require additional configuration.

- TODO: does EduGAIN have the notion "relying party"?

Open issues

- Licencing issue (Shibboleth has Apache2 style licence)
- Amount of code needs to be included into eduGainBase is unknown. Worst case: whole Java Shibboleth library (`edu.internet2.middleware.shibboleth.common.*`)

Filtering

Both HBE and RBE need to filter the set of attributes released and accepted.

Shibboleth2 comes with a filtering code that is the same both for the attribute publisher and the consumer. Filtering rules may be based on (among others):

- requester/issuer
- attribute values
- authentication context

Változat #1
document-uploader hozta létre 2025-08-07 12:11:23 CEST
document-uploader frissítette 2025-08-07 12:11:23 CEST