

An Identity Provider's Guide to the Core Attributes

Ian A. Young
SDSS, EDINA, University of Edinburgh

McShib meeting, 14th December 2007

Problem Statement

- Federated Access Management is all about the attributes released by the IdP to the SP
- IdP and SP need to agree on:
 - attributes to exchange
 - their definitions
 - their quality
- (more) easily resolved if IdP = SP
 - e.g., internal institutional applications
- If they're not the same party, this is *hard*

Festive Caricatures (1)

- Service provider: I want a pony!
 - I'd like attributes A, B, C, and D through Z please
 - if you give me more, I can do more
 - if you give me attribute Y, my code will be easier to write
 - I've already written code that needs attribute X
 - Summary: as much as possible, please

Festive Caricatures (2)

- Identity Provider: No, you can't have a pony!
 - we don't even have all of that information, we'd have to collect it
 - then we'd have to maintain it to make sure it was correct
 - we can't release attribute X to you without talking to our lawyers
 - We don't see why you have a real need for Y.
 - Summary: as little as possible, please

Where to Begin?

- Some SPs tell us what they want:
 - <http://tinyurl.com/2y92cj>
 - this tends to encourage standardisation
- Some SPs prefer to negotiate with IdPs
- If you have more information, let us know!
- Remember: information release is your responsibility so it's your call

Finding a Balance

- Core attributes:
 - minimal set of four very flexible attributes
 - chosen from eduPerson for interoperability
 - good enough for *most* situations
 - of course, not sufficient for *all* situations
- SPs told: you may have problems if you ask for something outside this set
- IdPs told: you may not be able to access some popular services if you can't provide this set

Stored vs. Transmitted

- The attributes you transmit don't have to be the same attributes you have stored.
- Attributes can be gathered from multiple sources.
- Attributes can be transformed, e.g., by scripts you write.
- So, no requirement to alter your directory schema.
- Release only after positive policy decision.

eduPersonScopedAffiliation (ePSA)

- Possibly the most important attribute in the UKf
- Describes the subject's *relationship* with their institution
- *What are they to you?*
- Example: member@ed.ac.uk
- Only a few permissible values (this is *good*)
- ... but even fewer see real use

ePSA Values (1)

- student, staff, faculty, employee, member, affiliate, alum, library-walk-in
- multi-valued attribute for each subject
- value space has structure:
 - e.g., student@ implies member@ as well
- only release what the service provider needs!
- normally safe to release member@ to everyone

ePSA Values (2)

- Most important value: member
 - “member in good standing of the ... community”
 - corresponds to most “authorised users” in the JISC model license
 - safe to release, adequate for many SPs
- Er, that’s it...
- Upcoming: library-walk-in
 - recently profiled by MACE-Dir for new eduPerson
 - corresponds to the other “authorised users”

Scripting eduPersonScopedAffiliation

- Your directory says “role is student” in code
- ... but you want ePSA = “student”
- ePSA can be derived from “unscoped” ePA:

```
<ScriptletAttributeDefinition
  id="urn:mace:dir:attribute-def:eduPersonAffiliation">
  <DataConnectorDependency requires="directory"/>
  <Scriptlet><![CDATA[
    Attributes attributes =
      dependencies.getConnectorResolution("directory");
    Attribute roles = attributes.get("roles");
    if (roles.contains("00142")) {
      resolverAttribute.addValue("student");
    }
  ]]></Scriptlet>
</ScriptletAttributeDefinition>
```

eduPersonTargetedID (ePTI)

- ePTI is a *opaque, directed, persistent* identifier for the user
 - *opaque*: doesn't give the user's identity away
 - *directed*: each SP sees a different value
 - *persistent*: the SP will see the same value every time the user comes back to them
- Primary use is for *personalisation*
- ePTI is *not* stored in your directory
 - options are storage-backed and computed

Storage-backed ePTI

- Store opaque (e.g., random) tokens in a DB
- Pro:
 - Supports more future SAML functionality
 - Supports ePTI revocation for privacy purposes
 - No problems with local identifier re-use
- Con:
 - Not bundled with 1.x IdP, so not many examples
 - Basic implementation bundled with 2.0 IdP
 - Fully resilient implementation is more complex

Computed ePTI (1)

- Mix (hash) together:
 - a secret
 - a unique (non-reassigned) local identifier
 - probably *not* the login name
 - most directories have some kind of UUID/GUID
 - the SP's entity name
- Pro:
 - No storage required
 - Implementation bundled with 1.x IdP

Computed ePTI (2)

- Con:
 - Doesn't support advanced SAML functionality
 - Doesn't support revocability
 - If SHA-1 is broken, becomes insecure
 - Reuse of local identifier causes ePTI reuse
 - and SPs really don't want that to happen, *ever*
- Summary: computed ePTI is acceptable for now if carefully implemented
- ... but expect to need to migrate

eduPersonEntitlement (ePE)

- eduPersonGetOutOfJailFreeCard
- Value is arbitrary URI (e.g., URN or URL)
- Values can be agreed between IdP and SP
- Can be used to delegate authorisation to IdP
- E.g., “IdP says OK to access resource X”
- Multi-valued: each user may have many
- ... only release values appropriate to each SP

Scripting eduPersonEntitlement

```
<ScriptletAttributeDefinition
  id="urn:mace:dir:attribute-def:eduPersonEntitlement">
  <DataConnectorDependency requires="directory"/>
  <AttributeDependency
    requires="urn:mace:dir:attribute-def:eduPersonAffiliation" />
  <Scriptlet><![CDATA[
    Attributes attributes =
      dependencies.getConnectorResolution("directory");

    Attribute entitlement = attributes.get("eduPersonEntitlement");

    // add values from directory
    for (int i = 0; entitlement != null && i < entitlement.size(); i++) {
      resolverAttribute.addValue(entitlement.get(i));
    }

    // add common-lib-terms for staff and student
    Attribute attribute = attributes.get("eduPersonAffiliation");
    if (attribute.contains("staff") || attribute.contains("student")) {
      resolverAttribute.addValue("http://sp.example.com/contract0732");
    }
  ]]>
  </Scriptlet>
</ScriptletAttributeDefinition>
```

eduPersonPrincipalName (ePPN)

- Usually scoped version of login name
 - `my.name@ed.ac.uk`
- This counts as personal information
- Privacy and legal concerns mean use as last resort
- Can often be replaced by ePTI or ePE

Contacts

- UK federation: <http://www.ukfederation.org.uk/>
- Technical Recommendations for Participants:
 - <http://tinyurl.com/ywm895>
- Recommendations for use of personal data:
 - <http://tinyurl.com/2fud6b>
- Speaker: ian@iay.org.uk
- And you've been good this year, so...

...all right, you can have a pony



photo © cc-by-2.0 by flickr user <http://flickr.com/photos/jonmclean/>

Contacts

- UK federation: <http://www.ukfederation.org.uk/>
- Technical Recommendations for Participants:
 - <http://tinyurl.com/ywm895>
- Recommendations for use of personal data:
 - <http://tinyurl.com/2fud6b>
- Speaker: ian@iay.org.uk