Connecting a Shibboleth IdP to your identity infrastructure

Nate Klingenstein
Internet2
ndk@internet2.edu

国立情報学研究所

２００７年６月１日

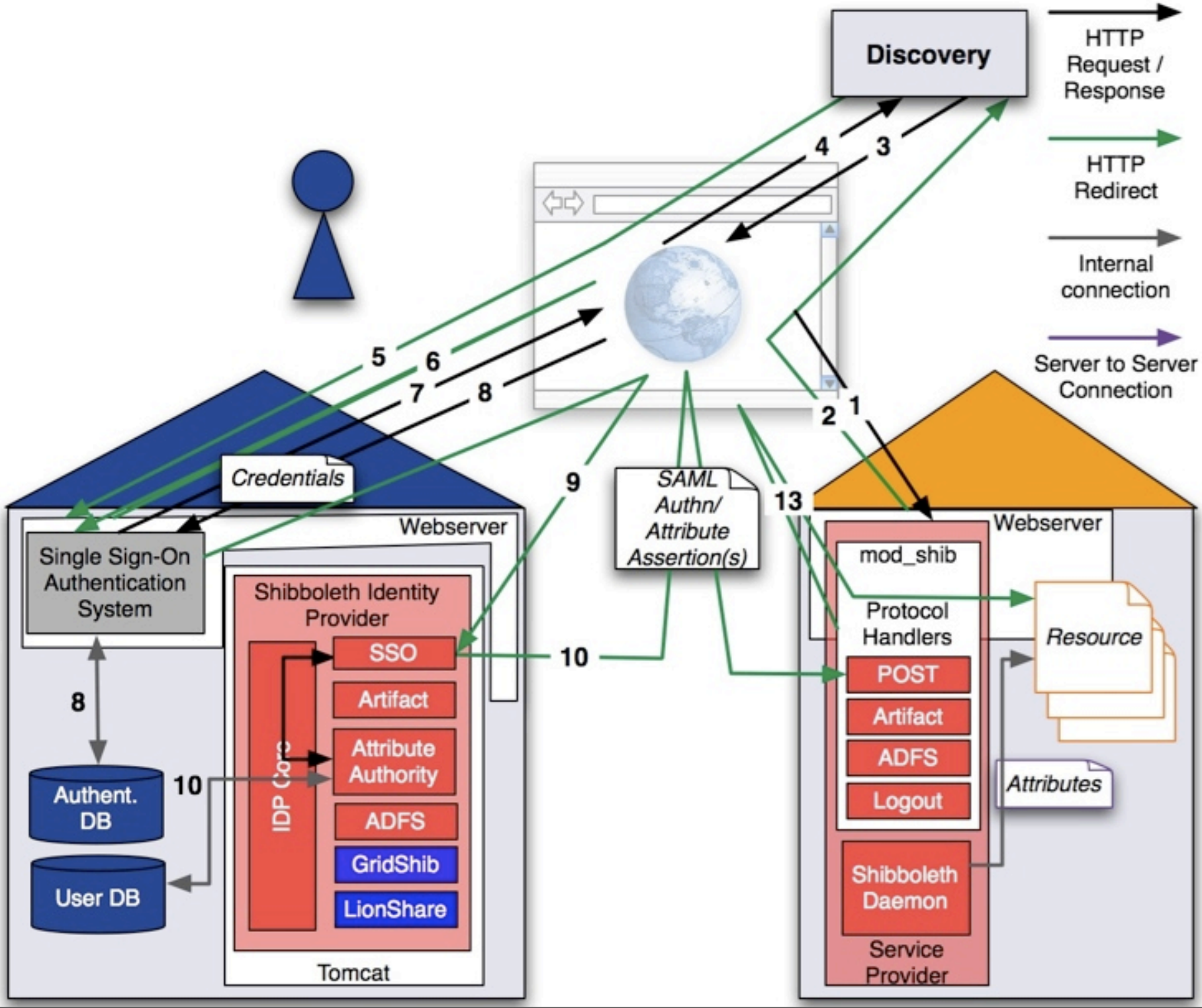
Our Goals (Again)

- Give the application what it needs
- Centralize as much as possible
  - But don't force it
- Have as much security as possible
  - But it still has to be usable, or people will just make their own
- Keep things as simple as possible
  - But less complexity now sometimes means more complexity later

Shibboleth Integration

- Shibboleth was designed to use the attributes and authentication you already have

- Authentication
- Attributes
  - Roles, groups, privileges, authorizations, etc.
- Be ready: this will be technical


Discovery
HTTP Request / Response
HTTP Redirect
Internal connection
Server to Server Connection
4
3
5
6
7
8
2
1
Credentials
9
SAML Authn/ Attribute Assertion(s)
13
Webserver
Single Sign-On Authentication System
Shibboleth Identity Provider
SSO
Artifact
Attribute Authority
ADFS
GridShib
LionShare
IDP Core
Tomcat
8
10
Authent. DB
User DB
10
Webserver
mod_shib
Protocol Handlers
POST
Artifact
ADFS
Logout
Shibboleth Daemon
Service Provider
Resource
Attributes

Connecting an Authentication Source

- Shibboleth 1.3 doesn't authenticate users itself
- Instead, you can use any web-based authentication method
  - PKI
  - Kerberos
  - CAS
  - Username/Password
  - LDAP Authentication
  - Most commercial SSO's

Connecting an Authentication Source

- Must be able to populate the REMOTE_USER header variable
- Multiple different authentication systems can be used by one IdP
- The IdP can send a default or dynamic AuthenticationMethod
  - This tells the SP how a user was authenticated technically, but not how to decide trust

Authentication in Shibboleth 2.0

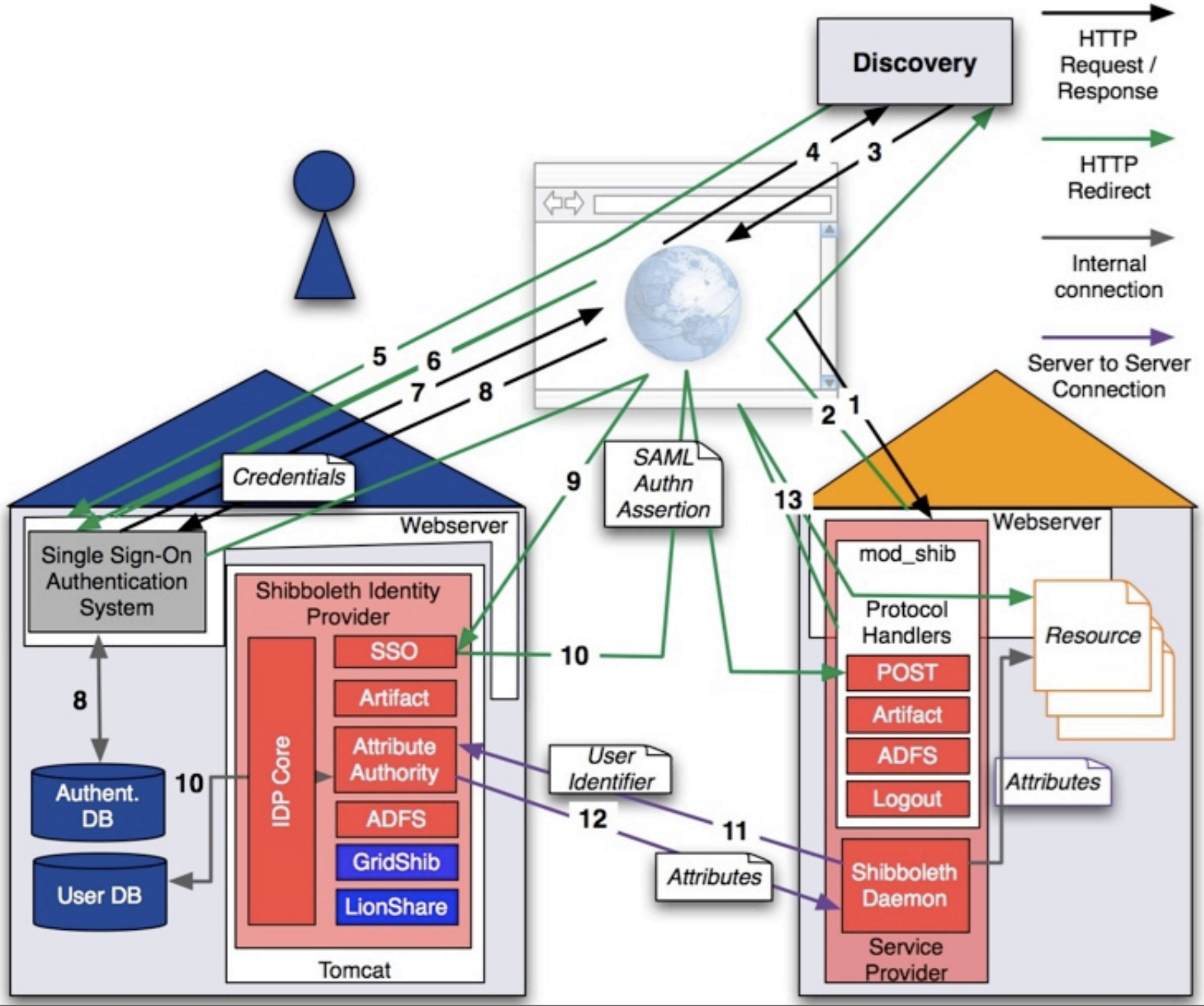
- SAML 2.0 has AuthnRequests, which can contain instructions on how to authenticate a user
- To honor some of these requests, the IdP must handle the message before the authentication method
  - isPassive
  - AuthenticationContext
- New authentication API with hooks in different places

Shibboleth 1.3 Authentication Assertion

<Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" AssertionID="_64ca3df56f9067c56ded9fc3cdc1e099" IssueInstant="2007-05-29T07:44:11.769Z" Issuer="https://idp.testshib.org/shibboleth/testshib/idp" MajorVersion="1" MinorVersion="1"><Conditions NotBefore="2007-05-29T07:44:11.769Z" NotOnOrAfter="2007-05-29T07:49:11.769Z"><AudienceRestrictionCondition><Audience>urn:mace:shibboleth:testshib</Audience><Audience>https://stc-test3.cis.brown.edu/shibboleth/testshib/sp</Audience></AudienceRestrictionCondition></Conditions><AuthenticationStatement AuthenticationInstant="2007-05-29T07:44:11.769Z" AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:unspecified"><Subject><NameIdentifier Format="urn:mace:shibboleth:1.0:nameIdentifier" NameQualifier="https://idp.testshib.org/shibboleth/testshib/idp">_53d6a655f0c0a6b35aeec839c21fec6a</NameIdentifier><SubjectConfirmation><ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</ConfirmationMethod></SubjectConfirmation></Subject><SubjectLocality IPAddress="193.167.182.40"></SubjectLocality></AuthenticationStatement></Assertion>

Connecting Attributes to Shibboleth 1.3

- The Shibboleth IdP has a powerful "Attribute Authority" (AA) built into it
- The AA:
  - Connects to attribute sources
  - Transforms attributes
  - Filters the attributes to be told to an SP
  - Places the attributes in an assertion and sends them to the SP


Discovery
HTTP Request / Response
HTTP Redirect
Internal connection
Server to Server Connection
4
3
5
6
7
8
2
1
9
13
Credentials
SAML Authn Assertion
Webserver
Single Sign-On Authentication System
Shibboleth Identity Provider
SSO
Artifact
Attribute Authority
ADFS
GridShib
LionShare
IDP Core
10
8
Authent. DB
User DB
10
User Identifier
12
11
Attributes
Tomcat
Webserver
mod_shib
Protocol Handlers
POST
Artifact
ADFS
Logout
Resource
Attributes
Shibboleth Daemon
Service Provider

Where does this information start or end?

- Source systems often have more information than applications
  - SIS, HR, departments, etc.
  - Expiration, “opt-in”, “opt-out”, “delegation”, etc.
- Applications also often have more information than source systems
  - Preferences, specific privileges, application-specific attributes
- Don’t over- or under-centralize

Getting Attributes from SOR to Shibboleth

- JNDI directory connector
  - Connects to LDAP directories
  - Using LDAPS/StartTLS or LDAP
- JDBC database connector
  - Connects to relational databases
  - SQL queries may be performed
- UK Federation working on very simple Active Directory installation

Transforming Attributes

- The built-in attribute resolver can do many transformations
  - Automatically adds scope
  - Changes name to anything appropriate for the SAML assertion
    - So you can have a local name for an attribute and a federated name for an attribute
    - Or one attribute with different names for different SP's
- If that's not enough, you can write a Java class to do anything

Transforming Attributes in Shibboleth 2.0

- Much more powerful
- Built on top of Apache's Velocity templating engine
- Able to use scripts written in (almost) any language
  - Except Perl, because there is no interpreter

Attribute Release Policies (ARP's)

- Now that we have collected and transformed attributes, we need to decide which ones to send
- Attribute Release Policies (ARP's) allow the IdP to choose which attributes to send
- Default deny

ARP Syntax

```
<Rule>
    <Target>
        <AnyTarget/>
    </Target>
    <Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation">
        <AnyValue release="permit"/>
    </Attribute>
</Rule>
```

ARP Syntax

- Four things can be used in the decision:
  - Attribute name
  - Attribute value
  - SP entityID
  - Username
- More options in ShARPE, done by MAMS in Australia

Attribute Filter Policies

- ARP's are being replaced in Shibboleth 2.0 with more flexible and powerful "attribute filter policies"
- Allows for dependencies, group-based attribute release policies, and more

Shibboleth 1.3 Attribute Assertion

<Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion" AssertionID="_27189d3cb12b41a7bb8ef10ce1aa73ee" IssueInstant="2007-05-29T07:35:56.255Z" Issuer="https://idp.testshib.org/shibboleth/testshib/idp" MajorVersion="1" MinorVersion="1"><Conditions NotBefore="2007-05-29T07:35:56.255Z" NotOnOrAfter="2007-05-29T08:05:56.255Z"><AudienceRestrictionCondition><Audience>https://sp.testshib.org/shibboleth/testshib/sp</Audience></AudienceRestrictionCondition></Conditions><AttributeStatement><Subject><NameIdentifier Format="urn:mace:shibboleth:1.0:nameIdentifier" NameQualifier="https://idp.testshib.org/shibboleth/testshib/idp">_3a9ea43f5265fa3fa767cb9c9fef8472</NameIdentifier></Subject><Attribute AttributeName="urn:mace:dir:attribute-def:eduPersonPrincipalName" AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"><AttributeValue Scope="testshib.org">myself</AttributeValue></Attribute></AttributeStatement></Assertion>

SAML Attributes

- A name
  - May be different in SAML; Shibboleth uses URI naming
  - Names are different in SAML 2.0, SAML 1.1, LDAP, and the other protocols from earlier
- At least one value
  - Sometimes more…
- A NameFormat or AttributeNamespace
  - Maybe a mistake...

SAML Attributes

- Practically anything can be placed into this structure
  - But the person who receives it must be able to understand it
- May include other things, like scope, hierarchy, XACML policies, or even SAML assertions
- The information accompanying the attribute can affect its meaning
  - Other attributes, the rest of the assertion, etc.

The Scope Experience

- Shibboleth 1.3 and earlier create scoped attributes with structure:
  - <AttributeValue Scope="nii.ac.jp">
  - Not “member@nii.ac.jp"
- おっと！
- Painful mistake has meant less interoperability, even though applications see “member@nii.ac.jp"
- Shibboleth 2.0 will fix this for SAML

Attribute Encoders

- Shibboleth 2.0 speaks many protocols simultaneously
- Each protocol has its own representation of attribute information
- An “attribute” inside the IdP will have many encoders attached to it
- These encoders will run to create the right attribute when it’s needed for a protocol

Commercial Interoperability

- Shibboleth 1.3 can be configured to talk to many SAML 1.1 providers
  - Burton Group Catalyst SAML “Bake-off”
  - But several specific changes are sometimes needed, and knowing which requires detailed SAML knowledge
- Shibboleth 2.0 SP has already been tested extensively with 2 SAML 2.0 vendor products
- Interoperability has many colors

Privileges, Attributes, Groups, and Roles

- The types of information have different meanings to us
  - An attribute is some piece of information about someone
  - A group is a group of people
  - An “privilege” is the right to do something
  - A role is something someone does for their organization

Privileges, Attributes, Groups, and Roles

- We “know” what the difference is... but what is the difference in syntax?
- Don’t think attribute, role, group, or privilege
- Think “what data structure is needed?”

Privileges, Attributes, Groups, and Roles

- Return to goals: what do applications want to receive?
  - Applications seem to like name/value pairs, or even just values
  - For smarter applications, more complicated expression can be used
- Let's see what we can put into name/value form

What if everything were an attribute?

Name	Value
emailAddress	ndk@internet2.edu
Edit wiki page	Is allowed to
Go Club	Member
University Role	Professor