Notes on Shibboleth

A system of federated administration
(more than an authorisation system)

These notes attempt to be a distillation of the excellent and readable 'Shibboleth-Architecture DRAFT v05', dated 2 May 2002 at
http://shibboleth.internet2.edu/draft-internet2-shibboleth-arch-v05.html

If you're interested, please read that document. Anything that is unclear in these notes is because I've misinterpreted something!! Text highlighting like this reflects hyperlinks to definitions (only works in the electronic version)

Document History

Version	Date	Comments
0.1	23 Jan 2004	First draft version, on reading the draft v05 document (2 May 02). Not yet proofed.

1. Introduction

Current problem: by having to manage lots of 'foreign' (non-'home' users), the resource provider (at one university) winds up in the role of system administrator other universities' users, without actually relieving those other universities of any of their system administration tasks.

- collaboration and resource sharing among institutions (and virtual organisations – VOs)
- requirement to have an access control scheme that is NOT based on identity (pseudonymity etc. for academics/students)
- "One difference between Shibboleth and other efforts in the access control arena is Shibboleth's emphasis on user privacy and control over information release."
- Shibboleth provides federated administration: a resource provider leaves the administration of user identities and attributes to the users' origin site

Shibboleth, then, is a system for securely transferring attributes about a user from the user's origin site to a resource provider site

- Shibboleth assumes that users employ browsers and that the resources are accessible via standard browser technologies
- Of course, AAs and attribute release policies only work with resource provider sites that have implemented Shibboleth's protocols for acquiring attributes.

2. Definitions

2.1. Attribute Authority (AA)

Each origin site (i.e. a site with administrative authority over users who access resources at remote providers) has its own AA. The AA's job is to provide attributes about a user to a resource provider. But the AA also has the responsibility of providing a means for users to specify exactly which of their allowable attributes gets sent to each site they visit.

The AA can store attributes directly, or it may itself do look-ups into the institution's LDAP (or other database).

2.2. Shibboleth Attribute Requester (SHAR)

The SHAR is based at the service provider's web resource and interacts with the Attribute Authority (AA) at the origin site to get attributes about the user.

2.3. Attribute release policy

One important aspect of the Shibboleth model is that users are empowered to choose which attributes (which may include their real name) are revealed to service providers. It is expected that the user will probably choose a default policy for use with new services, but that there will probably be a web page where s/he can edit the attributes that are sent when service providers request credentials from the AA.

(Shibboleth doesn't specify how attribute release policies are stored and managed.)

In contrast, services will have *Attribute Acceptance Policies*. These will be used by the SHAR to determine which attributes are required, and which are acceptable etc. etc.

2.4. Attribute query message (AQM)

The attribute request that the SHAR sends to the AA.

2.5. Attribute response message (ARM)

The response that the AA sends to the SHAR.

2.6. Handle and 'Handle Service' (HS)

A handle is a temporary name or pseudonym (used to protect the privacy of the user). The HS is responsible for making sure the user is authenticated locally at the origin site, and for creating a handle that can be used to retrieve attributes about the user.

2.7. Shibboleth Indexical Reference Establisher (SHIRE)

(Keeping up the Tolkein parallels of Middleware and Middle Earth).
The part of the "Shibbolized" web server that manages the process of acquiring a handle.

2.8. Where Are You From? (WAYF) service

Its primary job is to map an origin site name (like harvard.edu) to the HS information for that site. The WAYF's other responsibility is to ask the user's HS to send a handle for the user to the SHIRE. Conceptually, the WAYF could live anywhere on the Internet, and could be shared (e.g. like DNS), but could also be replicated at every SHIRE.

2.9. Attribute Query Handle Request

See handle request below.

3. The mechanics

User reaches service via HTTP (web). Service uses its SHAR and asks AA at user's home institutions for credentials (attributes) via an AQM. The AA sends an ARM containing the attributes back to the SHAR.

However, to do all this, the service (or SHAR) needs to identify the user in some way – but the user should be anonymous! Therefore, the SHAR uses a 'handle' to identify the users. There are three distinct mechanisms for this process:

- Direct Access to Destination Site
- Local Navigation or Resource Listing Sites and
- Client Certificates

3.1. Direct Access to Destination Site

The SHIRE (at the service provider's machine) has a "pointing" reference to that user and uses that connection with the browser to help initiate the process of securely getting a handle for the user.

i.e. the server is not allowed to know is who the person requesting access, but is able to get the attention of their home handle service (HS) and point out the users' browser and say 'that one!'.

How does the SHIRE know where the user's *home* is? It uses the Where Are You From? (WAYF) service, of course! This tends to use information, such as .ox.ac.uk to know to immediately contact the Oxford HS. However, the WAYF may directly (and obviously) ask the user and use that information to choose the right HS.

- The SHIRE then asks the (correct) HS for a handle (= the handle request).
- User then may have to log into their home site's authentication system (if they haven't done this already.
- The HS gives a 'handle package' to the user's browser to post in an HTML form to the SHIRE

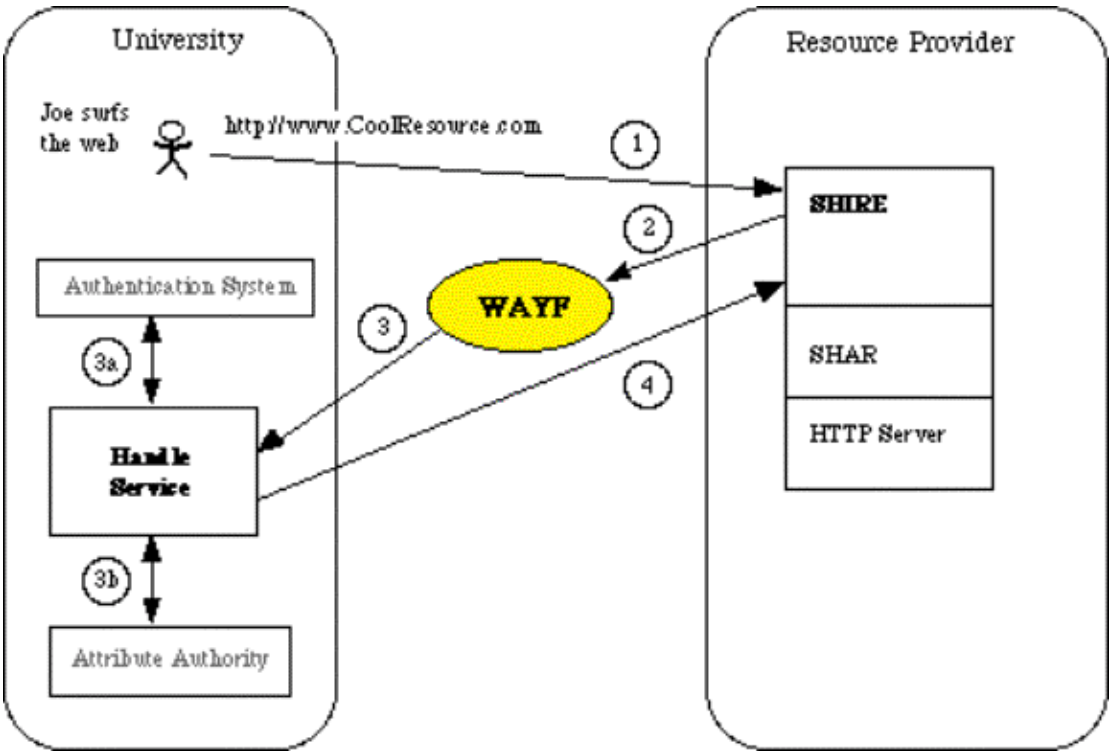

Diagram reproduced (without permission) from the original document.

After the above has been completed, the SHIRE can send an AQM to the AA and receives the attributes back in an ARM (as mentioned above).

Why is the Handle Service seen as separate from the Attribute Authority? Because there could be many AAs at the home institution. However, both the HS and the AA must jointly understand that a given handle belongs to a particular user.

3.2. Local Navigation or Resource Listing Sites

(Relevant for portal implementation!)
Assumes that the access to the service was via a 'thin local portal' at the origin site. The link on the page would contain a "handle acceptance URL" and there would be no need for a WAYF. The browser arrives at the service with the 'handle package' as described above.

3.3. Client Certificates

Client certificates **can be used for local authentication** – but **Shibboleth has nothing to say about this** (as local authentication is none of its business).

There is an acceptance that Shibboleth should 'make room' for the use of Client Certificates in a way that would make the WAYF and the handle requests unnecessary. Presumably, the certificate contains the attributes that the service requires and the certificate is 'read'.

However, the Internet2 group may not have agreed on mechanisms for this yet. Issues such as pseudonymity, the changing of attributes and revocation need to be standardised, presumably.

4. SAML (Security Assertion Markup Language)

The following is taken straight from the 'Shibboleth Architecture Draft' paper:

> "Shibboleth has much in common with SAML, an emerging OASIS standard... Shibboleth uses SAML formats and binding protocols whenever possible and appropriate. Of particular note: Shibboleth uses the SAML query and response protocol and formats for the AQM and ARM messages, and Shibboleth uses SAML's attribute statement and assertion format. As SAML evolves, Shibboleth will evolve along with it. Shibboleth, though, is both narrower and broader than SAML.
>
> "Shibboleth is narrower in that its use cases are far more limited than those of SAML: Shibboleth focuses on the browser user, while SAML also includes complex scenarios involving buyers, sellers, and brokers. SAML also specifies message exchanges and formats for authorization decisions, in addition to the attribute queries and assertions that are used by Shibboleth."

(MN's thoughts: – where services are not HTTP-based, e.g. Z39.50 database access, maybe SAML could be used eventually to bridge the gap between browser-dependent, pure Shibboleth and client software that could speak SAML to the local HS and AA to present credentials to the service?).

5. Etymology

This section does not come from the main 'Shibboleth Architecture Draft' paper.

5.1. Hebrew word used by Jephthah (Judges xii. 4-6)

In the story, two Semitic tribes, the Ephraimites and the Gileadites, have a great battle. The Gileadites defeat the Ephraimites, and set up a blockade to catch the fleeing Ephraimites. The sentries asked each person to say the word shibboleth. The Ephraimites, who had no *sh* sound in their language, pronounced the word with an *s* and were thereby unmasked as the enemy and slaughtered.

5.2. In UK English, it also has a more negative(!) meaning

The OED: "A catchword or formula adopted by a party or sect".

SCOTT, 1894: "Knaves and fools invent catch-words and shibboleths to keep them ['honest' persons] from coming to a just understanding".

I think we'll stick with the idea that it lets people in who should come 'in' and also gives a vague identification of who they are.

6. References

6.1. Main reference

6.1.1. http://shibboleth.internet2.edu/draft-internet2-shibboleth-arch-v05.html