

SERAPIS Project

Integrating Shibboleth with the AHDS Catalogue and Delivery System

**Arts and Humanities Data Service
King's College London**

Author: Sanjay Vivek

Contact: Mark Hedges

7 March 2007



Table Of Contents

<u>1.</u>	<u>Introduction</u>	<u>4</u>
<u>2.</u>	<u>Shibboleth Attributes</u>	<u>5</u>
<u>3.</u>	<u>Process Logic.....</u>	<u>6</u>
<u>4.</u>	<u>LazySessions.....</u>	<u>8</u>

1. Introduction

The document describes the integration of the AHDS Catalogue and Delivery system with Shibboleth. The AHDS Catalogue and Delivery system (hereafter referred to as the catalogue for brevity) is a web-based application for searching, browsing and downloading resources held in the AHDS repository. Access to many of these resources is free and unrestricted. In other cases, however, access is subject to restrictions imposed by the terms of the licence agreements made with the depositors or resource owners.

Currently, access to restricted resources is managed by a local registration and login system that is specific to the repository. In the SERAPIS project, this local registration system was replaced (or rather supplemented) by a system based on Shibboleth. The catalogue application is protected by a Shibboleth Service Provider registered with the SDSS federation, and will be accessible to users within any federated IdP. The local registration system was retained in addition to the Shibboleth-based system as some users (e.g. private researchers) will not be affiliated to an institution with a Shibboleth IdP.

In brief, the modified catalogue application works as follows:

- A user wishes to access a protected resource and clicks on the corresponding download link.
- The user is redirected to a page that lets the user decide whether to access the protected resource via Shibboleth (if the user's institution is part of the UK Federation) or via the local registration system (otherwise).
- If the Shibboleth option is selected, the user is sent to the UK Federation WAYF. The WAYF redirects the user to the selected local institutional Single Sign On (SSO) system, which allows the user to be authenticated.
- Once the user is authenticated, a range of Shib_Headers is available to the catalogue application, which is protected by a Shibboleth Service Provider. The catalogue application uses these Shib_Headers to determine whether a user is authorized to access the protected resource.
- Once the user has been authorized to access the resource, the catalogue application redirects the user to the targeted resource. The catalogue application initiates a LazySession and tells the WAYF where to redirect the user.

2. Shibboleth Attributes

Each service within the UK Federation requires certain attributes to be released by an IdP in order for a user to access the service. The full list of attributes that are required for access to specific UK Federation services is provided at <http://sdss.ac.uk/content/Documents/AttributeUsage?AttributeUsage>. The default attribute release policy for a Shibboleth IdP releases only the eduPersonScopedAffiliation attribute to Service Providers. The arp.xml policy file for an IdP has to be configured to release any other attributes.

Due to licence agreements between the AHDS and the collection owners, certain collections are restricted to “registered” users, and every download of a collection has to be recorded and associated with the user, who must in principle be traceable. In addition, some collections are further restricted to users in higher and further education.

The obvious choice for identifying the user in a Shibboleth context is the eduPersonPrincipalName (EPPN) attribute. However, in practice it is unlikely that IdPs will release the EPPN attribute for privacy reasons. An alternative would be to use the opaque identifier eduPersonTargettedID (EPTI), which is guaranteed to be unique for a given combination of user, IdP and SP. However, the UK Federation policy on the usage of EPTI is unclear as yet, consequently in the initial prototype version of the catalogue application EPPN is used, although this is likely to be replaced in the live system.

In summary, in this prototype the attributes that need to be released and seen by our Shibboleth SP for accessing restricted collections are:

- eduPersonScopedAffiliation (EPSA), e.g. staff@ahds.ac.uk, student@ahds.ac.uk. The domain component of these is used to determine whether a user is affiliated to a higher or further education institution.
- eduPersonPrincipalName (EPPN) (e.g. sanjay.vivek@ahds.ac.uk), to link a download to a user and to provide traceability.

3. Process Logic

This section describes the changes to the processing logic within the catalogue code required to support Shibboleth-based access management for restricted resources. This processing logic is represented in Figure 1.

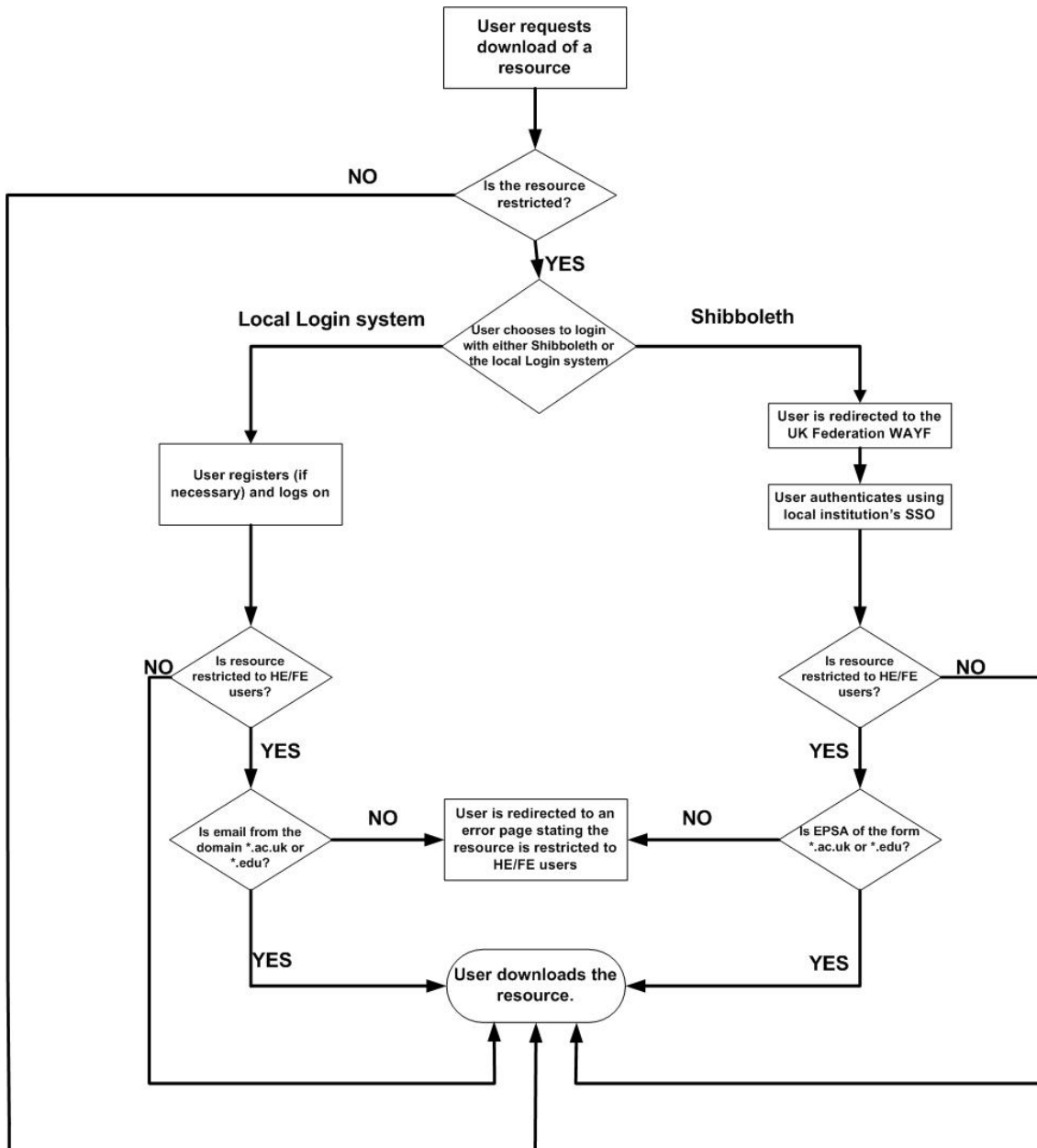


Figure 1 Access to restricted AHDS resources

In brief, the processing proceeds as follows:

1. A user wishes to access a resource and clicks on the corresponding “Download” link.
2. If the resource is restricted, then the user is redirected to a page that lets the user decide whether to access the protected resource via Shibboleth (if the user belongs to an institution that is part of the UK Federation) or via the local registration system (otherwise). If the resource is not restricted, the user is allowed to access it freely.
3. If the user chooses the Shibboleth option, he is send to the UK Federation WAYF. The WAYF redirects the user to the institutional Single Sign On (SSO) and the user subsequently authenticates himself.
4. Once the user is authenticated, the EPSA and EPPN attributes are requested from the IdP and passed to the catalogue application (which is protected by a Shibboleth SP). The catalogue application checks the EPSA attribute to determine whether the user is allowed to access the resource. The EPSA attribute will return a null value if the attribute is not within the scope defined in the AAP.xml, in which case access is rejected; otherwise it is in general permitted. Some of the protected resources, however, are further restricted to users who are associated with an institution of higher or further education; in this case the user will only be able to access the restricted resource if the domain component of the EPSA indicates that the institution is HE or FE (the prototype grants access if the domain is of the form of *.ac.uk or *.edu). The EPPN is requested for reasons described in Section 2. Currently, if the EPPN attribute is not asserted, the user has to use the local registration system to access restricted resources, although this will be modified in the live system.

Once the user has been authorized to access the resource, the catalogue application redirects the user to the targeted resource. The catalogue application initiates a LazySession and tells the WAYF where to redirect the user. The LazySessions feature is described in greater detail in Section 4.

The key Java source code for managing access to restricted resources can be found in the uk.ac.ahds.crosscat.download package, which has been modified to support the processing logic described above. The modified code is on a separate branch of the catalogue source in the Subversion source repository; this branch will be merged with the truck of the source after the completion of the SERAPIS project, as a number of unrelated changes have been made to the catalogue application since the start of the project.

4. LazySessions

In most deployments, session setup is handled automatically by configuring rules based on resource URLs or with platform-specific web server configuration. For example, to Shibboleth-protect access to <https://ahds.ac.uk/secure>:

```
<Location /secure>
    AuthType shibboleth
    ShibRequireSession On
    require valid-user
</Location>
```

The above configuration requires users to authenticate and authorize themselves via Shibboleth if the user wishes to access any files in /secure.

However, this was not possible in our scenario because our resource URLs are constructed on the fly and are dynamic URLs. This requires a finer degree of control and requires the session to be initiated from within the application code.

The solution implemented was for the catalogue application to make use of Shibboleth's LazySessions feature to enable a Shibboleth session even when the user has not authenticated himself. The LazySessions feature lets the application decide if the user has to authenticate himself before establishing a Shibboleth session. In normal deployments, the web server always initiates the authentication but this is not the case when using LazySessions. The LazySessions feature requires the directory that is protected by Shibboleth to load the Shibboleth module in a passive mode. The RequireSession attribute of the Path or Host element protecting it is set to false:

```
AuthType shibboleth
require shibboleth
ShibRequireSession off
```

This application must be aware of three pieces of information:

1. **handlerURL** attribute of a Sessions element: The URL of the handler associated with this Application.
2. **location** attribute of a SessionInitiator element: The session initiator you want to use.
3. **target**: The application URL that should be accessed after the session is established; usually, this will be the application's own URL.

These three pieces of information must be combined by the application to an appropriately formed URL to trigger session initiation as follows. To request a session, the application returns an HTTP redirect that sends the browser to the handler URL with a parameter, and the *target*, containing the URL of the resource to return to with a

session. In the case of the catalogue application, this is the URL that triggers the redirect. The Shibboleth module will generate the redirect the user to the requested resource after he authenticates himself at the WAYF and the rest proceeds as a standard Shibboleth flow. This combined URL takes the form:

`https://Sessions_handlerURLSessionInitiator_locationtarget=applicationURL`

For example, the catalogue application with a link to a protected resource, responds to a user click by redirecting the browser (after authentication and authorization) to:

`https://xenophobe.ahds.ac.uk:440/Shibboleth.sso/WAYF/SDSS?target=https%3A%2F%2Fxenophobe.ahds.ac.uk%3A440%2Fcatalogue%2Fdownload.htm%3Furi%3D" + downloadIdentifier + "%26type%3D" + downloadType`

Where downloadIdentifier and downloadType are 2 different parameters relevant to the catalogue application. The complete applicationURL without URL encoding is:

<https://xenophobe.ahds.ac.uk:440/catalogue/download.htm?uri=downloadIdentifier&type=coll>