



# **ShibboLEAP Project**

**Final Report:**

**King's College London**

**KING'S**  
*College*  
**LONDON**

**University of London**

**Richard Warren**

**May 2006**

## Overview of Shibboleth implementation at King's College London

This report provides an overview of the Shibboleth implementation at King's College London, highlighting the implementation and configuration decisions along with issues identified and overcome. It explains the reasons behind the College choosing to adopt the Shibboleth and where it can be used to enhance access to resources for the College users.

The document highlights the steps taken to get a functioning Identity Provider and includes some of the configuration scripts and files. Whilst these are configured for use at King's College, they could, in-part, apply to any implementation of Shibboleth.

## Contents

Overview of Shibboleth implementation at King's College London	- 2 -
Contents	- 3 -
Background to Shibboleth	- 4 -
What is it?	- 4 -
How does it work?	- 5 -
The key benefits of Shibboleth are:	- 5 -
ShibboLEAP	- 5 -
Shibboleth implementation decisions at King's College London	- 6 -
Operating System Platform	- 6 -
Local Authentication System (LAS)	- 6 -
Lightweight Directory Access Protocol: (LDAP)	- 6 -
Secure Socket Layer (SSL) Certificate:	- 7 -
Apache, Java, Tomcat	- 7 -
Attribute Release Policies (ARP's)	- 7 -
User Databases	- 7 -
Implementation experience at King's College London	- 9 -
Firewall changes	- 9 -
DNS registration	- 9 -
Installation	- 9 -
Secure Socket Layer (SSL) Certificate	- 9 -
Apache/Java/Tomcat	- 9 -
Attribute Release Policies (ARP's)	- 10 -
Connection to LSE:	- 11 -
Connection to LSE project Wiki:	- 11 -
Current situation at King's College London	- 12 -
Future Plans relating to IdP implementation	- 12 -
Athens, Metalib	- 12 -
Appendix 1 – Installation at King's College London	- 14 -
Configuration files relating to Shibboleth IdP implementation.	- 14 -
Modifications to Shibboleth IdP files	- 15 -
Modifications to Apache files:	- 21 -
Modifications to Tomcat configuration files	- 23 -

## Background to Shibboleth

### What is it?

In simple terms, Shibboleth is a standards-based protocol for exchanging information about users through the web. It provides a Single Sign On (SSO) mechanism by allowing tailored attributes about a user to be passed between a user's home institution and a resource institution that avoids unnecessary multiple authentication mechanisms on each resource.

For example, in the non-Shibboleth world a user at site A may request access to a resource in site B. Whilst the user may have already authenticated against Site A to gain access to the resources there, the resource at site B doesn't know anything about this person, so is not going to freely give out the resource. Therefore the user needs to be authenticated again against at the location of this new resource at site B, thus requiring the user to hold (for example), an additional username and password. Upon successful authentication the user would then be given access to that resource.

The problem with this is that for each institution holding user information, a new set of credentials is required, thus making the user remember multiple usernames and passwords for each protected resource. Invariably this gets messy and becomes an administrative burden for each of the resource controllers.

Shibboleth overcomes this issue but providing a mechanism for a user to be authenticated against their own home institution for resources at remote sites. It does not attempt to authenticate the user itself, but instead provides the means to which the user, or attributes about the user which are required for accessing the remote resource, to be transferred from the home institution to the remote institution securely.

This means that the user trying to access a Shibboleth protected resource at any institution (to which the home institution is associated with) would only need the same user credentials as at their home institution because it is there that they are being authenticated, not at the remote institution.

## How does it work?

Shibboleth requires two main elements:

1. Service Provider (SP) at a site which controls access to a resource to which users wish to have access to;
2. An Identity Provider (IdP), which resides at the site from which the user is located, and which knows the required details about the user.

When a user tries to access a resource which is protected by Shibboleth the SP redirects the user back to their home location to be authenticated, and upon successful authentication, a session is generated and the user is directed back to the resource they requested.

The key benefits of Shibboleth are:

- Reducing the administrative burden on maintaining multiple user databases (LDAP...).
- Less time is taken up with monotonous tasks such as password resetting.
- Users do not have to remember multiple usernames and passwords for each resource they require access to
- Athens and other resources are Shibboleth compliant, thus making a wide range and large number of resources available to the user.
- Most of the processing is done at the member institutions so there is very little requirement for the central infrastructure at each resource.

## ShibboLEAP

ShibboLEAP is a consortium of 7 institutions led by London School of Economics (LSE),

- To create a Shibboleth Identity-Provider ('Origin') service for all academic and support staff (at each of seven partner institutions in the SHERPA-LEAP consortium)
- Involved in controlled access to their respective institutional Eprints servers
- Implement modifications to their respective Eprints servers to enable them all as Shibboleth Resource-Providers ('Targets')

The 6 supporting partners are Birkbeck College, Imperial College, King's College London, Royal Holloway, School of Oriental & African Studies (SOAS), and University College London.

## Shibboleth implementation decisions at King's College London

### Operating System Platform

The backbone of the IT infrastructure at KCL is UNIX based. Email, Web and LDAP services are provided via Solaris operating systems, although steps are being taken to move some services towards Microsoft Active Directory services and Microsoft Exchange. However, for the implementation of Shibboleth at King's, it was decided to install the software on new servers with Linux as the main operating system to make use of resources available within the IT department and minimise impact on normal production. Additionally have two servers allowed flexibility in configuration and problem solving described later in this document.

Fedora & Red Hat Enterprise Linux operating systems were considered and a decision to go with Red Hat was based on problems experienced with Fedora (explained later in this document).

### Location of Servers

Although LDAP, Shibboleth and Apache can run on the same server, for King's only Apache and Shibboleth were installed on the new Red Hat server with LDAP being on a separate server.

### Local Authentication System (LAS)

A LAS is required for the web server to be able to query the LDAP server. Various options were available, but we opted for `mod_auth_ldap` as this was straightforward to configure and worked without problems.

### Lightweight Directory Access Protocol: (LDAP)

King's already had LDAP authentication for other services, so there was no need to implement a separate solution. By cloning the live LDAP server (an IPlanet LDAP server for Solaris) we could test our IdP implementation, including installing the `eduPerson` class into the LDAP server, without interrupting the live service.

Cloning the LDAP server was undertaken by the UNIX (Corporate Systems) team who are responsible for maintaining the UNIX services.

The test server was set up and configured within two days, and subsequently the `eduPerson` object class was installed. For testing purposes a selection of user accounts were manually created and the `eduPerson` class attributes were populated with a variety of values to allow us to test the passing of data between the IdP and the Service Provider at LSE.

Decisions surrounding the configuration of the LDAP server were taken by other teams with the exception of the decision on `eduPerson` attribute requirements which were death by Corporate Applications team.

Such decisions as "what is the unique identifier going to be", and the organisation of the OU's had already been resolved, so it was straight

forward to configure the Shibboleth Attribute Release Policy to release the preconfigured attributes.

### Secure Socket Layer (SSL) Certificate:

It was recommended we use a GlobalSign SSL certificate so that testing against the SDSS federation would be possible. However, due to King's owning a 'wildcard' Comodo certificate which is suitable for all top-level domain resources, it was opted to make use of this certificate, after confirmation from LSE that it would work. As Shibboleth was installed at the top level domain (shibboleth.kcl.ac.uk), we were able to use this certificate instead of obtaining a new one thereby saving time on waiting for new certification to be purchased.

### Apache, Java, Tomcat

It was decided to install the latest available versions of Apache (v2), Java (Sun 1.5), Tomcat (5.5) as instructed by the implementation documentation. Although Systems' staff at King's have more experience with Apache 1.3, implementing Apache (v2) didn't prove to be troublesome.

### Attribute Release Policies (ARP's)

Attributes required by Shibboleth were clearly defined after consultation with LSE and the other participating institutions. As a minimum, the following attributes are populated in LDAP, and are sent between the IdP and SP:

- eduPersonPrincipalName (Remote\_User):
  - value: 'login@domain.ac.uk'
  - eg: [abcd1234@kcl.ac.uk](mailto:abcd1234@kcl.ac.uk)
- eduPersonScopedAffiliation (HTTP\_SHIB\_EP\_AFFILIATION):
  - value(s): "MEMBER" (compulsory) , "STUDENT", "STAFF"
- eduPersonTargetedID
  - encrypted value
  - used for holding persistent id information for multiple sessions.
- mail (HTTP\_SHIB\_INETORGPERSO\_MAIL):
  - value: email address.
- ou (HTTP\_SHIB\_INETORGPERSO\_OU): Department (for detection of members of library staff)

### User Databases

Amongst a large array of databases held at King's, one of the most fundamental is the "User Database" (UDB). This is maintained by the Corporate Applications staff, and it has feeds from a number of other databases including finance, personnel and the student system, and feeds outwards to services including the LDAP server, PAWS (Public Access Workstations) and more. LDAP receives information about accounts to be created, modified and deleted.

The UDB contains information on the users which is required for providing access to the resources and services they are entitled to. Included in this are the users' department, school and subject. By

employing the data in these fields, we can populate the eduPerson fields automatically by adding the additional data to the feed to the LDAP server. This functionality is currently in development and it would be relatively simple to implement any eduPerson requirements into the system as much of it is in place already.

Having the eduPerson object class implemented in the test server will allow for thorough testing prior to implementation on the live LDAP server.

## Implementation experience at King's College London

**This section outlines various issues which were raised during the installation, configuration and testing of the IdP.**

### Firewall changes

The local and institution firewall needed to be opened on specific ports to allow the IdP to talk to the service provider. The ports were 80 (http) 443 (https) 8080 and 8443.

### DNS registration

The shibboleth server was named as shibboleth.kcl.ac.uk. By registering as a top level domain service we could use the King's owned wildcard Comodo SSL certificate.

### Installation

We tried using Fedora for the initial installation, but testing with InQueue proved problematic, with sporadic transfer of data between the IdP and the InQueue Service Provider. After repeated attempts to resolve the issue (regenerating the temporary self-signed certificates, and refreshing the metadata), another server was commissioned in parallel, this time using Red Hat Enterprise. This allowed us to compare the differences. InQueue testing with the Red Hat server proved successful, so it was decided to stay with this platform.

### Secure Socket Layer (SSL) Certificate

After discussions between King's and LSE, it was agreed to use the Comodo wildcard certificate currently owned by KCL and use this for top-level domain servers (such as the web server and webmail server). The self signed certificates originally created for testing would not be secure enough for a live system.

Installation of the Comodo certificate was straightforward, though we initially did not obtain the full certificate bundle necessary for the service provider to verify that the correct information has been sent. Once this was realised, obtained and installed, subsequent connection testing was successful.

### Apache/Java/Tomcat

Installation was undertaken by following the point by point instructions as supplied on various websites and by the support provided by LSE. The following points outline the installation:

- 1) On-line documentation (<https://authdev.it.ohio-state.edu/twiki/bin/view/Shibboleth/InQueueIdPInstall>) obtained and appropriate fields completed to allow for a customised installation document.
- 2) IdP was installed from shibboleth-idp-1.3c.tar.gz downloaded from <http://wayf.internet2.edu/shibboleth/>
- 3) Installation was configured, leaving most options as default:
  - a. Tomcat home /usr/local/tomcat
  - b. Home /usr/local/shibboleth-idp

- 4) Configuring xml files where appropriate.
  - a. idp.xml
  - b. resolver.ldap.xml
  - c. arp.sites.xml

See the appendix for more information on the additions made to these files

### Attribute Release Policies (ARP's)

Initial problems connecting to the LSE SP were mostly down to certificate issues and mis-matching metadata. After connecting we found that attributes were not being received by the SP whilst they were being read by the IdP from LDAP correctly. The 'resolvertest' utility, a tool designed to verify whether attributes were being successfully read from the LDAP server displaying the attribute and its value, proved successful as the attribute values expected were being returned.

The resolver.ldap.xml and arp.sites.xml were slightly mis-configured, thus not releasing all the attributes correctly. Amending this allowed the eduPerson attributes to be received correctly, apart from the non-eduPerson attributes (cn, sn, givenName).

Further testing showed 2 different issues causing this problem, the first at the SP end where some attributes were being filtered out inadvertently; and secondly, the virtual host (8443) was missing two important lines, thus causing the main configuration to not work properly.

- `SSLVerifyClient optional_no_ca`
- `SSLOptions +StdEnvVars +ExportCertData`

Subsequent testing of ARP rules to only allow certain attributes to be sent if the target was LSE proved successful.

Identifying which eduPerson attributes are populated was decided through consultation with LSE and the other institutions.

- `eduPersonAffiliation` - value of "MEMBER" (at a minimum)
- `eduPersonPrincipalName` - [joebloggs@domainname.ac.uk](mailto:joebloggs@domainname.ac.uk)
- `eduPersonTargetedID` - encrypted value

Identifying and populating at least one additional attribute for distinguishing staff, students (and possibly researchers, etc) is still outstanding.

Other non-eduPerson attributes that could be allowed through are:

- `Mail (required)`
- `cn (e.g. abcd1234)`
- `departmentNumber (eg: qcc)`

## Lightweight Directory Access Protocol (LDAP)

King's configured a test LDAP server, with a handful of accounts and installed the eduPerson object class, manually populating the attributes for initial testing. KCL will clone their live LDAP server and automatically populate the attributes required via a feed from the UDB.

This decision is still to be resolved, but with the test server hosting a copy of the live LDAP data, and with the simplicity of modifying the Shibboleth configuration to point to a different server, this is not a complicated change.

The automatic feed and population of the eduPerson attributes needs to be tested thoroughly, but as mentioned previously, with the data feed between the UDB and the LDAP server already in place, adding additional fields which currently reside in the UDB to the feed would not be a complicated task and can be thoroughly tested without interrupting the live service.

### Connection to LSE:

(<https://gabriel.lse.ac.uk/simon/cgi-bin/printenv.pl>)

Connection to LSE Service Provider has been successful, passing (for testing purposes) the following attributes.

- eduPersonAffiliation
- eduPersonTargetedID
- eduPersonPrincipalname
- mail
- cn
- sn
- givenName

Based on this, the results in the following script were received:

- HTTP\_SHIB\_EP\_UNSCOPEDAFFILIATION student
- HTTP\_SHIB\_TARGETEDID  
4SPbLnuuvMSlc3WWYyMcjqw7Zvl=@kcl.ac.uk
- HTTP\_SHIB\_INETORGPERSO\_N\_GIVENNAME RICHARD
- HTTP\_SHIB\_INETORGPERSO\_N\_MAIL [richard.warren@kcl.ac.uk](mailto:richard.warren@kcl.ac.uk)
- HTTP\_SHIB\_PERSON\_COMMONNAME stty4985
- HTTP\_SHIB\_PERSON\_SURNAME WARREN
- REMOTE\_USER [richard.warren@kcl.ac.uk](mailto:richard.warren@kcl.ac.uk)

### Connection to LSE project Wiki:

(<https://gabriel.lse.ac.uk/twiki/bin/view/Projects/ShibboLeap>)

The only issue encountered with connecting to project Wiki was with registering the username containing an "@" symbol which is an illegal character in the Wiki username. This was changed to the "-" symbol, and subsequently the registration page information was updated to inform users of this requirement.

## Current situation at King's College London

King's have a working Identity Provider (IdP) implemented (see below for specific details) in a test set-up environment. The IdP is talking to a test LDAP server which is set up like the live one but contains a limited data set.

We have successfully tested against the LSE test Service Provider and the project WiKi, with proven transfer of the eduPerson attributes. All testing has been done using a clone of the LDAP server instead of our live instance to avoid any unnecessary disruption to other services.

King's makes use of various resources which are Shibboleth compliant, including Athens and Metalib, and are in a position to look at implementing access to these via shibboleth. Currently, we need to maintain multiple usernames and passwords for each user, and whilst this has been satisfactory, it is an administrative burden which we could alleviate by having a single resource for authentication (LDAP).

Athens access is currently via the standard Classic authentication, and Metalib is accessed by Exlibris's own "Patron Data System" (PDS), which authenticates the users via the LDAP server. With the Athens classic authentication mechanism terminating in July 2008 (though you can still use it if you pay for it), this would be a suitable candidate for shibbolizing in the near future.

## Future Plans relating to IdP implementation

King's is currently undertaking a project to incorporate Active Directory and Exchange server services to a wide user community. Whilst this would not have an impact on the current LDAP server and Shibboleth IdP, it may be a consideration to look at the benefits of providing LDAP and Shibboleth from Active Directory. By doing this we would have everything under the same umbrella, thus making the administration of the services more coherent, and less resource-heavy. Maintaining multiple systems reliant on different platforms – Windows, UNIX, Linux is costly in staff and time.

## Athens

Athens is used to provide access to around 7,000 ejournals, around 2,000 ebooks and around 100 databases via hundreds of different interfaces and services.

From an E-Resource and Athens support point of view the main issue is this: JISC will no longer support Athens from July 2008.

Before this July 2008, institutions will need to either:

1. Have decided to stick with Athens and therefore PAY to use the Athens service (NOTE if we stick with Athens we MUST move to AthensDA so users don't have to have a separate Athens username and password),

OR

2. Have implemented Shibboleth,
- OR
3. Some other SSO access solution

## Metalib

Metalib, from Exlibris, is a library portal which enables users to access their institutions e-collections, obtain relevant services and work in a personalized environment.

MetaLib works with Shibboleth and AthensDA and establishes an SSO session for resources using the Athens/Shibboleth gateway. Therefore if you have logged into MetaLib using Shibboleth / AthensDA, then you can link to an Athens resource and go straight into it without logging in again. The same should apply to any service we set-up locally to work with shibboleth.

We have to either adopt AthensDA and Shibboleth. Many other institutions have AthensDA already in place, but if we go with Shibboleth then we can leap frog AthensDA.

The other major issue is that publishers are more likely to adopt Shibboleth than Athens as they have to pay to use Athens Shibboleth is 'free' (apart from the resource requirement).

## Appendix 1 – Installation at King's College London

Software installed & configured on the server (Dell PowerEdge 2850)

RedHat Enterprise Linux v.3  
Apache v.2  
Tomcat v.5.5  
Sun Java v.1.5  
Shibboleth IdP v.1.3  
mod\_auth\_ldap

Configuration files relating to Shibboleth IdP implementation changed during set up.

```
/etc/httpd/conf/httpd.conf  
/etc/httpd/conf/Comodo/wildcard.kcl.ac.uk.key  
/etc/httpd/conf/Comodo/newrealwild.kcl.ac.uk.crt  
/etc/httpd/conf/Comodo/ca_new.crt  
/etc/httpd/conf.d/mod_jk.conf  
/etc/httpd/conf.d/auth_ldap.conf  
/etc/httpd/conf.d/ssl.conf  
/etc/httpd/conf.d/jk.conf  
/etc/httpd/conf/user.db  
/etc/ntp.conf  
/etc/sysconfig/iptables  
/usr/local/shibboleth-idp/etc/idp.xml  
/usr/local/shibboleth-idp/etc/resolver.ldap.xml  
/usr/local/shibboleth-idp/etc/shibboleap.xml  
/usr/local/shibboleth-idp/etc/arp.site.xml  
/usr/local/shibboleth-idp-1.3c-install/conf/log4j.properties  
/usr/local/tomcat/conf/server.xml  
/usr/local/tomcat/conf/workers.properties
```

## Modifications to Shibboleth IdP files

### shibboleap.xml

metadata file was provided by LSE to allow connection to their Service Provider. –  
Extract of KCL section:

```
<!-- This entity describes the main KCL IdP. -->
<IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol
urn:mace:shibboleth:1.0">
  <Extensions>
    <!-- This is a Shibboleth extension to express attribute scope rules. -->
    <shib:Scope xmlns:shib="urn:mace:shibboleth:metadata:1.0">kcl.ac.uk</shib:Scope>
  </Extensions>
  <KeyDescriptor use="signing">
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
MIIFKjCCBJOgAwIBAgI...eKtc0yIuQhkhvzjSuMEQa7pt/8JQRhoqHGQEOOs+z
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </KeyDescriptor>
  <ArtifactResolutionService index="1"
    Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
    Location="https://shibboleth.kcl.ac.uk/shibboleth-idp/Artifact"/>
    <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
    <SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
      Location="https://shibboleth.kcl.ac.uk/shibboleth-idp/SSO" />
  </IDPSSODescriptor>
  <AttributeAuthorityDescriptor
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol">
    <Extensions>
      <shib:Scope xmlns:shib="urn:mace:shibboleth:metadata:1.0">kcl.ac.uk</shib:Scope>
    </Extensions>
    <!-- The certificate has to be repeated here (or a different one specified if necessary).
    -->
    <KeyDescriptor use="signing">
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>
MIIFKjCCBJOgAwIBAgI...eKtc0yIuQhkhvzjSuMEQa7pt/8JQRhoqHGQEOOs+z
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <AttributeService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
      Location="https://shibboleth.kcl.ac.uk:8443/shibboleth-idp/AA"/>
    <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
    </AttributeAuthorityDescriptor>
    <Organization>
      <OrganizationName xml:lang="en">King's College London Identity
Provider</OrganizationName>
      <OrganizationDisplayName xml:lang="en">King's College
London</OrganizationDisplayName>
      <OrganizationURL xml:lang="en">http://www.kcl.ac.uk/</OrganizationURL>
    </Organization>
    <ContactPerson contactType="technical">
      <GivenName>Richard Warren</GivenName>
      <EmailAddress>richard.warren@kcl.ac.uk</EmailAddress>
    </ContactPerson>
  </EntityDescriptor>
```

## idp.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Shibboleth Identity Provider configuration -->
  <IdPConfig
    xmlns="urn:mace:shibboleth:idp:config:1.0"
    xmlns:cred="urn:mace:shibboleth:credentials:1.0"
    xmlns:name="urn:mace:shibboleth:namemapper:1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:mace:shibboleth:idp:config:1.0 ../schemas/shibboleth-
idpconfig-1.0.xsd"

    AAUrl="https://shibboleth.kcl.ac.uk:8443/shibboleth-idp/AA"

    resolverConfig="file:/usr/local/shibboleth-idp/etc/resolver.ldap.xml"

    defaultRelyingParty="https://lse.ac.uk/shibboleth/target/1"

    providerId="https://kcl.ac.uk/shibboleth/origin/1">
    <RelyingParty name="https://lse.ac.uk/shibboleth/target/1"
providerId="https://kcl.ac.uk/shibboleth/origin/1" signingCredential="inqueue_cred">
      <NameID nameMapping="shm" />
    </RelyingParty>

    <RelyingParty name="urn:mace:inqueue"
providerId="https://kcl.ac.uk/shibboleth/origin/1" signingCredential="inqueue_cred">
      <NameID nameMapping      ="shm" />
    </RelyingParty>

    <ReleasePolicyEngine>

    <ArpRepository
implementation="edu.internet2.middleware.shibboleth.aa.arp.provider.FileSystemArpRepos
itory">
      <Path>file:/usr/local/shibboleth-idp/etc/arps</Path>
    </ArpRepository>

    </ReleasePolicyEngine>

    <Logging>

    <ErrorLog level="WARN" location="file:/usr/local/shibboleth-idp/logs/shib-error.log"
/>

    <TransactionLog level="INFO" location="file:/usr/local/shibboleth-idp/logs/shib-
access.log" />

    </Logging>

    <NameMapping
    xmlns="urn:mace:shibboleth:namemapper:1.0"
    id="shm"
    format="urn:mace:shibboleth:1.0:nameIdentifier"
    type="SharedMemoryShibHandle"
    handleTTL="28800"/>

    <ArtifactMapper
implementation="edu.internet2.middleware.shibboleth.artifact.provider.MemoryArtifactMa
pper" />

    <Credentials xmlns="urn:mace:shibboleth:credentials:1.0">

<FileResolver Id="inqueue_cred">
  <Key>

  <Path>file:/etc/httpd/conf/Comodo/wildcard.kcl.ac.uk.key</Path>

  </Key>
  <Certificate>
  <Path>file:/etc/httpd/conf/Comodo/newrealwild.kcl.ac.uk.crt</Path>

  <CAPath>file:/etc/httpd/conf/Comodo</CAPath>

  </Certificate>

</FileResolver>

</Credentials>
```

```

    <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.ShibbolethV1SSOHandle
r">
    <Location>https?://[^\:\/]+(:443|80)?/shibboleth-idp/SSO</Location>
</ProtocolHandler>

    <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.SAMLv1_AttributeQuery
Handler">
    <Location>.+:8443/shibboleth-idp/AA</Location>
</ProtocolHandler>

    <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.SAMLv1_1ArtifactQuery
Handler">
    <Location>.+:8443/shibboleth-idp/Artifact</Location>
</ProtocolHandler>

    <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.Shibboleth_StatusHand
ler">
    <Location>https://[^\:\/]+(:443)?/shibboleth-idp/Status</Location>
</ProtocolHandler>

    <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadata"
    uri="file:/usr/local/shibboleth-idp/etc/shibboleap.xml"/>
</IdPConfig>

```

## resolver.Ldap.xml

```
<AttributeResolver xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mace:shibboleth:resolver:1.0"
xsi:schemaLocation="urn:mace:shibboleth:resolver:1.0 shibboleth-resolver-1.0.xsd">
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonEntitlement">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonAffiliation">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonNickname">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonPrimaryAffiliation">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonPrimaryOrgUnitDN">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonOrgUnitDN">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonOrgDN">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation" smartScope="kcl.ac.uk">
    <AttributeDependency requires="urn:mace:dir:attribute-def:eduPersonAffiliation" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonPrincipalName"
smartScope="kcl.ac.uk">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <!-- Persistent ID attribute based on cn number -->
  <PersistentIDAttributeDefinition id="urn:mace:dir:attribute-def:eduPersonTargetedID"
scope="kcl.ac.uk" sourceName="cn">
    <DataConnectorDependency requires="directory" />
    <Salt keyStorePath="file:///usr/local/shibboleth-idp/etc/persistent.jks"
keyStoreKeyAlias="handleKey" keyStorePassword="*****" keyStoreKeyPassword="*****" />
  </PersistentIDAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:cn">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:sn">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:mail">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <SimpleAttributeDefinition id="urn:mace:dir:attribute-def:givenName">
    <DataConnectorDependency requires="directory" />
  </SimpleAttributeDefinition>
  <JNDIDirectoryDataConnector id="directory">
```

```
<Search filter="uid=%PRINCIPAL%">
  <Controls searchScope="SUBTREE_SCOPE" returningObjects="false" />

</Search>
  <Property name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory" />
  <Property name="java.naming.provider.url" value="ldap://ldap-
test.kcl.ac.uk:1024/ou=people,dc=kcl,dc=ac,dc=uk" />
  </JNDIDirectoryDataConnector>
</AttributeResolver>
```

## arp.site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<AttributeReleasePolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mace:shibboleth:arp:1.0" xsi:schemaLocation="urn:mace:shibboleth:arp:1.0
shibboleth-arp-1.0.xsd" >

  <Description>KCL Site ARP.</Description>
  <Rule>

<Target>
  <AnyTarget />
</Target>

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

<Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName">
  <AnyValue release="permit"/>
</Attribute>

<Attribute name="urn:mace:dir:attribute-def:eduPersonTargetedID">
  <AnyValue release="permit"/>
</Attribute>

</Rule>
<Rule>
  <Target>

<Requester>https://lse.ac.uk/shibboleth/target/1</Requester>

</Target>
  <Attribute name="urn:mace:dir:attribute-def:cn">
  <AnyValue release="permit"/>
</Attribute>

  <Attribute name="urn:mace:dir:attribute-def:mail">
  <AnyValue release="permit"/>
</Attribute>

  <Attribute name="urn:mace:dir:attribute-def:sn">
  <AnyValue release="permit"/>
</Attribute>

  <Attribute name="urn:mace:dir:attribute-def:givenName">
  <AnyValue release="permit"/>
</Attribute>

</Rule>
</AttributeReleasePolicy>
```

## Modifications to Apache files:

### ssl.conf

```
Listen 443
Listen 8443
<VirtualHost default:443>
DocumentRoot "/var/www/html"
ServerName shibboleth.kcl.ac.uk:443

ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn

SSLEngine on
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
SSLCertificateKeyFile /etc/httpd/conf/Comodo/wildcard.kcl.ac.uk.key
SSLCertificateFile /etc/httpd/conf/Comodo/newrealwild.kcl.ac.uk.crt
SSLCACertificateFile /etc/httpd/conf/Comodo/ca_new.crt
<VirtualHost default:8443>
#   General setup for the virtual host
DocumentRoot "/var/www/html"
ServerName shibboleth.kcl.ac.uk:8443
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
#
# modify logleve to debug (from warn) - 21.11.05, rw
LogLevel warn

#

SSLEngine on
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
SSLVerifyClient optional_no_ca
SSLOptions +StdEnvVars +ExportCertData
#SSLProtocol all -SSLv2

SSLCertificateKeyFile /etc/httpd/conf/Comodo/wildcard.kcl.ac.uk.key
SSLCertificateFile /etc/httpd/conf/Comodo/newrealwild.kcl.ac.uk.crt
SSLCACertificateFile /etc/httpd/conf/Comodo/ca_new.crt
SSLCACertificatePath /etc/httpd/conf/Comodo
#

SSLVerifyDepth 10
#
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
</VirtualHost>
```

## auth\_ldap.conf

```
<IfModule mod_auth_ldap.c>

<Directory /var/www/html>
    AuthName "Enter KCL network username and password"
    AuthType Basic
    #SetHandler ldap-status
    #Order deny,allow
    #Deny from all
    #Allow from kcl.ac.uk

AuthLDAPEnabled on
    AuthLDAPURL ldap://ldap-test.kcl.ac.uk:1024/ou=people,dc=kcl,dc=ac,dc=uk?cn
        require valid-user

    #AuthLDAPURL
    ldap://ldap.kcl.ac.uk:389/ou=people,dc=kcl,dc=ac,dc=uk?uid?sub?(objectClass=*)
        #AuthLDAPAuthoritative on
        #require valid-user
</Directory>

<Location /shibboleth-idp/SSO>

AuthName "Enter KCL username and password"
    AuthType Basic
    AuthLDAPEnabled on
    AuthLDAPURL ldap://ldap-test.kcl.ac.uk:1024/ou=people,dc=kcl,dc=ac,dc=uk?uid
        require valid-user
</Location>

<Location /jsp-examples/*>
    AuthName "ldap"
    AuthType Basic
    AuthLDAPEnabled on
    AuthLDAPURL ldap://ldap-test.kcl.ac.uk:1024/ou=people,dc=kcl,dc=ac,dc=uk?cn
        require valid-user
</Location>
</IfModule>
```

## Modifications to Tomcat configuration files

### server.xml

```
<!-- Define an AJP 1.3 Connector on port 8009 -->  
  
<Connector port="8009" debug="1"  
    enableLookups="false" tomcatAuthentication="false" redirectPort="8443"  
    protocol="AJP/1.3" />
```

### jk.conf (Called by httpd.conf)

```
<IfModule !mod_jk.c>  
    LoadModule jk_module modules/mod_jk.so  
</IfModule>  
JkWorkersFile "/etc/httpd/conf.d/jk/workers.properties"  
JkLogFile "/etc/httpd/logs/mod_jk.log"  
  
JkLogLevel info  
JkMount /shibboleth-idp/* ajp13_worker  
JkMount /jsp-examples/* ajp13_worker
```