Archivos de configuración

A continuación aparecen los listados de los archivos de configuración más importantes del Proveedor de Servicios y de Identidad.

shib\src\conf\dist.idp.xml

```
<?xml version="1.1" 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://idp.example.org:8443/shibboleth-idp/AA"
 resolverConfig="$SHIB_HOME$/etc/resolver.xml"
 defaultRelyingParty="urn:mace:shibboleth:examples"
 providerId="https://idp.example.org/shibboleth">


 <!-- This section contains configuration options that apply only to a
site or group of sites.  This would normally be adjusted when a new
federation or bilateral trust relationship is established -->
 <RelyingParty name="urn:mace:shibboleth:examples"
signingCredential="example_cred"> <!-- (signingCredential) must
correspond to a <Credential/> element below -->
   <NameID nameMapping="shm"/> <!-- (nameMapping) must correspond to a
<NameMapping/> element below -->
 </RelyingParty>

 <!-- InQueue example (the schemaHack is needed for 1.1/1.2 SPs)-->
 <!--
 <RelyingParty name="urn:mace:inqueue"
signingCredential="inqueue_cred"
    schemaHack="true">
  <NameID nameMapping="shm"/>
 </RelyingParty> -->


 <!-- Configuration for the attribute release policy engine
  For most configurations this won't need adjustment -->
 <ReleasePolicyEngine>
  <ArpRepository
implementation="edu.internet2.middleware.shibboleth.aa.arp.provider.Fi
leSystemArpRepository">
    <Path>$SHIB_HOME$/etc/arps/</Path>
  </ArpRepository>
 </ReleasePolicyEngine>


    <!-- Logging Configuration
```

```
  The defaults work fine in this section, but it is sometimes helpful
to use "DEBUG" as the level for the <ErrorLog/> when trying to
diagnose problems -->
 <Logging>
  <ErrorLog level="WARN" location="$SHIB_HOME$/logs/shib-error.log" />
  <!--
  <TransactionLog level="INFO" location="$SHIB_HOME$/logs/shib-
access.log" />
  -->
  <!-- para PFC -->
  <TransactionLog level="DEBUG" location="$SHIB_HOME$/logs/shib-
access.log" />
 </Logging>
 <!-- Uncomment the configuration section below and comment out the
one above if you would like to manually configure log4j -->
    <!--
 <Logging>
  <Log4JConfig location="file:///tmp/log4j.properties" />
 </Logging> -->


 <!-- This configuration section determines how Shibboleth maps
between SAML Subjects and local principals. The default mapping uses
shibboleth handles, but other formats can be added. The mappings
listed here are only active when they are referenced within a
<RelyingParty/> element above -->
 <NameMapping
  xmlns="urn:mace:shibboleth:namemapper:1.0"
  id="shm"
  format="urn:mace:shibboleth:1.0:nameIdentifier"
  type="SharedMemoryShibHandle"
  handleTTL="1800"/>


 <!-- Determines how SAML artifacts are stored and retrieved
  The (sourceLocation) attribute must be specified when using type 2
artifacts -->
 <ArtifactMapper
implementation="edu.internet2.middleware.shibboleth.artifact.provider.
MemoryArtifactMapper" />


 <!-- This configuration section determines the keys/certs to be used
when signing SAML assertions -->
 <!-- The credentials listed here are used when referenced within
<RelyingParty/> elements above -->
 <Credentials xmlns="urn:mace:shibboleth:credentials:1.0">
  <FileResolver Id="example_cred">
   <Key>
    <Path>$SHIB_HOME$/etc/idp-example.key</Path>
   </Key>
   <Certificate>
    <Path>$SHIB_HOME$/etc/idp-example.crt</Path>
   </Certificate>
  </FileResolver>

  <!-- InQueue example (Deployments would need to generate an InQueue-
compatible certificate) -->
  <!--
  <FileResolver Id="inqueue_cred">
   <Key>
```

```
     <Path>$SHIB_HOME$/etc/idp-inqueue.key</Path>
    </Key>
    <Certificate>
     <Path>$SHIB_HOME$/etc/idp-inqueue.crt</Path>
    </Certificate>
   </FileResolver>
    -->
 </Credentials>


 <!-- Protocol handlers specify what type of requests the IdP can
respond to.  The default set listed here should work for most
configurations.  Modifications to this section may require
modifications to the deployment descriptor -->
 <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.Shibb
olethV1SSOHandler">

 <Location>https?://[^:/]+(:(443|80))?/$IDP_WEBAPP_NAME$/SSO</Location
> <!-- regex works when using default protocol ports -->
 </ProtocolHandler>
 <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.SAMLv
1_AttributeQueryHandler">
   <Location>.+:8443/$IDP_WEBAPP_NAME$/AA</Location>
 </ProtocolHandler>
 <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.SAMLv
1_1ArtifactQueryHandler">
   <Location>.+:8443/$IDP_WEBAPP_NAME$/Artifact</Location>
 </ProtocolHandler>


 <!-- This section configures the loading of SAML2 metadata, which
contains information about system entities and how to authenticate
them.  The metadatatool utility can be used to keep federation
metadata files in synch.
   Metadata can also be placed directly within this these elements. -->
 <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadat
a"
                uri="$SHIB_HOME$/etc/example-metadata.xml"/>


 <!-- InQueue example (Deployments would need to get updated InQueue
metadata) -->
 <!--
 <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadat
a"
                uri="$SHIB_HOME$/etc/IQ-metadata.xml"/> -->
</IdPConfig>
```

shib\src\conf\dist.sp.xml

```
<?xml version="1.1" encoding="ISO-8859-1"?>

<!-- Sample configuration file for the Java SP. It shares syntax with
the C++ SP, but some elements used only by C++ have been removed here.
[Note: at this time no all elements of this configuration file are
supported.]
  -->

<SPConfig xmlns="urn:mace:shibboleth:target:config:1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mace:shibboleth:target:config:1.0
../schemas/shibboleth-targetconfig-1.0.xsd"
 clockSkew="180">

 <!-- The Global section pertains to shared Shibboleth processes like
the shibd daemon. -->
 <Global logger="$SHIB_HOME$/etc/shibd.logger">

  <!-- A listener (TCP or Unix) is required by the syntax
   of the configuration file, but is not used by Java.
   At some point in the future there may be an RMI listener. -->
  <UnixListener address="bogus"/>

  <!--
  See deploy guide for details, but:
   cacheTimeout - how long before expired sessions are purged from the
cache
   AATimeout - how long to wait for an AA to respond
   AAConnectTimeout - how long to wait while connecting to an AA
   defaultLifetime - if attributes come back without guidance, how
long should they last?
   strictValidity - if we have expired attrs, and can't get new ones,
keep using them?
   propagateErrors - suppress errors while getting attrs or let user
see them?
   retryInterval - if propagateErrors is false and query fails, how
long to wait before trying again
  Only one session cache can be defined.
  -->
  <MemorySessionCache
   cleanupInterval="300"
   cacheTimeout="3600"
   AATimeout="30"
   AAConnectTimeout="15"
   defaultLifetime="1800"
   retryInterval="300"
   strictValidity="false"
   propagateErrors="false"
   />
  <!--
  <MySQLSessionCache cleanupInterval="300" cacheTimeout="3600"
AATimeout="30" AAConnectTimeout="15"
   defaultLifetime="1800" retryInterval="300" strictValidity="false"
propagateErrors="false"
   mysqlTimeout="14400" storeAttributes="false">
   <Argument>&#x2D;&#x2D;language=@-PREFIX-@/share/english</Argument>
   <Argument>&#x2D;&#x2D;datadir=@-PREFIX-@/data</Argument>
```

```
   </MySQLSessionCache>
   -->

   <!-- Default replay cache is in-memory. -->
   <!--
   <MySQLReplayCache>
    <Argument>&#x2D;&#x2D;language=@-PREFIX-@/share/english</Argument>
    <Argument>&#x2D;&#x2D;datadir=@-PREFIX-@/data</Argument>
   </MySQLReplayCache>
   -->
 </Global>

 <!-- The Local section pertains to resource-serving processes (often
process pools) like web servers. -->
 <Local localRelayState="true">
   <!--
   To customize behavior, map hostnames and path components to
applicationId and other settings.

   The RequestMapProvider specified here is authoritative when it
assigns an appliationId to resource directories under the control of
this SP. However, the information here about when to require
authentication is advistory, and may be overridden by the
configuration of the ResourceManager. In particular, the Servlet
Filter has initialization parameters in its web.xml that will override
what is configured here about requireSession.
   -->
   <RequestMapProvider
type="edu.internet2.middleware.shibboleth.sp.provider.NativeRequestMap
Provider">
    <RequestMap applicationId="default">
     <Host name="sp.example.org">
      <!-- Nominally require shibboleth authentication for all
documents under /secure. Note that the sample /secure application
distributed with the Filter overrides this to specify only specific
file names/types. -->
      <Path name="secure" authType="shibboleth" requireSession="true"
exportAssertion="true">
      </Path>
     </Host>
    </RequestMap>
   </RequestMapProvider>

 </Local>

 <!--
 The Applications section is where most of Shibboleth's SAML bits are
defined. Resource requests are mapped in the Local section into an
applicationId that points into to this section.
 -->
 <Applications id="default"
  providerId="https://sp.example.org/shibboleth"
  homeURL="https://sp.example.org/index.html"
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
  xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">

   <!--
   Controls session lifetimes, address checks, cookie handling, and the
protocol handlers. You MUST supply an effectively unique handlerURL
value for each of your applications.  The value can be a relative
path, a URL with no hostname (https:///path) or a full URL.  The system
```

```
can compute a relative value based on the virtual host. Using
handlerSSL="true" will force the protocol to be https. You should also
add a cookieProps setting of "; secure" in that case. Note that while
we default checkAddress to "false", this has a negative  impact on the
security of the SP. Stealing cookies/sessions is much easier with this
disabled.
  -->
  <Sessions lifetime="7200" timeout="3600" checkAddress="false"
   handlerURL="/Shibboleth.sso" handlerSSL="false" idpHistory="true"
idpHistoryDays="7">

    <!--
    SessionInitiators handle session requests and relay them to a WAYF
or directly to an IdP, if possible. Automatic session setup will use
the default or first element (or requestSessionWith can specify a
specific id to use). Lazy sessions can be started with any initiator.
The only Binding supported is the
"urn:mace:shibboleth:sp:1.3:SessionInit" lazy session profile.
    -->

    <!-- This default example directs users to a specific IdP's SSO
service. -->
    <SessionInitiator isDefault="true" id="example"
Location="/WAYF/idp.example.org"
     Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
     wayfURL="https://idp.example.org:443/shibboleth-idp/SSO"
     wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>

    <!-- This example directs users to a specific federation's WAYF
service. -->
    <SessionInitiator id="IQ" Location="/WAYF/InQueue"
     Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
     wayfURL="https://wayf.internet2.edu/InQueue/WAYF"
     wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>

    <!--
    md:AssertionConsumerService elements replace the old shireURL
function with an explicit handler for particular profiles, such as
SAML 1.1 POST or Artifact. The isDefault and index attributes are used
when sessions are initiated to determine how to tell the IdP where and
how to return the response.
    -->
    <md:AssertionConsumerService Location="/SAML/POST" isDefault="true"
index="1"
     Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
    <md:AssertionConsumerService Location="/SAML/Artifact" index="2"
     Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>

    <!--
    md:SingleLogoutService elements are mostly a placeholder for 2.0,
but a simple cookie-clearing option with a ResponseLocation or a
return URL parameter is supported via the
"urn:mace:shibboleth:sp:1.3:Logout" Binding value.
    -->
    <md:SingleLogoutService Location="/Logout"
Binding="urn:mace:shibboleth:sp:1.3:Logout"/>

  </Sessions>

  <!--
```

```
  You should customize these pages! You can add attributes with values
that can be plugged into your templates. You can remove the access
attribute to cause the module to return a standard 403 Forbidden error
code if authorization fails, and then customize that condition using
your web server.
  -->
  <Errors session="$SHIB_HOME$/etc/sessionError.html"
   metadata="$SHIB_HOME$/etc/metadataError.html"
   rm="$SHIB_HOME$/etc/rmError.html"
   access="$SHIB_HOME$/etc/accessError.html"
   supportContact="root@localhost"
   logoLocation="/shibtarget/logo.jpg"
   styleSheet="/shibtarget/main.css"/>

  <!-- Indicates what credentials to use when communicating -->
  <CredentialUse TLS="defcreds" Signing="defcreds">
   <!-- RelyingParty elements can customize credentials for specific
IdPs/sets. -->
   <!--
   <RelyingParty Name="urn:mace:inqueue" TLS="inqueuecreds"
Signing="inqueuecreds"/>
   -->
  </CredentialUse>

  <!-- Use designators to request specific attributes or none to ask
for all -->
  <!--
  <saml:AttributeDesignator AttributeName="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation"

 AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
  <saml:AttributeDesignator AttributeName="urn:mace:dir:attribute-
def:eduPersonTargetedID"

 AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
  -->

  <!-- AAP can be inline or in a separate file -->
  <AAPProvider
type="edu.internet2.middleware.shibboleth.aap.provider.XMLAAP"
uri="$SHIB_HOME$/etc/AAP.xml"/>

  <!-- Operational config consists of metadata and trust providers.
Can be external or inline. -->

  <!-- Dummy metadata for private testing, delete for production
deployments. -->
  <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadat
a"
   uri="$SHIB_HOME$/etc/example-metadata.xml"/>

  <!-- InQueue pilot federation, delete for production deployments. --
>
  <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadat
a"
   uri="$SHIB_HOME$/etc/IQ-metadata.xml"/>

  <!-- The standard trust provider supports SAMLv2 metadata with path
validation extensions. -->
```

```
   <TrustProvider
type="edu.internet2.middleware.shibboleth.common.provider.ShibbolethTr
ust"/>

   <!--
   Zero or more SAML Audience condition matches (mainly for Shib 1.1
compatibility). If you get "policy mismatch errors, you probably need
to supply metadata about your SP to the IdP if it's running 1.2.
Adding an element here is only a partial fix.
   -->
   <saml:Audience>urn:mace:inqueue</saml:Audience>

   <!--
   You can customize behavior of specific applications here. The
default elements inside the outer <Applications> element generally
have to be overridden in an all or nothing fashion. That is, if you
supply a <Sessions> or <Errors> override, you MUST include all
attributes you want to apply, as they will not be inherited.
Similarly, if you specify an element such as <MetadataProvider>, it is
not additive with the defaults, but replaces them.

   Note that each application must have a handlerURL that maps uniquely
to it and no other application in the <RequestMap>. Otherwise no
sessions will reach the application. If each application lives on its
own vhost, then a single handler at "/Shibboleth.sso" is sufficient,
since the hostname will distinguish the application.

   The example below shows a special application that requires use of
SSL when establishing sessions, restricts the session cookie to SSL
and a specific folder, and inherits most other behavior except that it
requests only EPPN from the origin instead of asking for all
attributes.
   Note that it will inherit all of the handler endpoints defined for
the default application but will append them to the handlerURL defined
here.
   -->
   <!--
   <Application id="foo-admin">
    <Sessions lifetime="7200" timeout="3600" checkAddress="true"
     handlerURL="/secure/admin/Shibboleth.sso" handlerSSL="true"
     cookieProps="; path=/secure/admin; secure"/>
    <saml:AttributeDesignator AttributeName="urn:mace:dir:attribute-
def:eduPersonPrincipalName"

 AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
   </Application>
   -->

 </Applications>

 <!-- Define all the private keys and certificates here that you
reference from <CredentialUse>. -->
 <CredentialsProvider
type="edu.internet2.middleware.shibboleth.common.Credentials">
   <Credentials xmlns="urn:mace:shibboleth:credentials:1.0">
    <FileResolver Id="defcreds">
     <Key format="PEM">
      <Path>$SHIB_HOME$/etc/sp-example.key</Path>
     </Key>
     <Certificate format="PEM">
      <Path>$SHIB_HOME$/etc/sp-example.crt</Path>
```

```
     </Certificate>
    </FileResolver>

    <!--
    Mostly you can define a single keypair above, but you can define
and name a second keypair to be used only in specific cases and then
specify when to use it inside a  <CredentialUse> element.
    -->
    <!--
    <FileResolver Id="inqueuecreds">
     <Key format="PEM" password="handsoff">
      <Path>$SHIB_HOME$/etc/inqueue.key</Path>
     </Key>
     <Certificate format="PEM">
      <Path>$SHIB_HOME$/etc/inqueue.crt</Path>
     </Certificate>
    </FileResolver>
    -->
   </Credentials>
 </CredentialsProvider>

 <!-- Specialized attribute handling for cases with complex syntax. --
>
 <AttributeFactory AttributeName="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"

 type="edu.internet2.middleware.shibboleth.common.provider.TargetedIDF
actory"/>

</SPConfig>
```

shib\src\arps\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>Simplest possible ARP.</Description>
 <Rule>
  <Target>
   <AnyTarget/>
  </Target>
  <Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation">
   <AnyValue release="permit"/>
  </Attribute>
  <!-- PFC: Eliminamos la entrega del siguiente atributo,
   al no aparecer la definición correspondiente en
   resolver.xml -->
  <!--
  <Attribute name="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation">
   <AnyValue release="permit"/>
  </Attribute>
  -->
 </Rule>
</AttributeReleasePolicy>
```

shib\src\resolver.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="echo"/>
 </SimpleAttributeDefinition>

 <SimpleAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonAffiliation">
   <DataConnectorDependency requires="echo"/>
 </SimpleAttributeDefinition>


 <!-- To use these attributes, you should change the smartScope value
to match your site's domain name. -->
 <!--
 <SimpleAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation" smartScope="shibdev.edu">
   <AttributeDependency requires="urn:mace:dir:attribute-
def:eduPersonAffiliation"/>
 </SimpleAttributeDefinition>

 <SimpleAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonPrincipalName" smartScope="shibdev.edu">
     <DataConnectorDependency requires="echo"/>
 </SimpleAttributeDefinition>
 -->


 <!-- Example persistent id attribute.  Since this configuration is
permanent, some thought is required before deploying in production.
Consider replacing this with a database-backed mechanism of some sort.
-->
 <!--
 <SAML2PersistentID id="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
sourceName="guid">
   <DataConnectorDependency requires="echo"/>
   <Salt keyStorePath="/conf/persistent.jks"
keyStoreKeyAlias="handleKey" keyStorePassword="shibhs"
keyStoreKeyPassword="shibhs"/>
 </SAML2PersistentID>
 -->
 <!--  Deprecated persistent id example, use only with SPs that are
already relying on your values. -->
 <!--
 <PersistentIDAttributeDefinition id="urn:mace:dir:attribute-
def:eduPersonTargetedID" scope="shibdev.edu" sourceName="guid">
   <DataConnectorDependency requires="echo"/>
   <Salt keyStorePath="/conf/persistent.jks"
keyStoreKeyAlias="handleKey" keyStorePassword="shibhs"
keyStoreKeyPassword="shibhs"/>
 </PersistentIDAttributeDefinition>
 -->
```

```
 <CustomDataConnector id="echo"
class="edu.internet2.middleware.shibboleth.aa.attrresolv.provider.Samp
leConnector"/>

</AttributeResolver>
```

shib\src\AAP.xml

```
<AttributeAcceptancePolicy xmlns="urn:mace:shibboleth:1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:mace:shibboleth:1.0
../schemas/shibboleth.xsd">

 <!--
 An AAP is a set of AttributeRule elements, each one
 referencing a specific attribute by URI. All attributes that
 should be visible to an application running at the target should
 be listed, or they will be filtered out.

 The Header and Alias attributes map an attribute to an HTTP header
and to an htaccess rule name respectively. Without Header, the
attribute will only be obtainable from the exported SAML assertion in
raw XML.

 Scoped attributes are also filtered on Scope via the Domain elements
in the site metadata.
 -->

 <!-- First some useful eduPerson attributes that many sites might
use. -->

 <AttributeRule Name="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation" Scoped="true" CaseSensitive="false"
Header="Shib-EP-Affiliation" Alias="affiliation">
   <!-- Filtering rule to limit values to eduPerson-defined
enumeration. -->
        <AnySite>
            <Value>MEMBER</Value>
            <Value>FACULTY</Value>
            <Value>STUDENT</Value>
            <Value>STAFF</Value>
            <Value>ALUM</Value>
            <Value>AFFILIATE</Value>
            <Value>EMPLOYEE</Value>
        </AnySite>

        <!-- Example of Scope rule to override site metadata. -->
        <SiteRule Name="urn:mace:inqueue:shibdev.edu">
         <Scope Accept="false">shibdev.edu</Scope>
         <Scope Type="regexp">^.+\.shibdev\.edu$</Scope>
        </SiteRule>
 </AttributeRule>

 <!--
 This attribute is provided mostly to ease testing because an IdP out
of the box only sends the unscoped version. It has little use because
it lacks the context needed to work in a multi-domain scenario and is
a subset of the scoped version anyway.
  -->
 <AttributeRule Name="urn:mace:dir:attribute-def:eduPersonAffiliation"
CaseSensitive="false" Header="Shib-EP-UnscopedAffiliation"
Alias="unscoped-affiliation">
        <AnySite>
            <Value>MEMBER</Value>
            <Value>FACULTY</Value>
            <Value>STUDENT</Value>
            <Value>STAFF</Value>
```

```
            <Value>ALUM</Value>
            <Value>AFFILIATE</Value>
            <Value>EMPLOYEE</Value>
        </AnySite>
 </AttributeRule>

    <AttributeRule Name="urn:mace:dir:attribute-
def:eduPersonPrincipalName" Scoped="true" Header="REMOTE_USER"
Alias="user">
   <!-- Basic rule to pass through any value. -->
        <AnySite>
            <Value Type="regexp">^[^@]+$</Value>
        </AnySite>
    </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:eduPersonEntitlement"
Header="Shib-EP-Entitlement" Alias="entitlement">
   <!-- Entitlements tend to be filtered per-site. -->

   <!--
   Optional site rule that applies to any site
   <AnySite>
    <Value>urn:mace:example.edu:exampleEntitlement</Value>
   </AnySite>
   -->

   <!-- Specific rules for an origin site, these are just
development/sample sites. -->
   <SiteRule Name="urn:mace:inqueue:example.edu">
    <Value Type="regexp">^urn:mace:.+$</Value>
   </SiteRule>
   <SiteRule Name="urn:mace:inqueue:shibdev.edu">
    <Value Type="regexp">^urn:mace:.+$</Value>
   </SiteRule>
 </AttributeRule>

 <!-- A persistent id attribute that supports personalized anonymous
access. -->

 <!-- First, the deprecated version: -->
 <AttributeRule Name="urn:mace:dir:attribute-def:eduPersonTargetedID"
Scoped="true" Header="Shib-TargetedID" Alias="targeted_id">
        <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <!-- Second, the new version: -->
 <AttributeRule Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" Header="Shib-
TargetedID" Alias="targeted_id">
        <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <!-- Some more eduPerson attributes, uncomment these to use them... -
->
 <!--

 <AttributeRule Name="urn:mace:dir:attribute-def:eduPersonNickname">
        <AnySite>
```

```
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-
def:eduPersonPrimaryAffiliation" CaseSensitive="false" Header="Shib-
EP-PrimaryAffiliation">
        <AnySite>
            <Value>MEMBER</Value>
            <Value>FACULTY</Value>
            <Value>STUDENT</Value>
            <Value>STAFF</Value>
            <Value>ALUM</Value>
            <Value>AFFILIATE</Value>
            <Value>EMPLOYEE</Value>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-
def:eduPersonPrimaryOrgUnitDN" Header="Shib-EP-PrimaryOrgUnitDN">
        <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:eduPersonOrgUnitDN"
Header="Shib-EP-OrgUnitDN">
        <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:eduPersonOrgDN"
Header="Shib-EP-OrgDN">
        <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 -->


 <!--Examples of common LDAP-based attributes, uncomment to use
these... -->
 <!--

 <AttributeRule Name="urn:mace:dir:attribute-def:cn" Header="Shib-
Person-commonName">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:sn" Header="Shib-
Person-surname">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>
```

```
 <AttributeRule Name="urn:mace:dir:attribute-def:telephoneNumber"
Header="Shib-Person-telephoneNumber">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:title" Header="Shib-
OrgPerson-title">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:initials"
Header="Shib-InetOrgPerson-initials">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:description"
Header="Shib-Person-description">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:carLicense"
Header="Shib-InetOrgPerson-carLicense">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:departmentNumber"
Header="Shib-InetOrgPerson-deptNum">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:displayName"
Header="Shib-InetOrgPerson-displayName">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:employeeNumber"
Header="Shib-InetOrgPerson-employeeNum">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:employeeType"
Header="Shib-InetOrgPerson-employeeType">
   <AnySite>
            <AnyValue/>
        </AnySite>
```

```
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:preferredLanguage"
Header="Shib-InetOrgPerson-prefLang">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:manager"
Header="Shib-InetOrgPerson-manager">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:roomNumber"
Header="Shib-InetOrgPerson-roomNum">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:seeAlso"
Header="Shib-OrgPerson-seeAlso">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-
def:facsimileTelephoneNumber" Header="Shib-OrgPerson-fax">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:street" Header="Shib-
OrgPerson-street">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:postOfficeBox"
Header="Shib-OrgPerson-POBox">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:postalCode"
Header="Shib-OrgPerson-postalCode">
   <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:st" Header="Shib-
OrgPerson-state">
   <AnySite>
```

```
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:givenName"
Header="Shib-InetOrgPerson-givenName">
  <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:l" Header="Shib-
OrgPerson-locality">
  <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:businessCategory"
Header="Shib-InetOrgPerson-businessCat">
  <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-def:ou" Header="Shib-
OrgPerson-orgUnit">
  <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 <AttributeRule Name="urn:mace:dir:attribute-
def:physicalDeliveryOfficeName" Header="Shib-OrgPerson-OfficeName">
  <AnySite>
            <AnyValue/>
        </AnySite>
 </AttributeRule>

 -->

</AttributeAcceptancePolicy>
```