Magic

Middleware for collaborative Applications
and Global vIrtual Communities

On line Training Material on AAI development

Service Provider Concepts, Implementation
and Interfederations

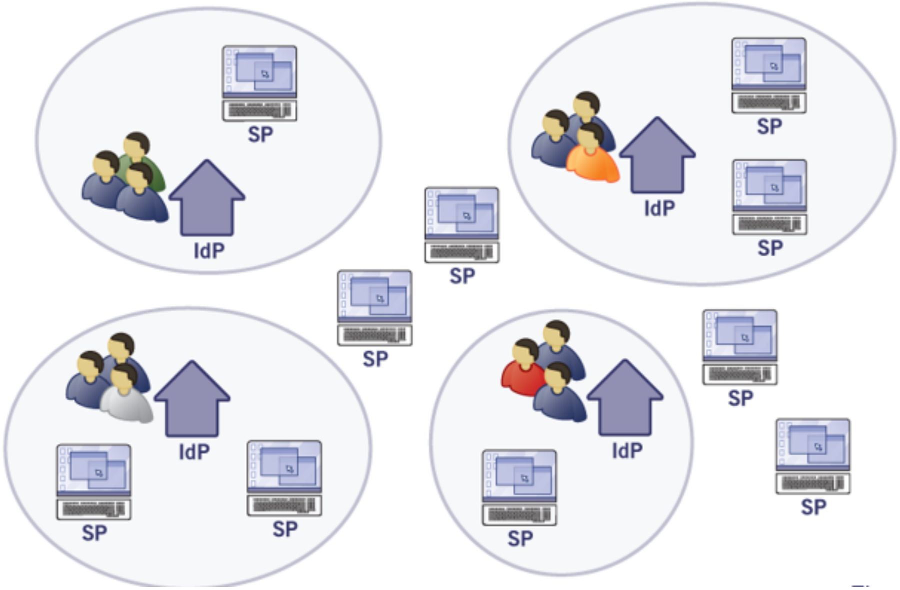
Francisco Leonardo Mota| RNP | 27-02-16 | WP2

European Commission

Service Providers (SP)

A Service Provider enables web applications written with any programming language or framework integrating natively with a web server.

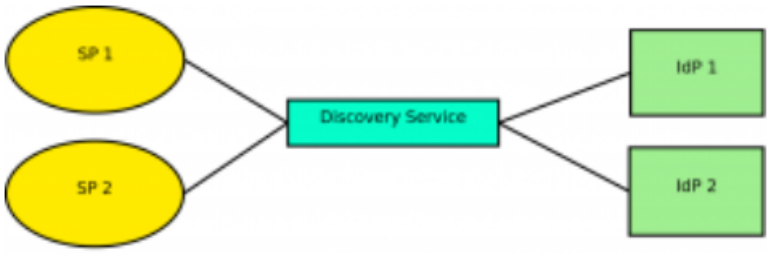
In a federation, it relies on a trusted Identity Provider (IdP) for authentication and authorization.

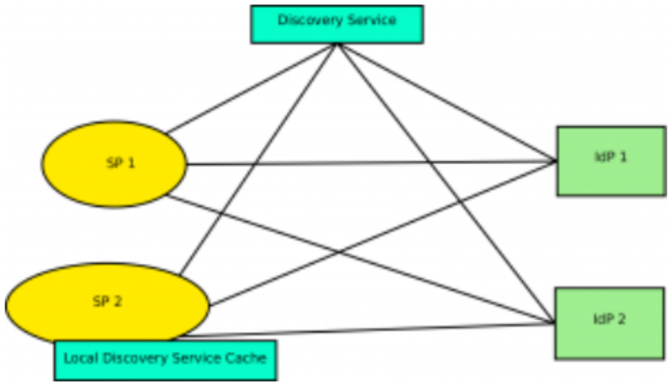


Service Providers (SP)

The SP in most situations only receive the minimum of information that is required to authorize the user in question, even in the most diverse possible architectures.

The modern developments allow users to access non web-based services via federation.

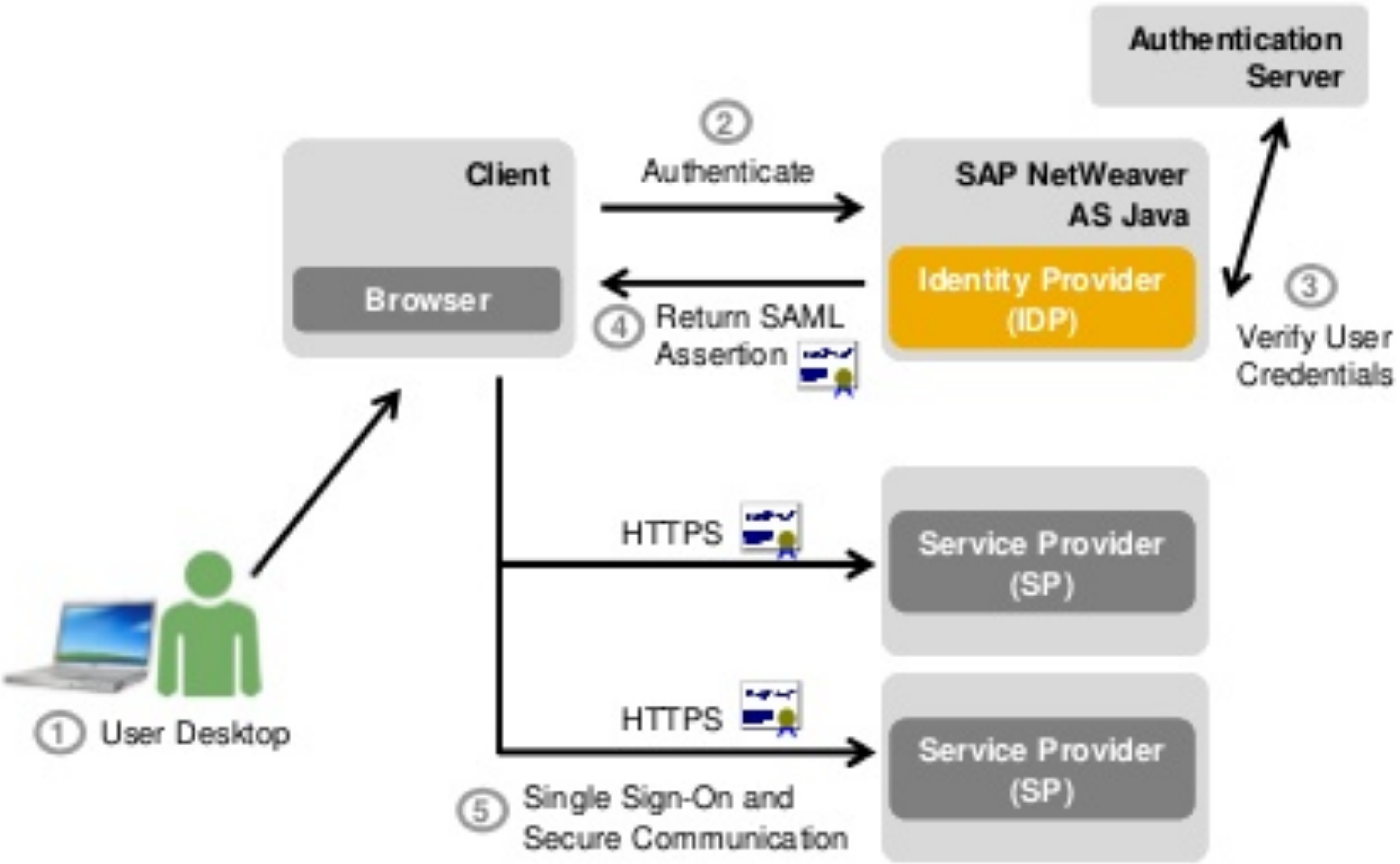




Service Providers - How does it Works?

- It intercepts access to a protected resource or application entry point;
- Discovers the user's choice of Identity Provider;
- Issue an authentication request to the selected Identity Provider;
- Process the authentication responses and extract the user information;
- Apply local policies and gather additional data;
- Pass identity information to application.

Service Providers - How does it Works?



SAML Protocol

The *Security Assertion Markup Language* – SAML – is a XML standard developed by Organization for the Advancement of Structured Information Standards (OASIS) that allows service providers and identity providers exchange authentication and autorization information.

SAML has the following components:

- Assertions: is a package of information that supplies one or more statements made by a SAML authority
- Protocols: elements of requests / responses packaging the assertions
- Bindings: define how messages SAML protocol are used in transport protocols
- Profiles: How to combine the bindings, protocols and assertions SAML for specific use cases

Motivation / Benefits

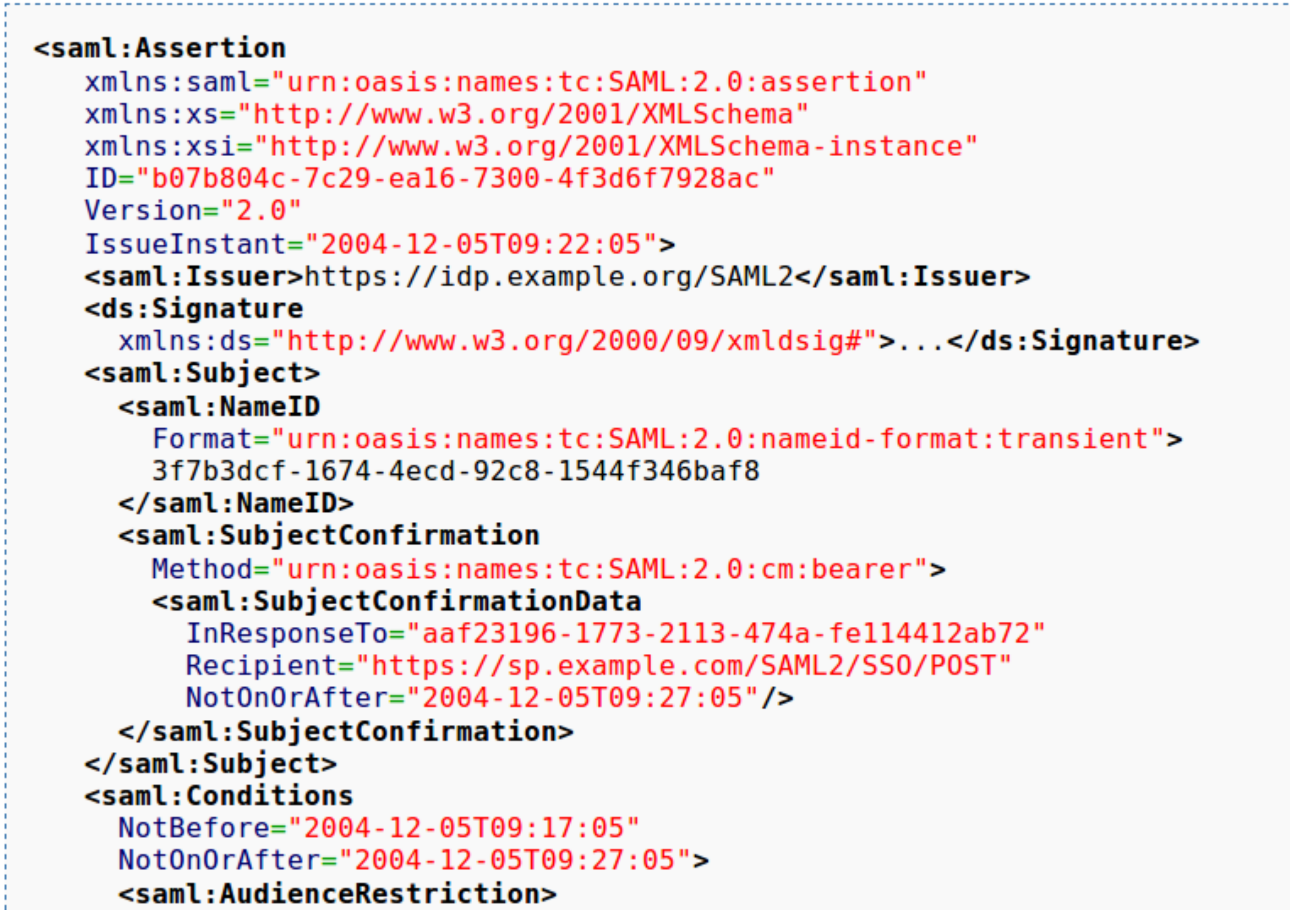
- Platform neutrality
- Loose coupling of directories
- Best online experience for end users
- Administrative cost reduction for the SP
- Transfer risk

SAML Assertions

- An assertion basically encodes the following informations:

- An assertion ("b07b804c-7c29-EA16-7300-4f3d6f7928ac") was issued at "2004-12-05T09: 22:05 Z" by the identity provider (https://idp.example.org/SAML2) about (3f7b3dcf -1674-4ecd-92c8-1544f346baf8) exclusively for the service provider (https://sp.example.com/SAML2)

SAML Assertions



```
<saml:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   ID="b07b804c-7c29-ea16-7300-4f3d6f7928ac"
   Version="2.0"
   IssueInstant="2004-12-05T09:22:05">
   <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
   <ds:Signature
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
   <saml:Subject>
     <saml:NameID
       Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
       3f7b3dcf-1674-4ecd-92c8-1544f346baf8
     </saml:NameID>
     <saml:SubjectConfirmation
       Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
       <saml:SubjectConfirmationData
         InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
         Recipient="https://sp.example.com/SAML2/SSO/POST"
         NotOnOrAfter="2004-12-05T09:27:05"/>
     </saml:SubjectConfirmation>
   </saml:Subject>
   <saml:Conditions
     NotBefore="2004-12-05T09:17:05"
     NotOnOrAfter="2004-12-05T09:27:05">
     <saml:AudienceRestriction>
```

SAML Protocols

- Authentication Request Protocol
  - SAML 2.0 in the flow begins in the service provider, sending a request for authentication to the identity provider
  - A <samlp: AuthnRequest> is transmitted to the identity provider by SP
  - The identity provider authenticates the user (if necessary) and sends an authentication response, which is transmitted back to the SP (through the browser)

SAML Protocols

- Artifact Resolution Protocols
  - The SAML message is transmitted by value or by reference. A reference to a SAML message is called artifact.
  - The artifact receiver resolves the reference by sending a request <samlp: ArtifactResolve> directly to the sender's device, which then responds with the message referenced by real artifact.

SAML Bindings

HTTP Redirect Binding and HTTP POST Binding

- For Web Browser SSO, the HTTP Redirect Binding and the Binding HTTP POST are commonly used.
- SP can use “HTTP Redirect” to send a request, while identity provider uses “HTTP POST” to transmit a reply.

SAML Bindings

- HTTP Redirect Binding and HTTP POST Binding
  - SAML protocol messages are often sent directly to the URL of an HTTP GET request.
  - The "HTTP Redirect" is suitable for short messages, such as <samlp: AuthnRequest> since the size of a URL is limited
  - The most beautiful messages (eg. Those that contain SAML assertions signatures) shall be transmitted through other bindings such as "HTTP POST"

SAML Bindings

- *HTTP POST Binding*
  - Example of a XHTML form submitted by SP via “HTTP POST Binding”

```
<form method="post" action="https://idp.example.org/SAML2/SSO/POST" ...>
  <input type="hidden" name="SAMLRequest" value="request" />
        ... other input parameter....
  <input type="submit" value="Submit" />
</form>
```

SAML Bindings

- *HTTP Redirect Binding*
  - *SAML requests or responses transmitted via HTTP Redirect have a SAMLRequest or SAMLResponse query string parameter, respectively. Before it's sent, the message is deflated (sans header and checksum), base64-encoded, and URL-encoded, in that order. Upon receipt, the process is reversed to recover the original message*

SAML Bindings

- HTTP Redirect Binding
  - For example, encoding the <samlp:AuthnRequest> message above yields:

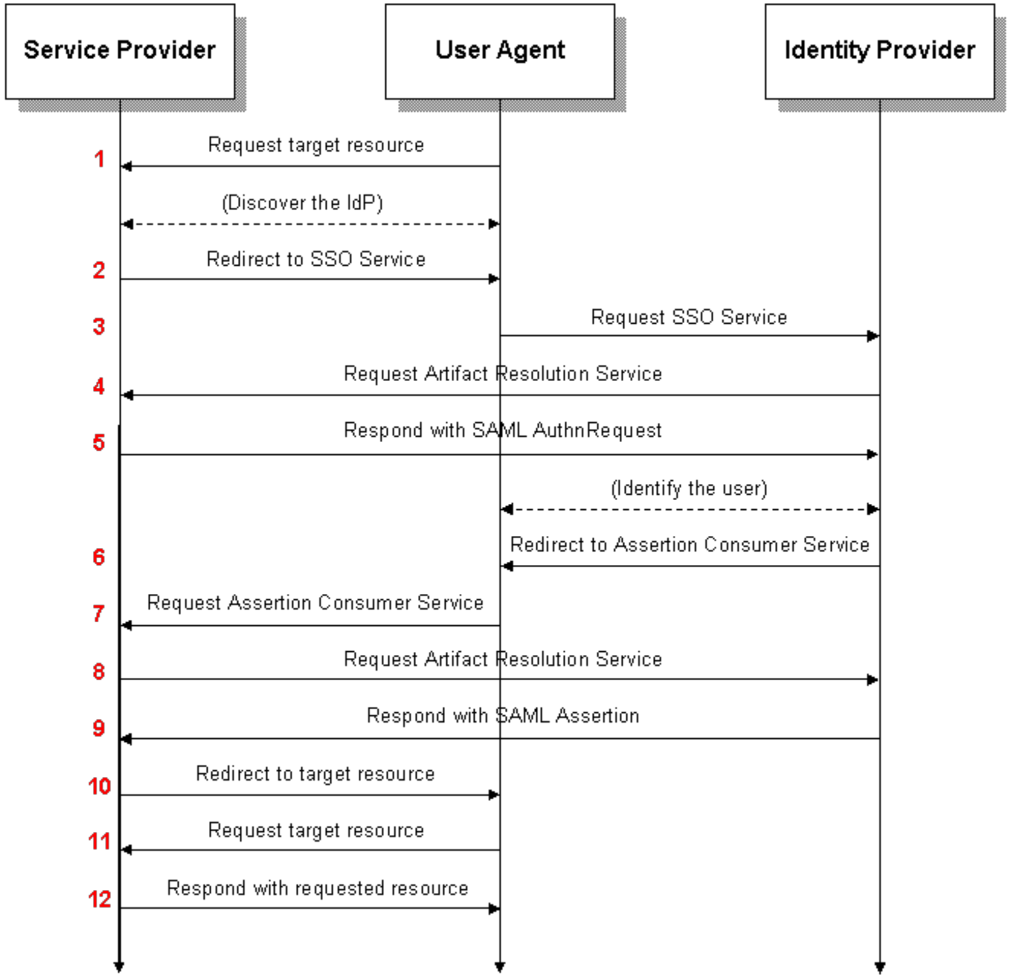
```
https://idp.example.org/SAML2/SSO/Redirect?SAMLRequest=fZFfa8IwFMXfBb9DyXvaJtZ1BqsURRC2
Mabbw95ivc5Am3TJrXPffmmLY3%2FA15Pzuyf33On8XJXBCaxTRmeEhTEJQBdmr%2FRbRp63K3pL5rPhYOpkVdY  ib
%2FCon%2BC9AYfDQRB4WDvRvWWksVoY6ZQTWlbgBBZik9%2FfCR7GorYGTWFK8pu6DknnwKL%2FWEetlxmR8s
BHbHJDWZqOKGdsRJM0kfQAjCUJ43KX8s78ctnIz%2Blp5xpYa4dSo1fjOKGM03i8jSeCMzGevHa2%2FBK5MNo1F
dgN2JMqPLmHc0b6WTmiVbsGoTf5qv66Zq2t60x0wXZ2RKydiCJXh3CWVV1CWJgqanfl0%2Bin8xutxYOvZL18NK
UqPlvZR5el%2BVhYkAgZQdsA6fWVsZXE63W2itrTQ2cVaKV2CjSSqL1v9P%2FAXv4C
```

  - The above message (formatted for readability) may be signed for additional security. In practice the <samlp:AuthnRequest> message is unsigned, leaving the identity provider to identify the sender via SAML metadata.

SAML 2.0 -Web Browser SSO Profile

Magic

Middleware for collaborative Applications and Global virtual Communities


Service Provider
User Agent
Identity Provider
1
Request target resource
(Discover the IdP)
2
Redirect to SSO Service
3
Request SSO Service
4
Request Artifact Resolution Service
5
Respond with SAML AuthnRequest
(Identify the user)
6
Redirect to Assertion Consumer Service
7
Request Assertion Consumer Service
8
Request Artifact Resolution Service
9
Respond with SAML Assertion
10
Redirect to target resource
11
Request target resource
12
Respond with requested resource

Single Sign-On

- The Single Sign-On allows authenticated users in the identity provider to be recognized as authenticated by another service that requests authentication in the same IdP. This applies to the same session of the web browser.

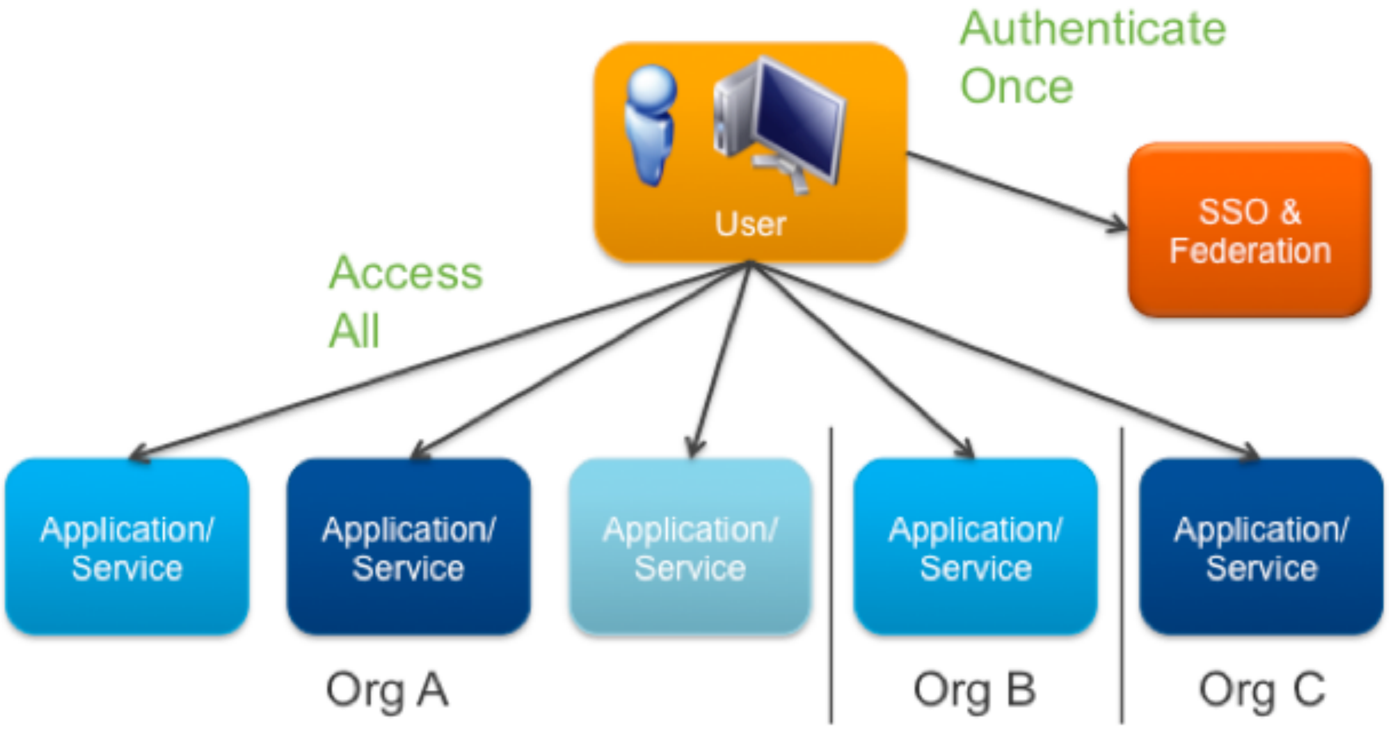
- The service receives the same handle created during authentication, requiring only request attributes and decide on the release of access or not.

- For the user it is transparent, since they do not need to enter your data again for authentication.

Single Sign-On

Magic

Middleware for collaborative Applications and Global vIrtual Communities



Single Logout

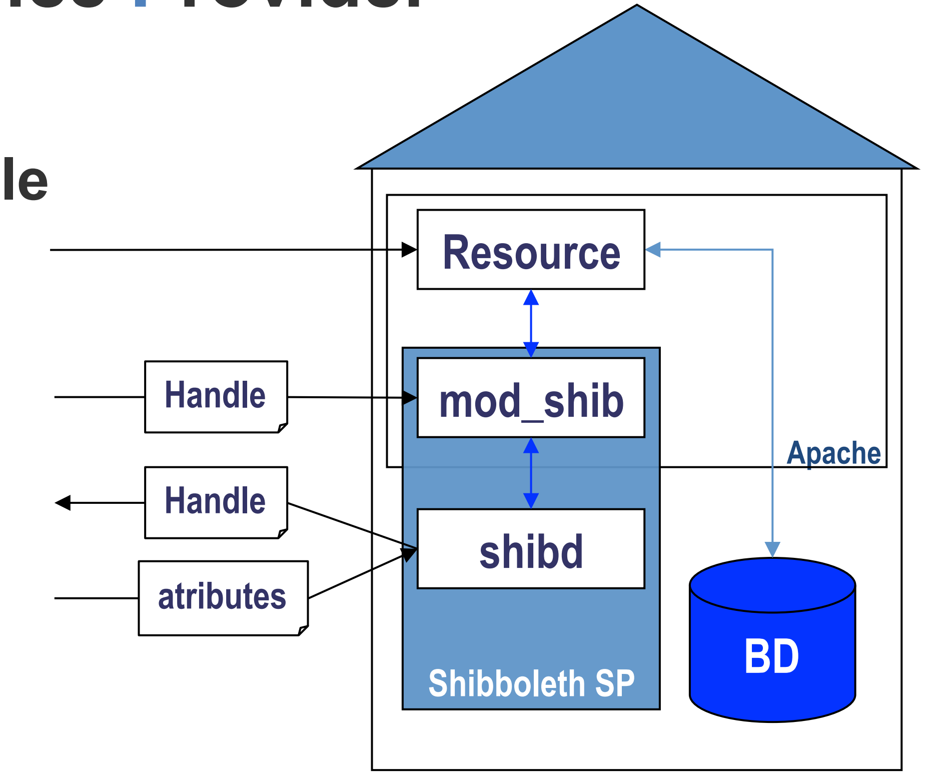
- Logout of all user-initiated applications
  - Issues:
    - When proceed with the logout not all applications are closed
    - Issues with user experience
    - Lack of security
    - Malicious access to services

Shibboleth SP

- Shibboleth Service Provider
  - Implement the SAML 2 protocol
  - Support any programming language or framework
  - Support for Apache and IIS web servers

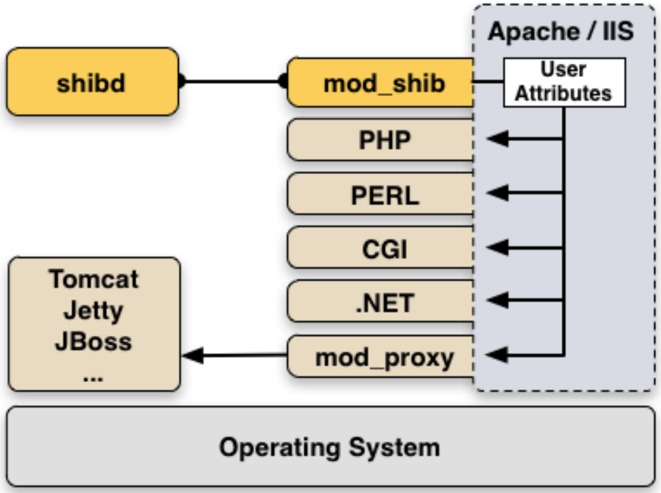
Shibboleth SP – Installation Overview

- Shibboleth Service Provider
  - mod_shib
    - Apache module
  - shibd
    - Daemon



Shibboleth SP - Installation

- Shibboleth SP is implemented in C ++ and can be compiled for different platforms.
- There are official binary distributions for Windows for use with Apache HTTPD or IIS, and RPM packages for use with Apache HTTPD on Linux systems.pen Source
- Can be compiled for different platforms



Shibboleth SP - Components

The shibboleth SP can be seen basically as two components:

- mod_shib: is an authentication module which can be easily carried by Apache. This component is responsible for initiating the authentication protocol;

- shibd: a daemon responsible for resolving artifacts, like order attributes of the authenticated user, for example.

How to install - Linux

- Shibboleth can be installed on most versions of 32- and 64-bit Linux, but is officially supported only on the following distributions at this time:
  - **SUPPORTED OS:**
    - OpenSUSE 11+
    - OS X 10.6+
    - RedHat 5+
    - SLES 10+
    - Solaris 2.9+
    - Windows 2003/2008/2012

How to install - Requirements

- Hardware
  - 3.0 MHz Core2 Duo Pentium-4 (or equivalent) processor
  - 2 GB of memory
  - 160 GB of disk space
- Software
  - NTP Client
  - Sudo
  - SSL enabled for Apache

How to install - Requirements

- For RedHat based distributions is strongly suggested that during any initial setup or testing, that SELinux be left disabled or in permissive mode.
- Root access
- Remove manually compiled/installed Service Provider

How to install

- Installing Apache with PHP and Shibboleth modules in Debian based systems:

```
sh# apt-get -y install apache2 libapache2-mod-php5 libapache2-mod-shib2
```

How to install

- You can test to ensure that the SP is running properly and the surrounding environment is correct by accessing https://localhost/Shibboleth.sso/Status from the actual web server machine.

- You MUST use "localhost" as the hostname or it WILL NOT WORK by default. If this test is successful, then the software is ready for further configuration.

How to install – Apache Configuration

- Open the file /etc/apache2/ports.conf using your preferred file editor.
- Check the existence of the lines “Listen 80” and “Listen 443”.
- If it doesn’t exist, add this lines in the end of the file.

/etc/apache2/sites-available/shibboleth-sp2

How to install – Apache Configuration

- Replace the file contents /etc/apache2/sites-available/default with the following lines:

```
NameVirtualHost *
<VirtualHost *>
        ServerName              $HOSTNAME
        ServerSignature Off

        # Redirecionamento para SSL
        RewriteEngine on
        RewriteCond %{HTTPS} !=on
        RewriteRule ^(.*) https://%{SERVER_NAME}/$1 [R,L]

        DocumentRoot /var/www/
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ErrorLog /var/log/apache2/error.log
        # Possible values include: debug, info, notice, warn, error,
crit,
        # alert, emerg.
        LogLevel info
        CustomLog /var/log/apache2/sp-access-80.log combined
</VirtualHost>
```

How to install – Apache Configuration

- Replace the file contents /etc/apache2/sites-available/shibboleth-sp2 with the following lines:

```
<VirtualHost $ENDERECO_IP:443>
        ServerName              $HOSTNAME
        ServerSignature Off

        SSLEngine               on
        SSLCertificateFile      /etc/ssl/certs/$HOSTNAME.crt
        SSLCertificateKeyFile   /etc/ssl/private/$HOSTNAME.key

        #ShibURLScheme https

        DocumentRoot /var/www/
        <Directory /var/www/>
                Options -Indexes -FollowSymLinks -MultiViews
                AllowOverride None
                Order deny,allow
                Allow from all
        </Directory>
        <Location /secure>
                AuthType shibboleth
                ShibRequireSession On
                require valid-user
                Order allow,deny
                allow from all
         </Location>
#          <Location /moodle/auth/shibboleth>
#                  AuthType shibboleth
#                  ShibRequireSession On
#                  require valid-user
#          </Location>

        ErrorLog /var/log/apache2/error.log
        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel info
        CustomLog /var/log/apache2/sp-access-443.log combined
</VirtualHost>
```

/etc/apache2/sites-available/shibboleth-sp2

How to install – Apache Configuration

- Activate the module Rewrite, Shibboleth e SSL on Apache with the following command lines:

- a2enmod shib2
- a2enmod ssl
- a2enmod rewrite

/etc/apache2/sites-available/shibboleth-sp2

How to install – Apache Configuration

- Enable the site and do deleting unnecessary files. To do so, run the following command line:
  - a2ensite shibboleth-sp2
  - rm -rf /var/www/index.html
  - rm -rf /etc/shibboleth/IQ-metadata.xml

How to install – Firewall Configuration

- Firewall rules:
- At the /etc/default/firewall, add the following lines:

```
# Liberação do Apache (Shibboleth-SP)                    #SHIB-SP
iptables -A INPUT -p tcp -m tcp --dport   80 -j ACCEPT   #SHIB-SP
iptables -A INPUT -p tcp -m tcp --dport  443 -j ACCEPT   #SHIB-SP
                                                          #SHIB-SP
```

SP Configuration - Certificates

- Aims to ensure that an entity is who they say they
- Based on asymmetric keys

SP Configuration - Certificates

- SSL with browser
- SSL with IP
- Message signature
- Certificate metadata signature

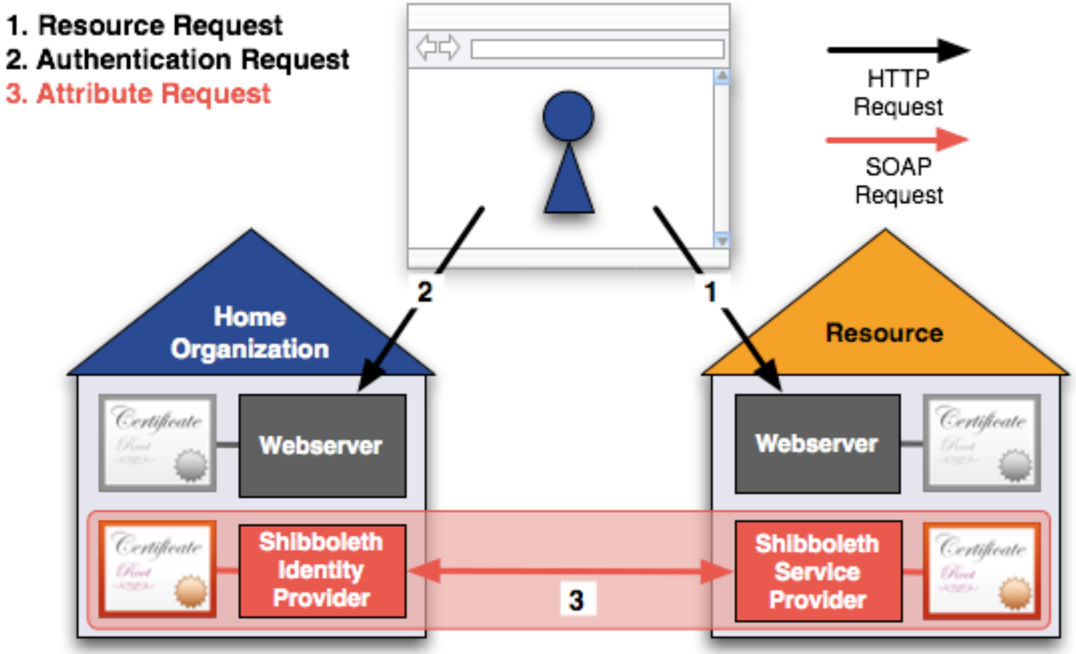


Figura 3.1

SP Configuration - Certificates

- Certificate generation with preset parameters (openssl.cnf)
  - Key Generation
    - #openssl genrsa 2048 -config openssl.cnf > `**nomekey**`.key
  - Self signed certificate Generation
    - #openssl req -new -x509 -nodes -days 1095 -sha1 -key `**nomekey**`.key -set_serial 00 -config openssl.cnf > `**nomeCert**`.crt

SP Configuration – shibboleth2.xml

- shibboleth2.xml file
  - Sets most SP settings
  - It was renamed to emphasize the incompatibility with shibboleth.xml file version 1.3

- Default location:
  - /etc/shibboleth/shibboleth2.xml

SP Configuration – shibboleth2.xml

- shibboleth2.xml file
  - Many attributes reference other elements defined within the file
  - Some settings made in this file:
    - Indication of the path to log files (OutOfProcess tags, InProcess)
    - Identification of SP (EntityId)
    - Display of Metadata partners (MetadataProvider)
    - Virtual Hosts Settings (RequestMapper)
    - key path statement and certificate for signing and SSL

SP Configuration – shibboleth2.xml

- shibboleth2.xml file
  - Some settings made in this file:
    - Customizing error pages
    - Mapping requests and their settings
    - specific settings for applications
    - IdP statement for Authentication (WAYF, DS, or IdP directly)
    - Path to atributes mapping file (Attribute-map.xml)
    - Way to atributes filter policy file (Attribute-policy.xml)

SP Configuration – shibboleth2.xml

- shibboleth2.xml structure

```
<SPConfig xmlns="urn:mace:shibboleth:sp:config:2.0‟
   xmlns:conf="urn:mace:shibboleth:sp:config:2.0‟
   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata‟
   logger="shibboleth/syslog.logger" clockSkew="180">
         <Extensions/>
         <OutOfProcess logger="shibboleth/shibd.logger"/>
         <InProcess logger="shibboleth/native.logger"/>
         <Listener/>
         <StorageService/>
          <SessionCache/>
          <ReplayCache/>
          <ArtifactMap/>
        <RequestMapper/>
        <ApplicationDefaults id="default" policyId="default"
          entityID=‟https://sp.example.org/shibboleth”
     homeURL="https://sp.example.org/index.html"/>
        <SecurityPolicies/>
</SPConfig>
```

SP Configuration – shibboleth2.xml

- Main Settings - Request Mapper
  - Sets the pages that will be protected by shibboleth
    - Equivalent resource <Location> in Apache
  - It allows the SP to isolate the differences of servers
    - Apache has resource Location, IIS does not have
  - Maps each request to the specific settings for each application (attribute applicationId)
  - Child element: RequestMap

SP Configuration – shibboleth2.xml

- Main Settings - RequestMap
  - attributes
    - Inherit the attributes of the root element RequestMapper
    - applicationId - must hold an applicationId attribute with the default value
  - Child elements
    - <Host> - virtual hosts configuration, you can reset the attribute applicationId with reference to another ApplicationDefaults
    - AccessControl - <.htaccess>, <AccessControlProvider> and <AccessControl>

SP Configuration – shibboleth2.xml

- Main Settings – RequestMap

```
<RequestMap applicationId="default">
  <Host name="www.example.org">
      <Path name="secure" authType="shibboleth" requireSession="true"/>
    </Host>
  <Host name="admin.example.org" applicationId="admin"
       authType="shibboleth" requireSession="true">
    <AccessControl>
     <Rule require="affiliation">faculty@osu.edu student@osu.edu</Rule>
    </AccessControl>
  </Host>
</RequestMap>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – atributes
  - Id
    - String – required
    - Unique identifier of the ApplicationDefaults, should have the value "default”
  - entityID
    - URI – required
    - It is the unique SAML identifier to name the SP
  - PolicyId
    - String – required
    - Reference to the policy element defined within the <Security Policies>

SP Configuration – shibboleth2.xml

- ApplicationDefaults – atributes
  - REMOTE_USER
    - attribute to be mapped as REMOTE_USER header
    - list of strings delimited by spaces
    - atributes listed in order of precedence
  - signing
    - true, false, front or back (default is false)
    - XML messaging signature control
  - encryption
    - true, false, front or back (default is false)
    - XML messaging encryption controls

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - <Sessions>
    - Session behavior settings for the application

```
<Sessions lifetime="28800" timeout="3600‶
        checkAddress="false" handlerURL="/Shibboleth.sso"
        handlerSSL="false"    exportLocation="http://localhost/
Shibboleth.sso/GetAssertion"  exportACL="127.0.0.1"
idpHistory="false" idpHistoryDays="7">

        <SessionInitiator/>
        <md:AssertionConsumerService/>
        <LogoutInitiator/>
        <md:SingleLogoutService/>
        <md:ManageNameIDService/>
        <md:ArtifactResolutionService/>
        <Handler/>
</Sessions>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - <Sessions>
    - <SessionInitiator> (prior to Version2.4)

```
<!--Default example directs to a specific IdP's SSO service (favoring SAML 2
        over Shib 1). -->

<SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Intranet"
  relayState="cookie" entityID="https://idp.curso/idp/">
   <SessionInitiator type="SAML2" defaultACSIndex="1" template="template.html"/>
   <SessionInitiator type="Shib1" defaultACSIndex="5"/>
</SessionInitiator>

<!--Example using an old-style WAYF, which means Shib 1 only unless an entityID
    is provided. -->

<SessionInitiator type="Chaining" Location="/WAYF" id="WAYF" relayState="cookie">
   <SessionInitiator type="SAML2" defaultACSIndex="1" template="template.html"/>
   <SessionInitiator type="Shib1" defaultACSIndex="5"/>
   <SessionInitiator type="WAYF" defaultACSIndex="5" URL="https://wayf.org/WAYF"/>
</SessionInitiator>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - <SSO> (Version 2.4 and higher)
    - Replaces the <Session Initiator> using earlier versions

A basic example using a single, fixed IdP, supporting the usual common SAML protocols:

```
<SSO entityID="https://idp.example.org/idp/shibboleth">
  SAML2 SAML1
</SSO>
```

An example using a SAML Discovery Service:

```
<SSO discoveryProtocol="SAMLDS" discoveryURL="https://examplefederation.org/DS">
  SAML2 SAML1
</SSO>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - <Errors>
    - Configuration of the error handling in the application

```
<Errors session="sessionError.html" metadata="metadataError.html"
        access="accessError.html" ssl="sslError.html"
        localLogout="localLogout.html‶ globalLogout="globalLogout.html"
        supportContact="root@localhost" logoLocation="/shibboleth-  sp/
logo.jpg" styleSheet="/shibboleth-sp/main.css"/>
```

SP Configuration – shibboleth2.xml

Magic

Middleware for collaborative Applications and Global virtual Communities

- ApplicationDefaults – child elements
  - RelyingParty
    - Overwrite settings customizing the behavior related to communication for providers of identity or specific groups of providers.

```
<RelyingParty Name="SpecialFederation" keyName="SpecialKey"
authtype="basic" signing="true" encryption="true“
timeout=“3600" signedAssertions="true"/>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - Notify
    - Configure application logout notification

```
<Notify Channel=“front" Location="https://#YOUR_HOSTNAME#/
#PATH_TO#/logout.php" />
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - MetadataProvider
    - Metadata of recognized identity providers

```
<MetadataProvider type="Chaining">
   <!-- Example of remotely supplied batch of signed metadata. -->
   <MetadataProvider type="XML"
        uri= "http://feder.org/federation-metadata.xml"
        backingFilePath="federation-metadata.xml" reloadInterval="7200">
        <MetadataFilter type="RequireValidUntil"
                maxValidityInterval="241920"/>
        <MetadataFilter type="Signature" certificate="fedsigner.pem"/>
   </MetadataProvider>

   <!-- Example of locally maintained metadata.-->
   <MetadataProvider type="XML" file="idp-metadata.xml"/>
</MetadataProvider>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - TrustEngine
    - Mechanism used by the SP to authenticate messages

```
<TrustEngine type="Chaining">
           <TrustEngine type="ExplicitKey" />
           <TrustEngine type="PKIX" />
</TrustEngine>
```

SP Configuration – shibboleth2.xml

Magic

Middleware for collaborative Applications and Global virtual Communities

- ApplicationDefaults – child elements
  - AttributeExtractor
    - How SAML attributes are encoded and exposed to applications
    - Reference attribute-map.xml file

```
<AttributeExtractor type="XML" path="attribute-map.xml" />
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - CredentialResolver
    - Sets the private key and certificate used by the SP to identify the providers of Identity

```
<CredentialResolver type="File" key="sp-key.pem"
       certificate="sp-cert.pem" />

                    Ou
<CredentialResolver type="File">
    <Key>
        <Path>/etc/shibboleth/sp.key</Path>
    </Key>
    <Certificate>
        <Path>/etc/shibboleth/sp.crt</Path>
   </Certificate>
</CredentialResolver>
```

SP Configuration – shibboleth2.xml

- ApplicationDefaults – child elements
  - <ApplicationOverride>
    - Elements <ApplicationOverride> inherits all settings of the <ApplicationDefaults> but can also replace them
    - Configure applications with different behavior from default
    - Some atributes of <ApplicationDefaults> become optional for <ApplicationOverride>

```
<ApplicationDefaults>
...
    <ApplicationOverride id="other-app" entityID="https://other.org/shibboleth">
             <Sessions lifetime="28800" timeout="3600" checkAddress="false"
                handlerURL="/otherapp/Shibboleth.sso" handlerSSL="false"
                exportLocation="http://localhost/Shibboleth.sso/GetAssertion"
                idpHistory="false" idpHistoryDays="7">
    </ApplicationOverride>
...
</ApplicationDefaults>
```

SP Configuration – shibboleth2.xml

- Main Settings - Application Defaults

```
<RequestMapper type="Native">
        <RequestMap applicationId="default">
            <Host name="service.university.org" authType="shibboleth"
                 requireSession="true"/>
            <Host name="other.university.org" applicationId="other-app"
                 authType="shibboleth" requireSession="true"/>
        </RequestMap>
</RequestMapper>

<ApplicationDefaults id="default" policyId="default"  entityID="https://
   service.university.org/shibboleth"       homeURL="https://
   service.university.org/welcome/" REMOTE_USER="eppn persistent-id targeted-id"
   signing="false" encryption="false">
  [...]
        <!-- Overrides for other-app -->
        <ApplicationOverride id="other-app"
         entityID="https://other.org/shibboleth"/>
</ApplicationDefaults>
```

SP Configuration – shibboleth2.xml

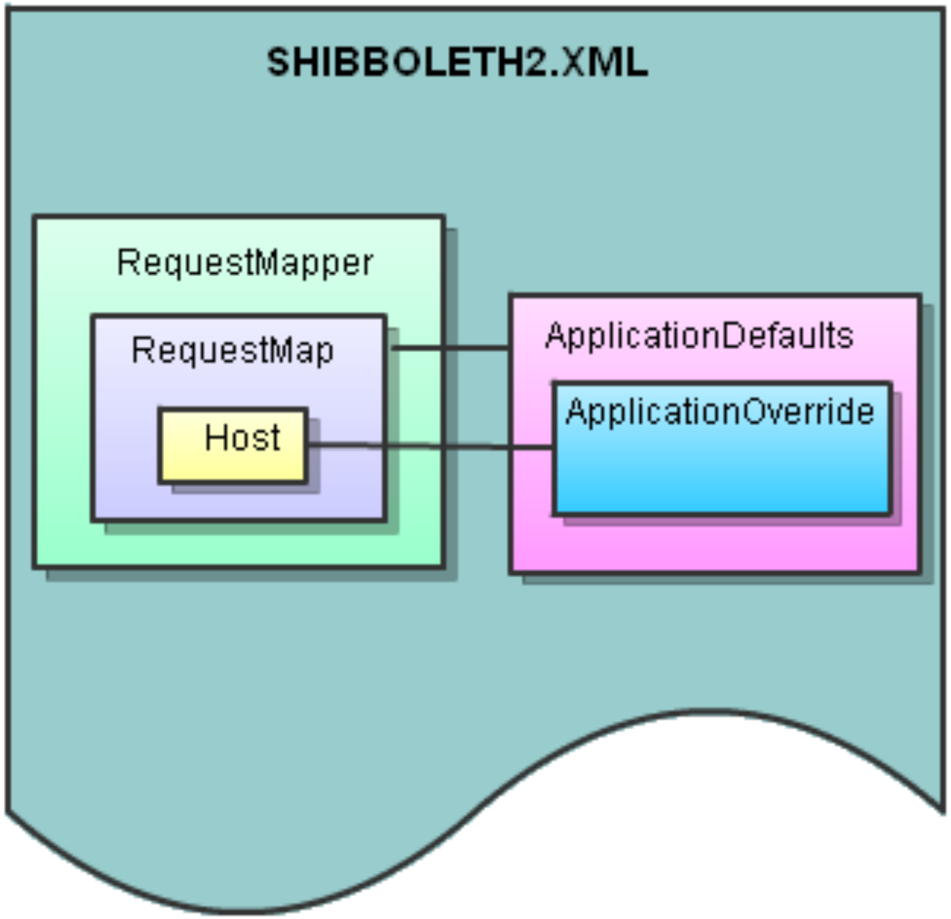


Figura 3.2

SP Configuration – attribute-map.xml

- attribute-map.xml file
  - Extraction and mapping attributes of SAML assertions to the environment variables and HTTP headers
  - Tag Attribute
    - Maps the attribute extracted in a variable or HTTP header
    - attribute name - Corresponds to the formal SAML name used for the attribute in the IdP, usually a URI.
    - id attribute - short name and determines the environment variable or HTTP header by which the attribute will be available to the application

```
<Attribute name="urn:oid:2.5.4.42" id="givenName"/>
<Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
```

SP Configuration – attribute-map.xml

- Apache / HTTP headers
  - Shibboleth 2 does not recommend the use of HTTP headers
  - Environment Variables are default to be more secure
  - To enable the use of HTTP headers enable the property
  - Example Apache:

```
<Location /homologa>
        AuthType shibboleth
        ShibRequereSession On
        require valid-user
        ShibUseHeaders On
</Location>
```

SP Configuration – attribute-map.xml

- child element - AttributeDecode
  - Optional tag that indicates the extractor to be used to perform the extraction of SAML Assertions
  - Used to extract more complex atributes

```
<Attribute name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation“
        id="affiliation">
     <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>
```

SP Configuration – attribute-map.xml

```
<Attributes xmlns="urn:mace:shibboleth:2.0:attribute-map" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName" id="eppn">
           <AttributeDecoder xsi:type="ScopedAttributeDecoder" />
  </Attribute>
  <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
          <AttributeDecoder xsi:type="ScopedAttributeDecoder" />
  </Attribute>
  <Attribute name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation"
         id="affiliation">
     <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
  </Attribute>
  ...
  <Attribute name="urn:oid:2.5.4.42" id="givenName"/>
  <Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
</Atributes>
```

SP Configuration – attribute-policy.xml

- Configuring the SP attribute acceptance policy regarding the IdP
- The first part of the file defines rules for acceptance

```
<afp:PermitValueRule id="eduPersonAffiliationValues" xsi:type="OR">
   <Rule xsi:type="AttributeValueString" value="faculty" />
   <Rule xsi:type="AttributeValueString" value="student" />
   <Rule xsi:type="AttributeValueString" value="staff" />
</afp:PermitValueRule>
```

  - If the value of the string has the values "faculty", "student" or "staff" the rule is accepted

SP Configuration – attribute-policy.xml

- Operators acceptance rules
  - ANY - Always evaluates to true
  - AND - Evaluates to True if all the rules are true
  - OR - Evaluates to TRUE if any rule contained is true
  - NOT - Evaluates to TRUE if the rule contained evaluates to false
  - AttributeRequesterRegex - Evaluates to TRUE if the attribute requesting entity ID meets certain regular expression
  - AttributeValueString - evaluates to TRUE if the value of a particular attribute corresponds to a given string

SP Configuration – attribute-policy.xml

- Second part of the file maps the rule set with a specific attribute or group of atributes

```
 <afp:AttributeFilterPolicy>
      <afp:PolicyRequirementRule xsi:type="ANY"/>
      <afp:AttributeRule attributeID="affiliation">
          <afp:PermitValueRule xsi:type="AND">
                 <RuleReference ref="eduPersonAffiliationValues"/>
                 <RuleReference ref="ScopingRules"/>
          </afp:PermitValueRule>
      </afp:AttributeRule>
      <afp:AttributeRule attributeID="unscoped-affiliation">
                 <afp:PermitValueRuleReference ref="eduPersonAffiliationValues"/>
      </afp:AttributeRule>
</afp:AttributeFilterPolicy>
```

- The attribute attributeID contains the name of the attribute which the rule applies in a same way as specified in the file attribute-map.xml

SP Configuration – attribute-policy.xml

```
<afp:AttributeFilterPolicyGroup xmlns="urn:mace:shibboleth:2.0:afp:mf:basic"
xmlns:basic="urn:mace:shibboleth:2.0:afp:mf:basic" xmlns:afp="urn:mace:shibboleth:2.0:afp"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
   <afp:PermitValueRule id="ScopingRules" xsi:type="AND">
            <Rule xsi:type="NOT">
               <Rule xsi:type="AttributeValueRegex" regex="@" />
           </Rule>
          <Rule xsi:type="saml:AttributeScopeMatchesShibMDScope"
          xmlns:saml="urn:mace:shibboleth:2.0:afp:mf:saml" />
  </afp:PermitValueRule>

  <afp:AttributeFilterPolicy>
         <afp:PolicyRequirementRule xsi:type="ANY"/>
          <afp:AttributeRule attributeID="affiliation">
             <afp:PermitValueRule xsi:type="AND">
               <RuleReference ref="ScopingRules" />
               <RuleReference ref="eduPersonAffiliationValues"/>
             </afp:PermitValueRule>
         </afp:AttributeRule>
         <afp:AttributeRule attributeID="*">
            <afp:PermitValueRule xsi:type="ANY" />
        </afp:AttributeRule>
  </afp:AttributeFilterPolicy>

</afp:AttributeFilterPolicyGroup>
```

SP Configuration – Log Polices

- SP Log files:
  - /var/log/shibboleth
    - shibd.log, shibd_warn.log
    - transaction.log
    - Signature.log
  - /var/log/apache2
    - access.log
    - error.log
- Logs levels configuration files
  - /etc/shibboleth
    - syslog.logger
    - shibd.logger, native.logger

Apache HTTPD configuration

- Load module *mod_shib*
- Protect settings with *authType shibboleth*
- SSL configuration is usually required

Apache HTTPD configuration

- Virtual host configuration example in apache

```
<VirtualHost *>
        ServerName              $HOSTNAME
        ServerSignature Off

        # Redirecionamento para SSL
        RewriteEngine on
        RewriteCond %{HTTPS} !=on
        RewriteRule ^(.*) https://%{SERVER_NAME}/$1 [R,L]

        DocumentRoot /var/www/
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
</VirtualHost>
```

Apache HTTPD configuration – Certificates

- Virtual Host configuration example for HTTPS

```
NameVirtualHost *:443
<VirtualHost ENDEREÇO_IP:443>
     ServerName  servidor.instituicao.br
     SSLEngine   on
     SSLCertificateFile /etc/shibboleth/nomeCertificate.crt
     SSLCertificateKeyFile /etc/shibboleth/nomekey.key
     DocumentRoot /var/www-ssl/
        <Location /secure>
                AuthType shibboleth
                ShibRequireSession On
                require valid-user
                Order allow,deny
                allow from all
         </Location>
</VirtualHost>
```

Discovery Service - DS

- It allows the user to choose the identity providers for Authenticator

- Usually it goes through intermediary for listing of possibilities

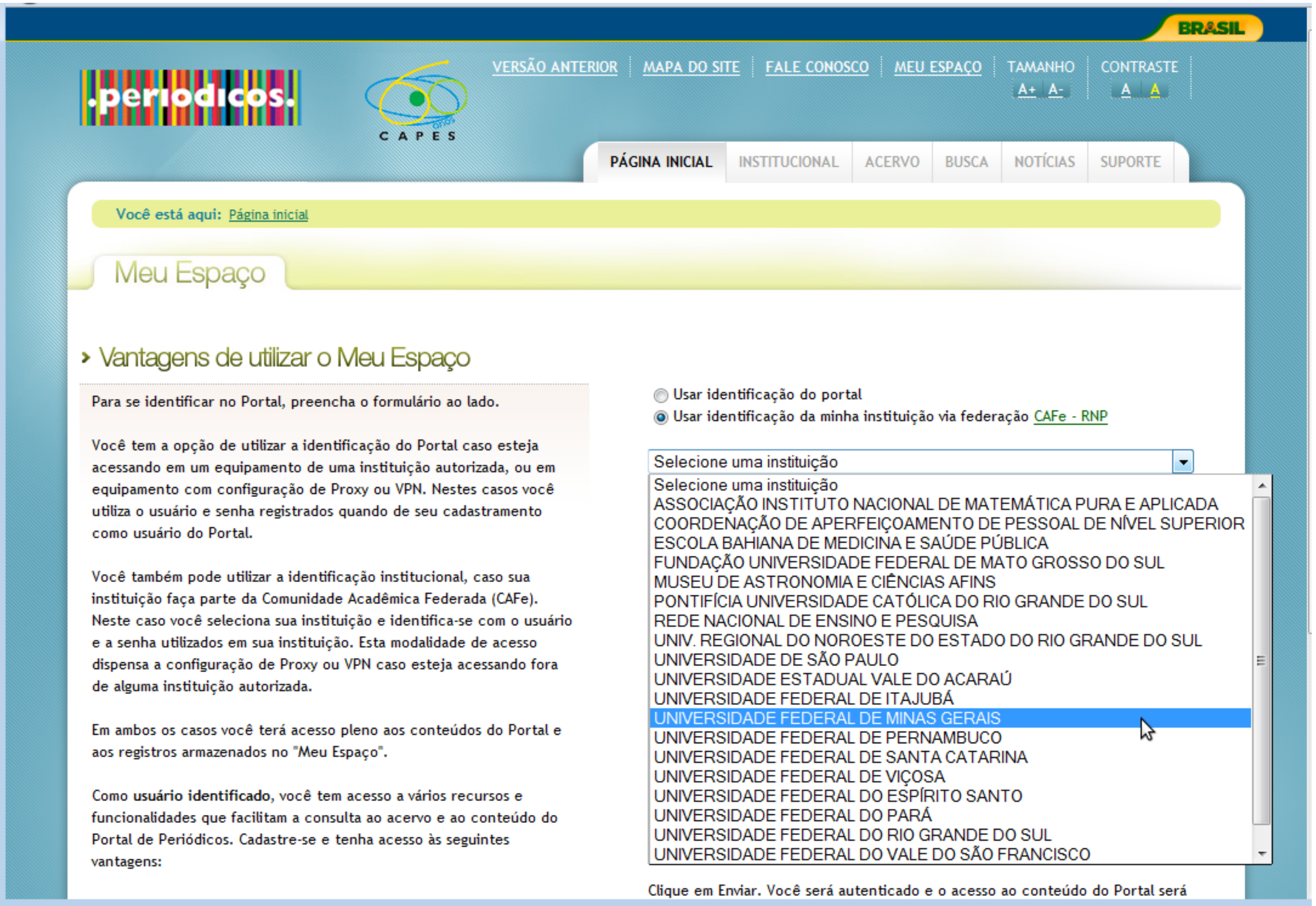
- You can also redirect directly to the authenticator

Architecture

- Embedded Discovery Service
  - The **Embedded Discovery Service** provides a list of institutions inside the page of SP allowing a user to select which Identity Provider they will use when accessing a it
- Centralized Discovery Service
  - The **Centralized Discovery Service** provides a external web interface that allows a user to select which Identity Provider they will use when accessing a Service Provider.

Embeded Discovery Service

- An example:



Você está aqui: Página inicial

Meu Espaço

› Vantagens de utilizar o Meu Espaço

Para se identificar no Portal, preencha o formulário ao lado.

Você tem a opção de utilizar a identificação do Portal caso esteja acessando em um equipamento de uma instituição autorizada, ou em equipamento com configuração de Proxy ou VPN. Nestes casos você utiliza o usuário e senha registrados quando de seu cadastramento como usuário do Portal.

Você também pode utilizar a identificação institucional, caso sua instituição faça parte da Comunidade Acadêmica Federada (CAFe). Neste caso você seleciona sua instituição e identifica-se com o usuário e a senha utilizados em sua instituição. Esta modalidade de acesso dispensa a configuração de Proxy ou VPN caso esteja acessando fora de alguma instituição autorizada.

Em ambos os casos você terá acesso pleno aos conteúdos do Portal e aos registros armazenados no "Meu Espaço".

Como **usuário identificado**, você tem acesso a vários recursos e funcionalidades que facilitam a consulta ao acervo e ao conteúdo do Portal de Periódicos. Cadastre-se e tenha acesso às seguintes vantagens:

○ Usar identificação do portal
● Usar identificação da minha instituição via federação CAFe - RNP

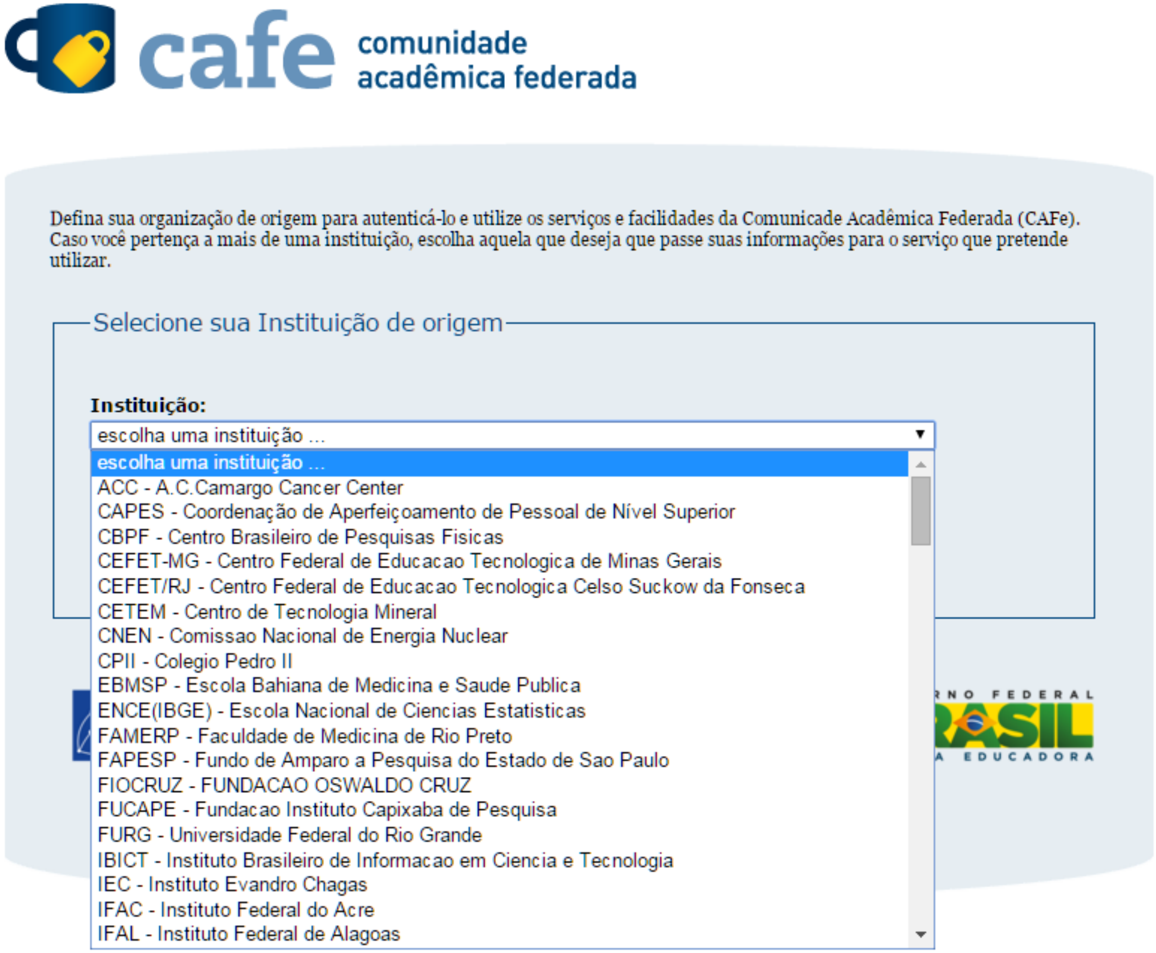
Selecione uma instituição

Selecione uma instituição
ASSOCIAÇÃO INSTITUTO NACIONAL DE MATEMÁTICA PURA E APLICADA
COORDENAÇÃO DE APERFEIÇOAMENTO DE PESSOAL DE NÍVEL SUPERIOR
ESCOLA BAHIANA DE MEDICINA E SAÚDE PÚBLICA
FUNDAÇÃO UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
MUSEU DE ASTRONOMIA E CIÊNCIAS AFINS
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
REDE NACIONAL DE ENSINO E PESQUISA
UNIV. REGIONAL DO NOROESTE DO ESTADO DO RIO GRANDE DO SUL
UNIVERSIDADE DE SÃO PAULO
UNIVERSIDADE ESTADUAL VALE DO ACARAÚ
UNIVERSIDADE FEDERAL DE ITAJUBÁ
UNIVERSIDADE FEDERAL DE MINAS GERAIS
UNIVERSIDADE FEDERAL DE PERNAMBUCO
UNIVERSIDADE FEDERAL DE SANTA CATARINA
UNIVERSIDADE FEDERAL DE VIÇOSA
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
UNIVERSIDADE FEDERAL DO PARÁ
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO

Clique em Enviar. Você será autenticado e o acesso ao conteúdo do Portal será

Centralized Discovery Service

- An example:



cafe comunidade acadêmica federada

Defina sua organização de origem para autenticá-lo e utilize os serviços e facilidades da Comunicade Acadêmica Federada (CAFe). Caso você pertença a mais de uma instituição, escolha aquela que deseja que passe suas informações para o serviço que pretende utilizar.

Selecione sua Instituição de origem

Instituição: escolha uma instituição ...

- escolha uma instituição ...
- ACC - A.C.Camargo Cancer Center
- CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
- CBPF - Centro Brasileiro de Pesquisas Fisicas
- CEFET-MG - Centro Federal de Educacao Tecnologica de Minas Gerais
- CEFET/RJ - Centro Federal de Educacao Tecnologica Celso Suckow da Fonseca
- CETEM - Centro de Tecnologia Mineral
- CNEN - Comissao Nacional de Energia Nuclear
- CPII - Colegio Pedro II
- EBMSP - Escola Bahiana de Medicina e Saude Publica
- ENCE(IBGE) - Escola Nacional de Ciencias Estatisticas
- FAMERP - Faculdade de Medicina de Rio Preto
- FAPESP - Fundo de Amparo a Pesquisa do Estado de Sao Paulo
- FIOCRUZ - FUNDACAO OSWALDO CRUZ
- FUCAPE - Fundacao Instituto Capixaba de Pesquisa
- FURG - Universidade Federal do Rio Grande
- IBICT - Instituto Brasileiro de Informacao em Ciencia e Tecnologia
- IEC - Instituto Evandro Chagas
- IFAC - Instituto Federal do Acre
- IFAL - Instituto Federal de Alagoas

Installing an Embedded DS (EDS)

- Download the Embedded Discovery Service Distribution
  - http://shibboleth.net/downloads/embedded-discovery-service/latest/shibboleth-embedded-ds-1.1.0.tar.gz)
- Unzip the distribution
  - tar -zxvf shibboleth-embedded-ds-1.1.0.tar.gz
- Copy the Javascript files (idpselect_config.js and idpselectjs, and the CSS (idpselect.css) files in to the location from where your web server is serving static content.
- For testing purposes you may chose to copy the index.html file to the same location.
- Once complete, open a browser and test to make sure each file is reachable.

EDS - Web Page Setup

- There are two ways to go about setting up the web page that will act as the actual discovery service.

- The first way, useful for testing, is to use the index.html file that comes with the EDS distribution.
- If, however, you'd like to embed the discovery service in to an existing page template you have for your site (the behavior we'd expect for an production install) then make the following modifications to your HTML page

EDS – Web Page Setup

- Within the head element add a link to the discovery service CSS page similar to:

```
<link rel="stylesheet" type="text/css" href="idpselect.css">
```

EDS – Web Page Setup

- Within the body element add the following div element in the location you wish to use for the upper-left corner of the discovery service (i.e., where you want it to start rendering)

```
<div id="idpSelect"></div>
```

EDS – Web Page Setup

- At the bottom of the body element (just before the closing tag) add the following script definitions:

```
<!-- Load languages scripts -->
<script src="idpselect_config.js" type="text/javascript"
language="javascript"></script>
<script src="idpselect.js" type="text/javascript"
language="javascript"></script>
```

EDS – Web Page Setup

- The precise behavior of the EDS is controlled by various configuration options which are set in the IdPSelectUIParms class defined in the idpselect_config.js file. This is the only file you should need to edit to configure the EDS.

- A complete list of user-changeable configuration options is presented below. For a production installation, at least the dataSource, defaultLogo, defaultLogoHeight, defaultLogoWidth, and helpURL options should be set.

- Parameters with a superscriptlike this indicate that the parameter is only available starting at the release whose number figures in the superscript.

EDS – Web Page Setup

- alwaysShow
  - default value: true
  - This controls whether the search box always shows results as soon as you start typing in it. If false, results will only be shown when there are fewer than maxResults of them.

EDS – Web Page Setup

- dataSource
  - default value: /Shibboleth.sso/DiscoFeed
  - This is the URL of the source of the data feed of IdPs to the DS. This feed must be at the same location as the DS itself and so it is usual for the protocol and host part of the URL (https://example.org) to be dropped.
  - The data source is a JSON file.

EDS – Web Page Setup

- defaultLanguage
  - *default value: 'en'*
  - This is the language to be used for display if it cannot be determined from the browser.

EDS – Web Page Setup

- defaultLogo
  - *default value: null*
  - This is the URL of the logo which will be displayed if a previously selected IdP has no logo declared for it (see 4. Metadata Considerations below).
  - It is recommended that this be a transparent box of size 80x60 pixels.

EDS – Web Page Setup

- defaultLogoHeight, defaultLogoWidth
  - *default value: null*
  - The dimensions of the default logo.

EDS – Web Page Setup

- defaultReturn
  - default value: null
  - If this is set it allows the EDS to be set up in a such a way that it does not need to be approached via the Discovery Service protocol. The *defaultReturn* and *defaultReturnIDParam* parameters supply the values which would normally be supplied via the protocol.
  - Note that the values will be URL-encoded prior to being sent back. That is, this value could be constructed by URL-decoding the value to the "return" parameter in a standard Discovery Service protocol request.

EDS – Web Page Setup

- defaultReturnIDParam
  - *default value: null*
  - If specified, this supplies the same value that the *returnIDParam* would in a DS protocol request.
  - Note that, according to the specification, if the value of *defaultReturn* is non-null and the value of *defaultReturnIDParam* is null, then the protocol default of *entityID* is used.

EDS – Web Page Setup

- doNotCollapse
  - *default value: true*
  - If set to false, then if none of the preferred IdPs have an associated logo, then the height of the EDS is collapsed appropriately

EDS – Web Page Setup

- helpURL
  - *default value: null*
  - This is the URL to which users are dispatched when they click on the help link.

EDS – Web Page Setup

- ie6Hack
  - *default value: false*
  - This can be used to ease some issues with IE6 and z-axis. IE6 is not supported by the EDS.

EDS – Web Page Setup

- ignoreKeywords
  - default value: false
  - If this value is set to true then the contents of any <mdui:Keywords/> will not be used to find suggested names.

EDS – Web Page Setup

- ignoreURLParams
  - *default value:false*
  - If value is set to false then the EDS entirely ignores all parameterization to the URL (including any DS protocol parameterization) and instead always relised on the **defaultReturn** configuration parameter

EDS – Web Page Setup

- insertAtDiv
  - *default value: 'idpSelect'*
  - This is the id of the <div> inside which the EDS will be constructed.

EDS – Web Page Setup

- langBundles
  - About Internationalization
  - (https://wiki.shibboleth.net/confluence/display/EDS10/3.3+Internationalization)

EDS – Web Page Setup

- maxResults
  - default value: 10
  - This controls how many results to display in response to typing in the search box. If alwaysShow is true, then the first maxResults values are shown.
  - If it is false then nothing is shown until there are no more than maxResults values.

EDS – Web Page Setup

- myEntityID
  - default value: null
  - If this is supplied, then the entityID supplied via the DS protocol is checked against this string.

EDS – Web Page Setup

- noWriteCookie1.1
  - default value: false
  - If this is set to true, then the EDS does not save the selected IdP as a cookie. If the EDS shares a domain with a Shibboleth SP, then this setting might be combined with enabling the SP's IdP history can be enabled via the idpHistory attribute on the SP's <Sessions> element.
  - This has the advantage that only successfully authenticated IdPs are store, whereas the EDS would otherwise save all selected IdPs.

EDS – Web Page Setup

- preferredIdP
  - *default value: null*
  - If this is supplied, then it must be an array of entityIDs of IdPs which are considered preferred by this SP.
  - Preferred IdPs are always displayed regardless of whether the user has previously selected them.

EDS – Web Page Setup

- hiddenIdPs
  - *default value: null*
  - If this is supplied then it must be an array of entityIDs of IdPs which are *not* to be displayed by the EDS.

EDS – Web Page Setup

- samlIdPCookieTTL
  - *default value: 730*
  - This is the lifetime (in days) of the cookie which is used to store the list of previously visited sites.
  - This cookie is in the standard _saml_idp format as described in the SAML profiles specification

EDS – Web Page Setup

- setFocusTextBox
  - default value: true
  - By default the text box is always given focus after the EDS has been drawn.
  - Web sites may wish to leave the focus somewhere else; in that case they should set this value to false.

EDS – Web Page Setup

- showListFirst1.1
  - default value: false
  - If set to true then the dropdown box of IdPs is initially shown.
  - If the value is absent or set to false then the free text "search as you type" box is displayed.

EDS – Web Page Setup

- testGUI
  - default value: false
  - It is usually a configuration error to browse to the EDS page, rather than be redirected to it by SP when you visit a secured page.
  - If this value is set to true, then none of the DS parameters are checked.
  - This allows the GUI to be tested without the full DS parameter string (and without dispatching the browser to the selected IdP).

Installing an Centralized DS

- This section describes the Centralized Discovery Service, which is primarily intended for use by identity federations and other large groups wishing to providing a backstop discovery service.
- Individual service providers, in particular, are recommended to install the Embedded Discovery Service.

Installing an Centralized DS

- **Before You Begin**
  - The first question you should ask is whether you need to install the Discovery Service. If you're working in a non-Java environment, you may find it easier to build a selection page in a more native fashion. The SP also supports the Embedded Discovery Service which is usually a better choice for SPs which need to implement discovery.
  - If you do decide to install this service, you'll need to collect the metadata sources that will contain the IdPs that users will select from. If you're planning to use SAML 2.0 or other protocols not supported by the old WAYF model, you may also need to provide metadata about your SPs to enable the DS to safely interact with the SP.

Installing an Centralized DS

- **Which Protocol?**
  - The Discovery Service will automatically handle both the legacy Shibboleth AuthnRequest message (so-called "WAYF mode") and the full Discovery Service Protocol.
  - No explicit configuration is required to select the right protocol.

Installing an Centralized DS

The Shibboleth Discovery Service, version 1.2.1, is a standard Java web application.

1. Download and decompress the Discovery Service package from the Shibboleth Download site (http://shibboleth.net/downloads/centralized-discovery-service/latest/shibboleth-discovery-service-1.2.1-bin.zip)
2. Change into the newly created distribution directory
3. Endorse Xerces and Xalan by copying the contents of the endorsed directory to the appropriate place on the web Server (for tomcat this is $TOMCAT_ROOT\common\endorsed).
4. Run either install.sh (on Unix systems) or install.bat (on Windows systems) as a suitably authorized user. This user must have the ability to create the Discovery Service home directory identified in the previous step.

Installing an Centralized DS

- Configure the Discovery Service to point to the metadata sources you identified above:
  - Declaring the metadata sources
    - Every potential source of metadata is declared to the Discovery Service by a <MetadataProvider> element in the $DS_HOME/ wayfconfig.xml file.
    - If the wayfconfig.xml file is changed, you need to restart the service

Installing an Centralized DS

- The <MetadataProvider> element must have the following attributes
  - identifier - This is the name by which the metadata identified
  - displayName - This is the name displayed for the metadata group.
  - url - This is the location of the metadata. Three URL forms are supported file:<LocalPath>, http://<RemotePath> and https://<RemotePath>

Installing an Centralized DS

- In addition, if the http:// or https://<RemotePath> is used, the following attributes may be specified:
  - backingFile - This is where the remote file will be stored. This is required for http:// and https:// urls.
  - timeout - This is the timeout for the http connection, this is deprecated starting in V1.2 and replaced by requestTimeout.
  - disregardSslCertificate (added in v1.2) - boolean flag indicating whether the servers SSL certificate should always be accepted regardless of whether its valid (defaults value: false)
  - requestTimeout - (added in v1.2) Maximum length of time to wait for the remote server to finish its response given in XML duration notation (default value: PT5S).
  - proxyHost (added in v1.2) - hostname of the HTTP proxy to use when fetching metadata
  - proxyPort (added in v1.2) - port of the HTTP proxy to use when fetching metadata
  - proxyUser (added in v1.2) - username used when connecting to the HTTP proxy to use when fetching metadata
  - proxyPassword (added in v1.2) - password used when connecting to the HTTP proxy to use when fetching metadata
  - basicAuthUser (added in v1.2) - HTTP BASIC authentication username used when connecting to the HTTP proxy to use when fetching metadata
  - basicAuthPassword (added in v1.2) - HTTP BASIC authentication password used when connecting to the HTTP proxy to use when fetching metadata

Installing an Centralized DS

- In all cases, for releases starting V1.2.0, the following attributes to perform extra data validation, or the reload frequency,
  - certicateFile - If specified, this is the path to a certificate file. This certificate is used to validate the signature on the root element of the incoming metadata. The filter will prevent loading of the metadata if it fails validation or if there is no certificate present.
  - maxValidityInterval - If specified, this value is used to ensure that the metadata contains a validUntil attribute on the root of the metadata. This ensures that old metadata, which may contain entities which have been removed/revoked, is not used. If the value is "0" then it specifies the interval, in seconds, from now within which the validUntil date must fall. A value of zero indicates no upper limit.
  - refreshDelayFactor (added in v1.2) - an number between 0.0 and 1.0, exclusive, used to determine the next metadata refresh cycle based on the current metadata's cache expiration time (default value: 0.75), see the IdP Documentation for more details.
  - minRefreshDelay (added in v1.2) - the minimum interval between successive metadata refresh operations given in XML duration notation (default value: PT5M), see the IdP Documentation for more details.
  - maxRefreshDelay (added in v1.2) - the maximum interval between successive metadata refresh cycles given in XML duration notation (default value: PT4H), see the IdP Documentation for more details.

Installing an Centralized DS

- Example Metadata Declarations

```
<MetadataProvider
     displayName="Local Federation"
     identifier="FileFed"
     url="file:///etc/DiscoveryService/metadata/sites.xml"/>

<MetadataProvider
     displayName="UK Federation"
     identifier="UkFed"
     certicateFile="/etc/metadata/ukfederation.pem"
     maxValidityInterval = "P7D"
     backingFile="/etc/metadata/ukfed_store.xml"
     url="http://metadata.ukfederation.org.uk/ukfederation-metadata.xml"/>
```

Obs: In all cases, the Discovery Service will reload metadata as soon as it has been changed. It is **not** necessary to restart the service

Installing an Centralized DS

- If the metadata contains <DiscoveryResponse> elements, then the binding attribute is checked.

- If an entity has an invalid binding then it is removed from the metadata and a note written to the log.

- If required the behavior can be limited to issuing a warning bu setting the element "warnOnBadBinding" in the <Default> configuration to be "true".

Installing an Centralized DS

- Using the metadata source.
  - Once a metadata source has been declared, it is associated with a specific location via the <DiscoveryServiceHandler> element.

- Example Discovery Service declarations

```
<DiscoveryServiceHandler [...]>
  <Federation identifier="UkFed"/>
  [...]
</DiscoveryServiceHandler>
```

Installing an Centralized DS

- **Filtering Metadata**
  - A <MetadataProvider> may have one or more custom filters added (written in Java). Each filter has to implement org.opensaml.saml2.metadata.provider.MetadataFilter and have a constructor which take a single parameter of type org.w3c.dom.Element (this being the element which defines the filter as described below).
  - A filter is associated with a Metadata Provider via a <Filter> element. This is unstructured. It may have any attributes and sub elements which can be used to provide parameters to the code. It must have the following attributes:

- identifier - An unique identifier for the filter
- type - The class name for the filter.

Installing an Centralized DS

- Example Metadata Filters declaration

```
<MetadataProvider [...]>
  <Filter identifier="Filter1"
    type="uk.ac.ed.sdss.FilterForStuff">
    <MoreSpecificStuff
       param="wibble"
    />
  </Filter>
  <Filter identifier="Filter2"
    type="edu.internet2.OtherFilter">
    <Stuff>
      <EvenMoreStuff/>
    </Stuff>
  </Filter>
</MetadataProvider>
```

Installing an Centralized DS

- **White and BlackList**
  - The DiscoveryService is shipped with a simple white-list and black-list filter. Given a list of entities, the metadata will be adjusted to remove all elements which are **not** in the list (white list operation) or to remove all entities which **are** on the list (blacklist operation).

Installing an Centralized DS

- Example Black List filter

```
<Filter identifier="Black"
        type="edu.internet2.middleware.shibboleth.wayf.plugins.provider.ListFilter"
        excludeEntries="true">
  <EntityId>https://first.blacklisted.entity.edu/IdP</EntityId>
  <EntityId>https://another.blacklisted.entity.edu/IdP</EntityId>
</Filter>
```

- The excludeEntries controls whether elements on the list are excluded from the metadata (blacklist operation) or have to be included (white list operation).

Installing an Centralized DS

- Important:
  - The resulting metadata must include all SPs which interact with the DS. This is particularly important to remember when buidling white list (excludeEntries="false") filters

Installing an Centralized DS

- After all, deploy the Discovery Service .WAR file, located in the Discovery Service's Home directory.

- Further configuration is described at :

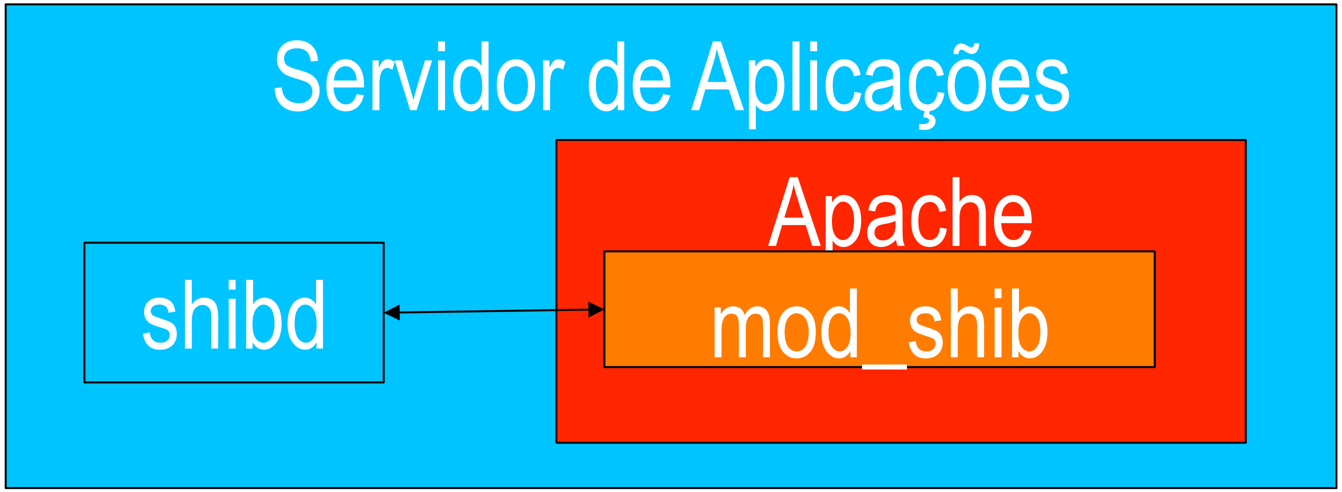
https://wiki.shibboleth.net/confluence/display/SHIB2/DSConfiguration

Building Fedarated Applications

- Overview
  - mod_shib
  - Protecting applications
- Attribute Recovery
  - Atributes and mapping
  - Authorization by application
- Constructing the users objetcs
  - JAVA
  - PHP

mod_shib Module

- Part of Shibboleth architecture is embbeded to the web server
- In case of Apache web server, this occurs through the mod_shib module
- Configuration like the mod_auth_basic e mod_ssl modules



mod_shib Module

- Can be protected elements of <Files>, <Directory> ou <Location> types through the apache configuration file or a *.htaccess* file

```
<Location /diretorio>
        AuthType shibboleth
        ShibRequireSession On
        ShibRequireAll On
        require mail ~.*@ufmg.br$
</Location>
```

mod_shib Options

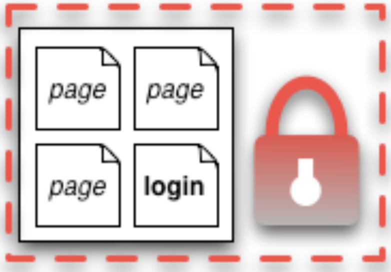
- Key policies
  - require
    - *require valid-user*
    - *require shibboleth*
  - ShibRequireAll on/off
  - ShibRequireSession on/off
  - ShibUseHeaders on/off

Protecting applications

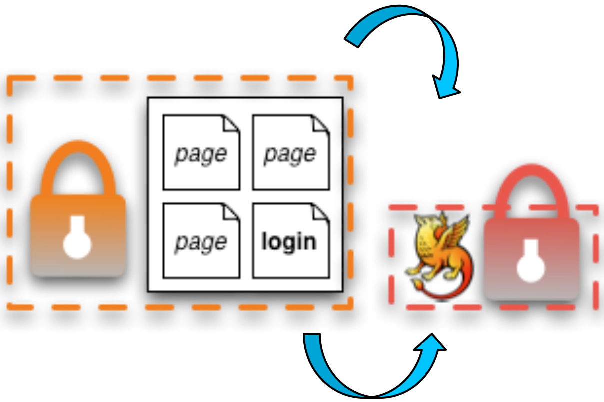
- The way an application is federalized depends on some characteristics:
  - Availability of source code
  - Public application with privileges to authenticated users
  - Needs of user data in the application

Protecting applications

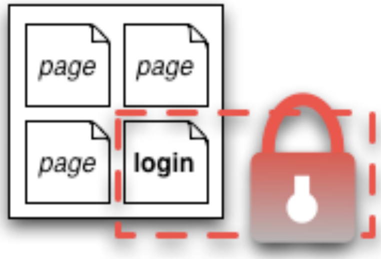
- All the application



- Application with the *lazy sessions mechanism*



- Only the pagy that creates a session in the application



Prottecting all the application

- A simple solution
  - All the required access must be made by the Shibboleth authentication
  - Handicap: there is no one part of the application acessible without authentication

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

Prottecting all the application

- This method allows authentication only actuate when necessary

```
<Location /aplicacao/login.php>
    AuthType shibboleth
    ShibRequireSession On
    require valid-user
</Location>
```

Lazy Sessions

- Protection by rules URL resource based
- Session checked by the environment variables
- Explicit redirect to SessionInitiator
- The access control stays at application own
- Url :

https://servidor/Shibboleth.sso/Login?target:"http://servidor/applicacao"

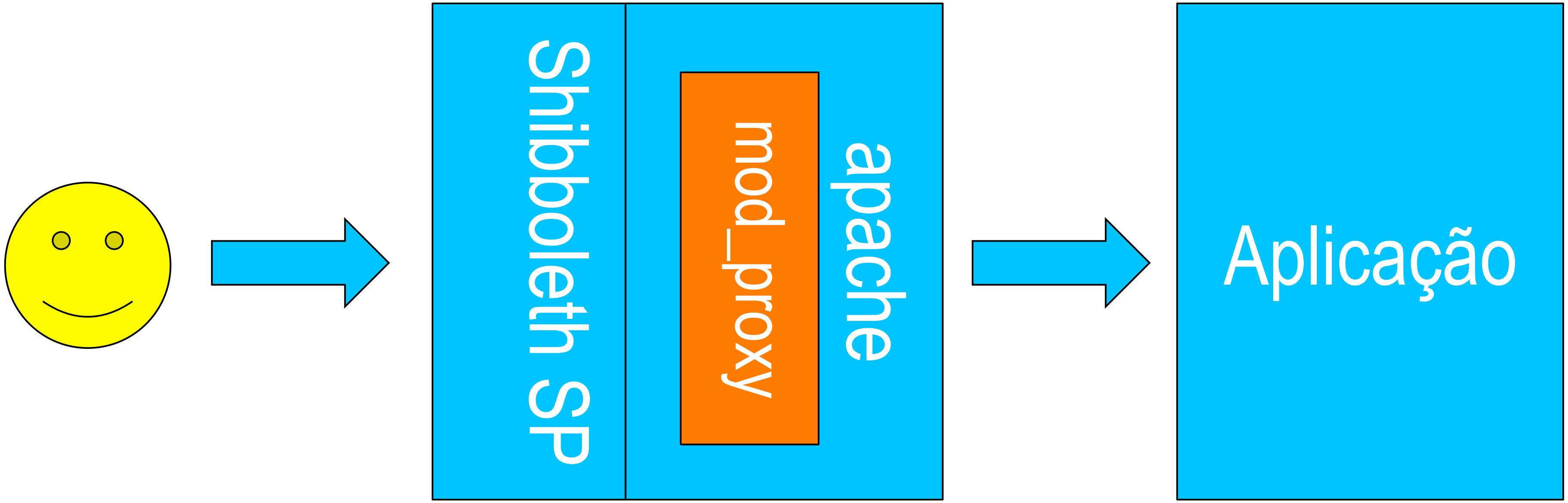
Lazy Sessions

- Used in:
  - Applications wich have features that don’t need of na user session all the time
  - Applications tha can support several authentication mechanisms

```
<Location /aplicacao>
         AuthType shibboleth
         ShibRequireSession Off
         require shibboleth
</Location>
```

Reverse Proxy

- Only viable option to a propietary application where the authentication method couldn't be adapted to Shibboleth



Reverse Proxy

- Only viable option to a propietary application where the authentication method couldn’t be adapted to Shibboleth

```
ProxyPass /app  http://Servidor/app
<Location /app>
         AuthType shibboleth
         ShibRequireSession ON
         require shibboleth
</Location>
```

Atributes and mappings

- Atributes
  - Represent information about the user, like Name, E-mail, Afiliattion, etc
  - Can be provided by SP to the application through environment variables or HTTP headers
  - The applications can use themto perform an user authorization

Atributes and mappings

- Mappings:
  - Define the header’s name or web environment variable web for each attribute
  - Was configurated in the attribute-map.xml SP file

Example of a mapping for the “cn” attribute:

```
<Attribute name="urn:mace:dir:attribute-def:cn" id="ShibCafe-Person-cn"/>
<Attribute name="urn:oid:2.5.4.3" id="ShibCafe-Person-cn"/>
```

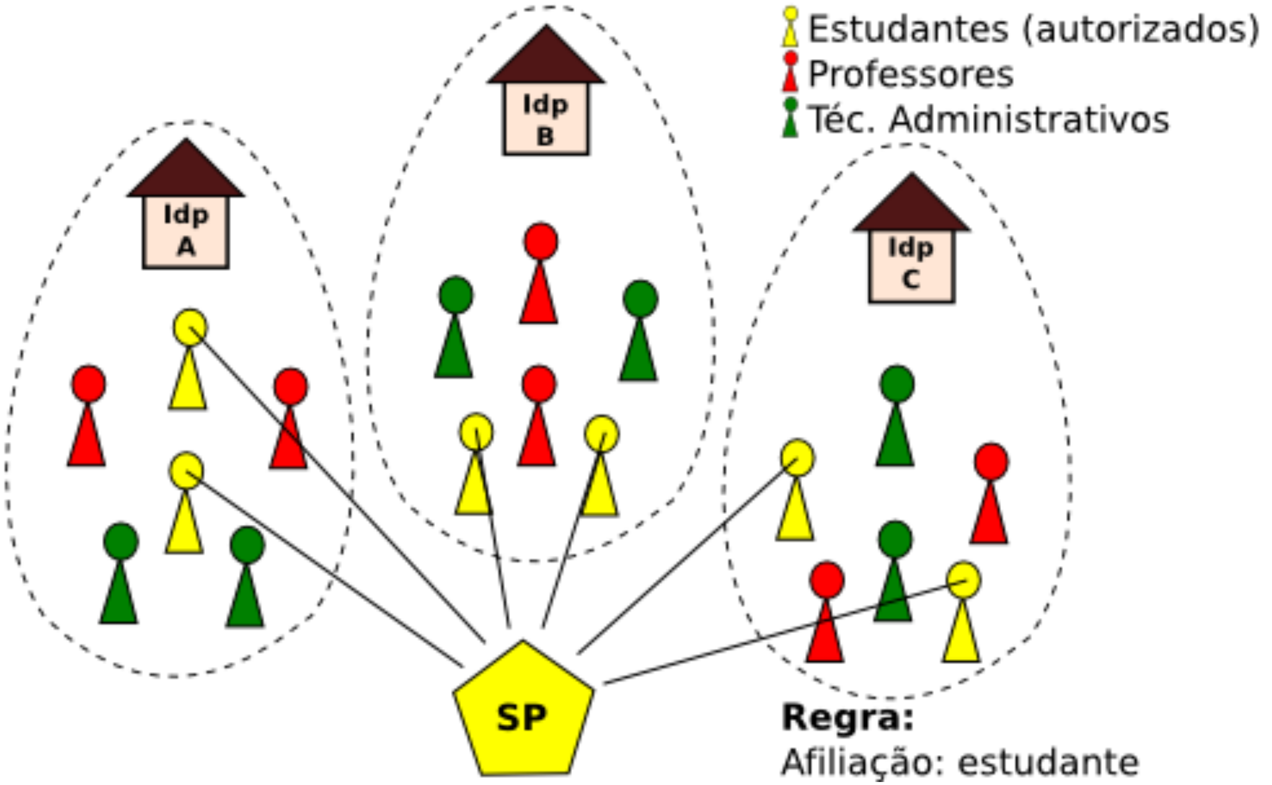
Authorization by application

- The Shibboleth structure autenticathes the user and provide the atributes to application
- The application is responsible for authorization
- Neither application, neither HTTP server manipulate the user's password
- Theres is a need of common/compatible atributes between the IDPs for a global authorization

Authorization by application

- Define atributes needed for authorization and advise institutions

Fonte: Adapted from Switch AAI - Federation



Estudantes (autorizados)
Professores
Téc. Administrativos

Regra:
Afiliação: estudante

Figura 3.1

Authorization by application

- Attributes provided by brEduPerson scheme that may be used for authorization
  - brEduAffiliationType
    - Relationship type with the institution
  - brEntranceDate
    - Relationship start date
  - brExitDate
    - Relationship end date

Authorization by application

- The authorization can also be assisted by a relational database providing
  - Greater flexibility
    - Freedom in setting the rules of Authorization
  - Integration Facility
    - Easier relationship with other bases
  - Maintenance request
    - The rules must be created and maintained by the application

Configuration in containers JEE

- For Java Web applications using Tomcat and Apache 2
  - Installing the module mod_proxy_ajp
  - Redirect requests to from Apache HTTPD to Tomcat
    - ProxyPass /minappp ajp://servidor:8009/appjee
  - Protect the context for user session creation

```
<Location /minhaapp>
   AuthType Shibboleth
   ShibRequireSession On
   require valid-user
</Location>
```

Configuration in containers JEE

- Recovery via attribute request

uid = request.getAttribute(“ShibCafe-InetOrgPerson-Uid");

- Recovery via HTTP header

uid = request.getHeader(“HTTP_SHIBCAFE_INETORGPERSON_UID”);

PHP Configuration

- PHP natively integrates Apache
- It is not necessary to use proxy
- resource protection is direct through Location directive

```
if (isset($_SERVER['HTTP_SHIBCAFE_INETORGPESON_UID']) and
                                                  !empty($_SERVER['HTTP_SHIBCAFE_INETORGPESON_UID'])){
   $uid= $_SERVER['HTTP_SHIBCAFE_INETORGPESON_UID'];
   $uid= utf8_decode($uid);
   // verificação de regras e outros processamentos...
}
else{
   // Erro: attribute não encontrado!
}
```

Configuration for other programming languages

- Perl

```
my $uid = $ENV{'SHIBCAFE_INETORGPERSON_UID'}
```

- Python

```
uid = environ.get('SHIBCAFE_INETORGPERSON_UID');
```

- ASP

```
Set uid = Request("HTTP_SHIB_IDENTITY_PROVIDER");
```

Authorization Implementation

- Authorization for relationship
  - Map types of link with a specific level of authorization
  - Identify the identity links
    - Check the value of the attribute brEduAffiliationType
  - Perform Authorization
    - Compare the page level Authorization (resource) with the levels of identity links

Interfederation

Interconnecting national federations
eduGAIN → Interfederation, eduroam → Confederation

- No longer a single legal or policy framework
- Each federation has its own eduGAIN has one as well

- Different sets of attributes used internationally

- No single 'interfederation helpdesk' in case of problems
- Debugging involves probably more parties
- Involved parties will generally know less about each other

eduroam

eduGAIN

Interfederations - eduGAIN

- ***G***lobal ***A***uthentication ***IN***frastructure for **edu**cation
- An interfederation service ***primarily*** for Research & Education
- Connects existing SAML-based academic identity **federations**
- Developed and funded by European GÉANT projects (www.geant.net) but open also to non-European federations

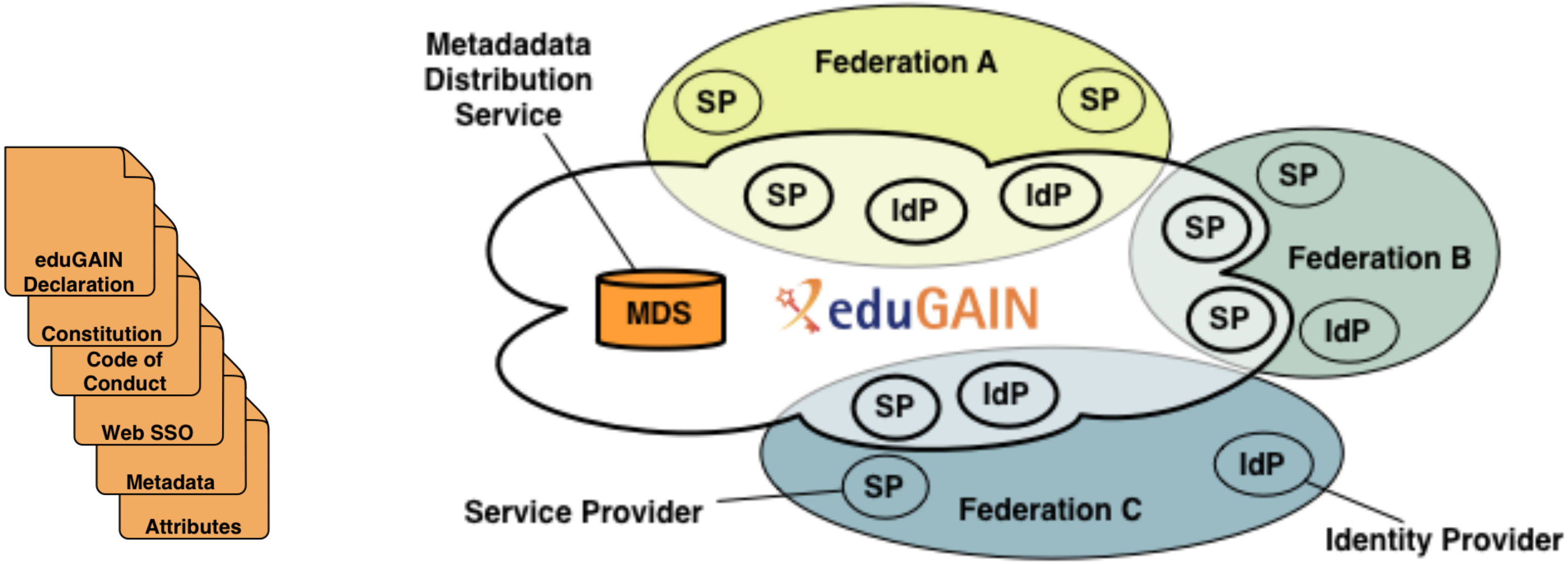
eduGAIN Example

eduGAIN provides policy framework and standards to build trust

SPs and IdPs of participating federations **opt-in** for eduGAIN

- Various local processes for what this means
- Opt out being piloted by some

MDS fetches, aggregates and republishes metadata



Metadata Exchange for eduGAIN

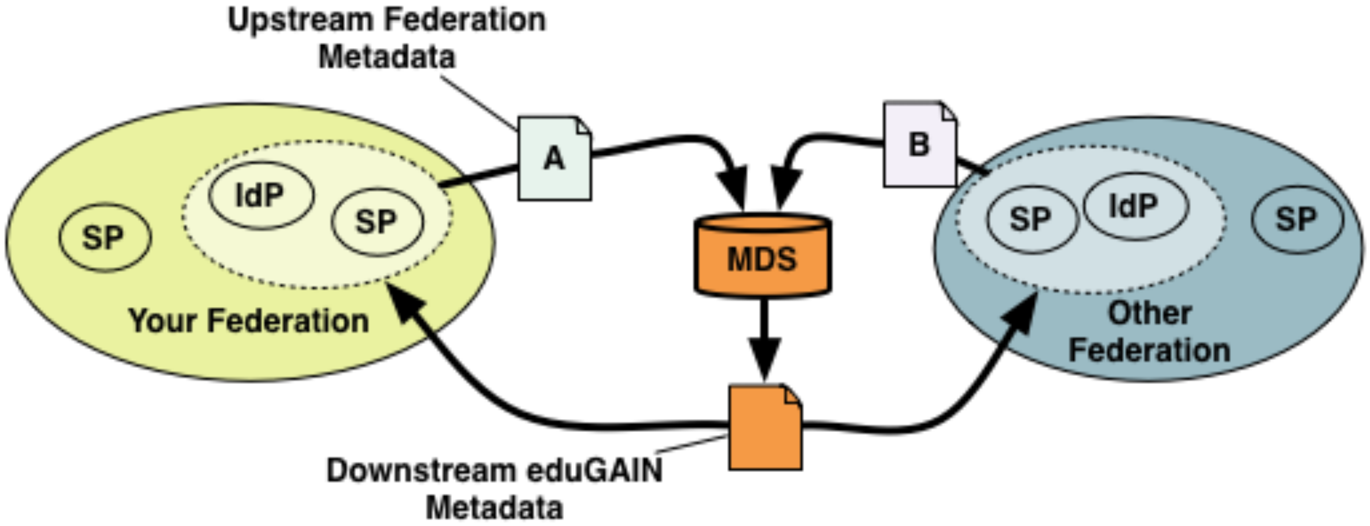
Magic

Middleware for collaborative Applications and Global virtual Communities

Each Federation publishes a Metadata file with the entities that want to interfederate.

The eduGAIN Metadata Data Service fetches them

eduGAIN MDS aggregates all metadata and republishes it



Federations fetch it and filter-out their own entities

Entities consume the filtered eduGAIN metadata file
in addition to the one from the federation

eduGAIN Constitution and Policy

https://technical.edugain.org/documents

Governance and Governing Bodies

- eduGAIN Executive Committee (**eEC**)
- eduGAIN Steering Group (**eSG**)
- Operational Team (**OT**)

Participant Federations MUST

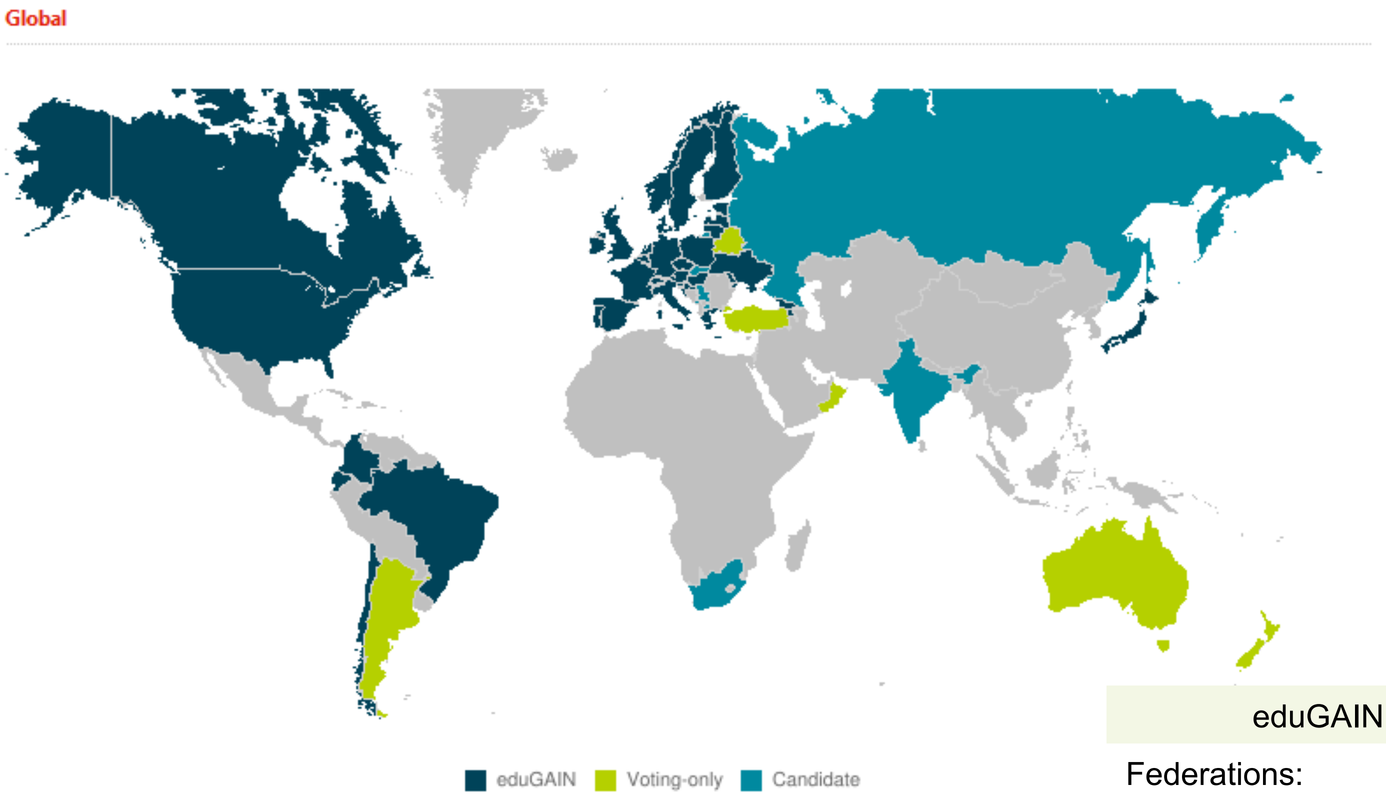
- Primarily serve the interests of the education and research sector
- Provide a point of contact for their Members for dealing with technical issues.
- Provide processes for handling complaints and incidents involving their Members.
- Have a published Metadata registration practice statement.
- Follow the eduGAIN SAML 2.0 Metadata Profile

No express right of communication

- For an Entity registered in an eduGAIN Participant Federation it does not imply any right of communication with any other Entity exchanged through eduGAIN.

eduGAIN status*

- https://technical.edugain.org/status



eduGAIN numbers	
Federations:	37
All entities:	3.099
IdPs:	1.993
SPs:	1.107
Standalone AAs:	3

* Feb, 16th, 2016

Interfederation Use Cases

- International research projects. Researchers from different countries and institutions can cooperate on a project.
- Access to the global services like FileSender or cloud infrastructures
- Sharing e-Learning materials/courses easily

Benefits - Identity Providers

- Offer more to your users - enables access to a wider range of services than are available locally or nationally

- No extra administrative burden - if you are already participating in a federation with Web Single Sign On set up

Benefits – Service Providers

- Grow your audience - offer services to a greater number of users
- Lower costs per user - your audience grows without increasing the demand for passwords and user support

Benefits - Federation

- More services for your users – enables them to access services from different federations
- Lower administration costs – thanks to easier technical integration
- Saves time - no need to make bilateral agreements with other federations
- Trustworthy - secure collaboration and exchange of information

Benefits - Users

- Access a wider range of services than are available nationally or locally.
- One digital identity and password for all services connected through eduGAIN
- eduGAIN is 'invisible' to you so you can access services without extra effort

How to connect

- Go trough the joining check list at:
  - https://technical.edugain.org/joining_checklist.php
- Have a contact email
- Signing declaration
- Metadata source and signing certificate
- Governance delegate and deputy
- Link to the federation web page
- Link to the federation policy
- Link to the metadata registration practice statement
- Contact eduGAIN OT at edugain-ot@geant.net

Quiz Time

Quiz Time

1. **Which of these requirements can eduGAIN most address:**
   a) Security incident response
   b) Attributes that cross national borders
   c) User friendliness and ease of use
   d) Authorisation under community control

2. **What is the difference between an Embedded Discovery Service and a Centralized Discovery Service?**

3. **What is the main goal of a service provider?**

4. **True or False? Participating eduGAIN members must**
   a) Primarily serve the interests of the education and research sector.
   b) Provide a minimum standard attribute release between entities
   c) Get approval of the eduGAIN SG for commercial entities in eduGAIN
   d) Provide processes for handling complaints and incidents involving their Members.

Quiz Time

5. **Which of the following is an option to a propietary application where the authentication method couldn’t be adapted to Shibboleth?**
    a) Lazy Sessions
    b) Reverse Proxy
    c) Tunnelling
    d) Hooking

6. **Which of the following is NOT a SAML component?**
    a) Assertions
    b) Bindings
    c) Profile
    d) PHP

Sources

- **Service Providers Module**
  - Service Providers
    (https://wiki.refeds.org/dosearchsite.action?queryString=service+provider)
    (https://shibboleth.net/products/service-provider.html)
    (https://wiki.shibboleth.net/confluence)
    (https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#overview)
  - Interfederation
    (www.edugain.org)