Integration of Shibboleth and (Web) Applications

MPG-AAI – Workshop

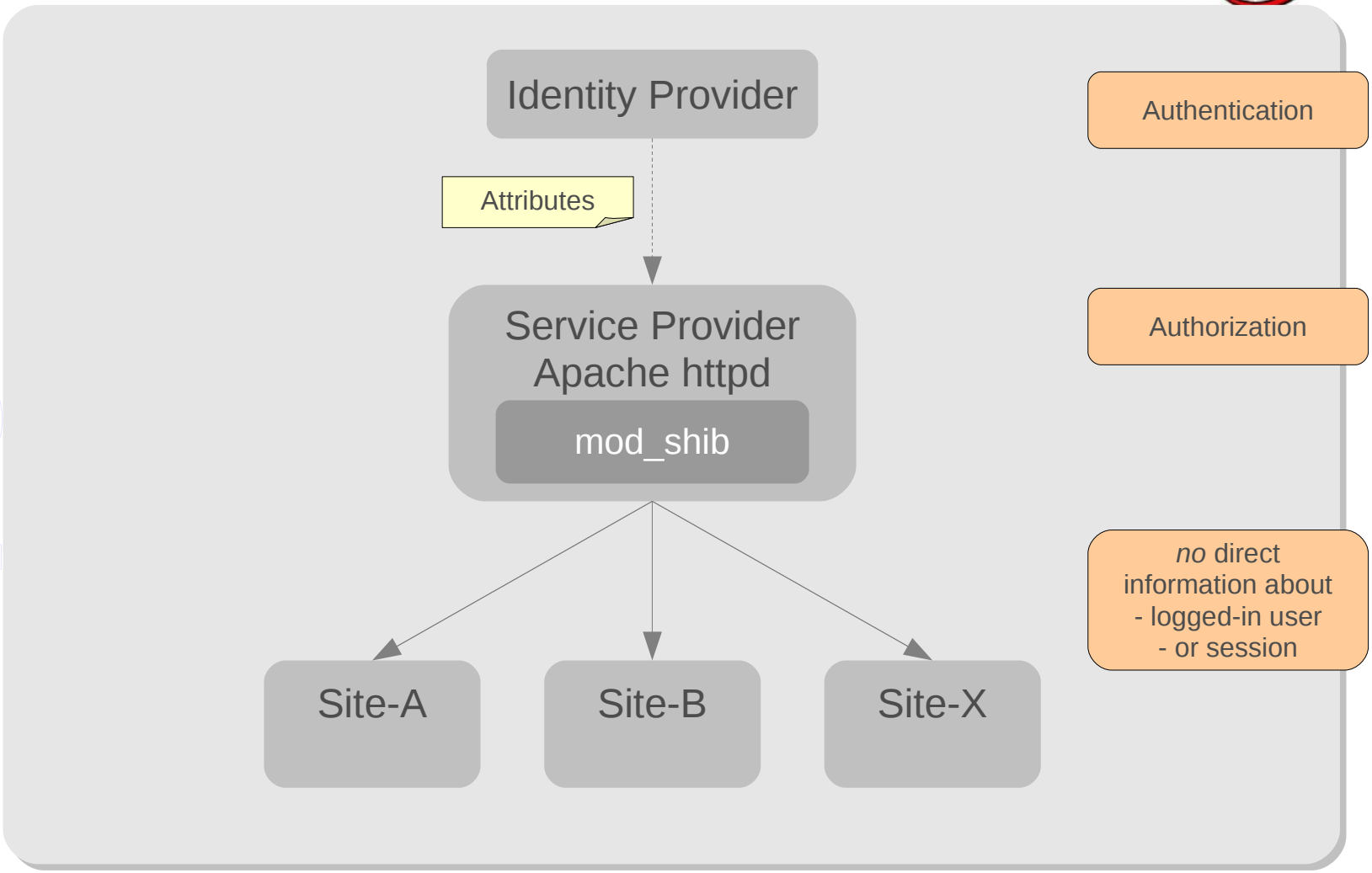
Clarin Centers Prague 2009

2009-11-06

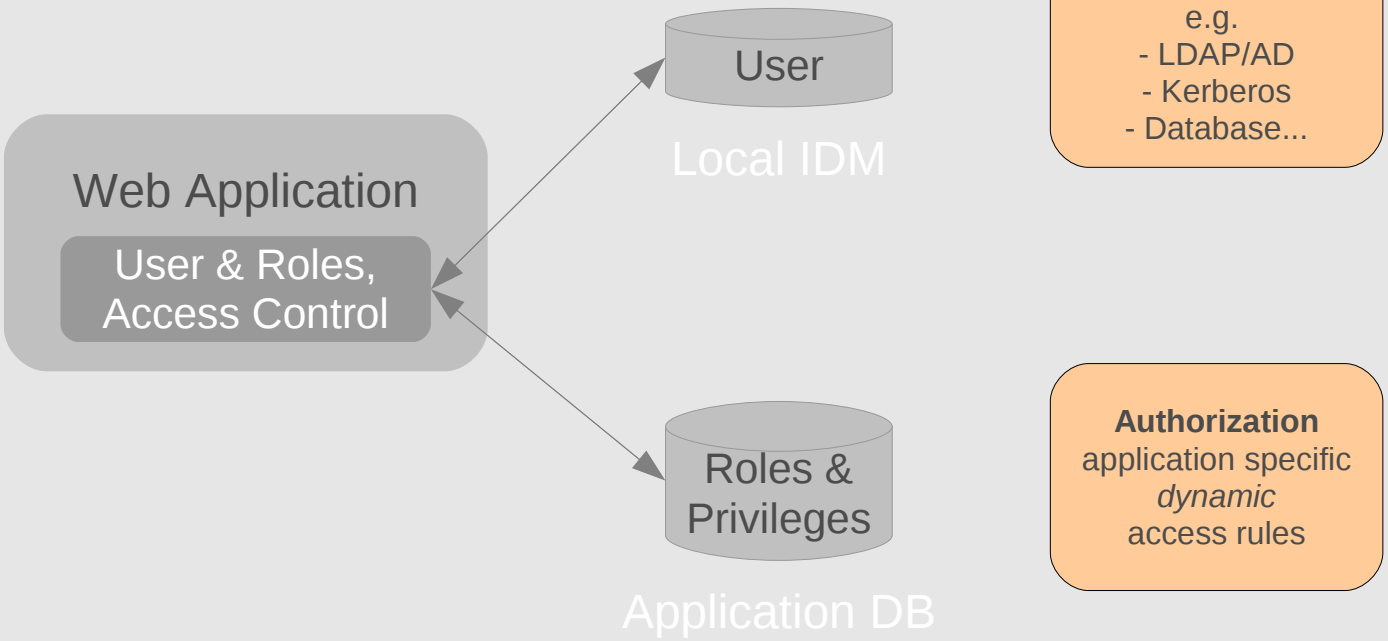
(Web) Application Protection Models

- Classical – Application behind Shibboleth Standard Session
- Application with Shibboleth Lazy Session / Passive Mode like
  - shhaa-filter
  - Mediawiki Plugin
- Application as Own Service Provider like
  - simpleSAMLphp
  - OIOSAML (java)
  - Guanxi (java)

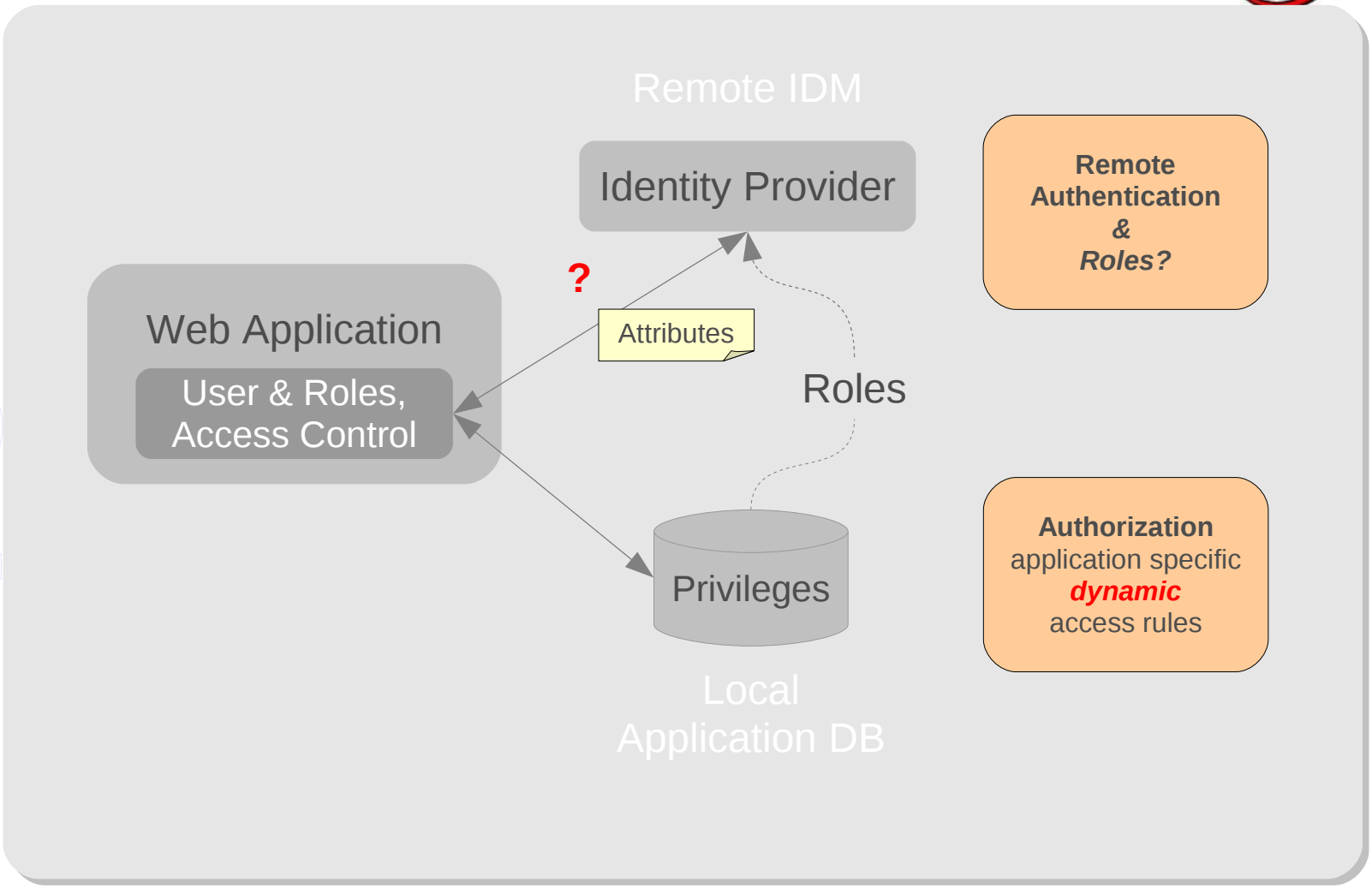
Classical – Standard Shibboleth Session



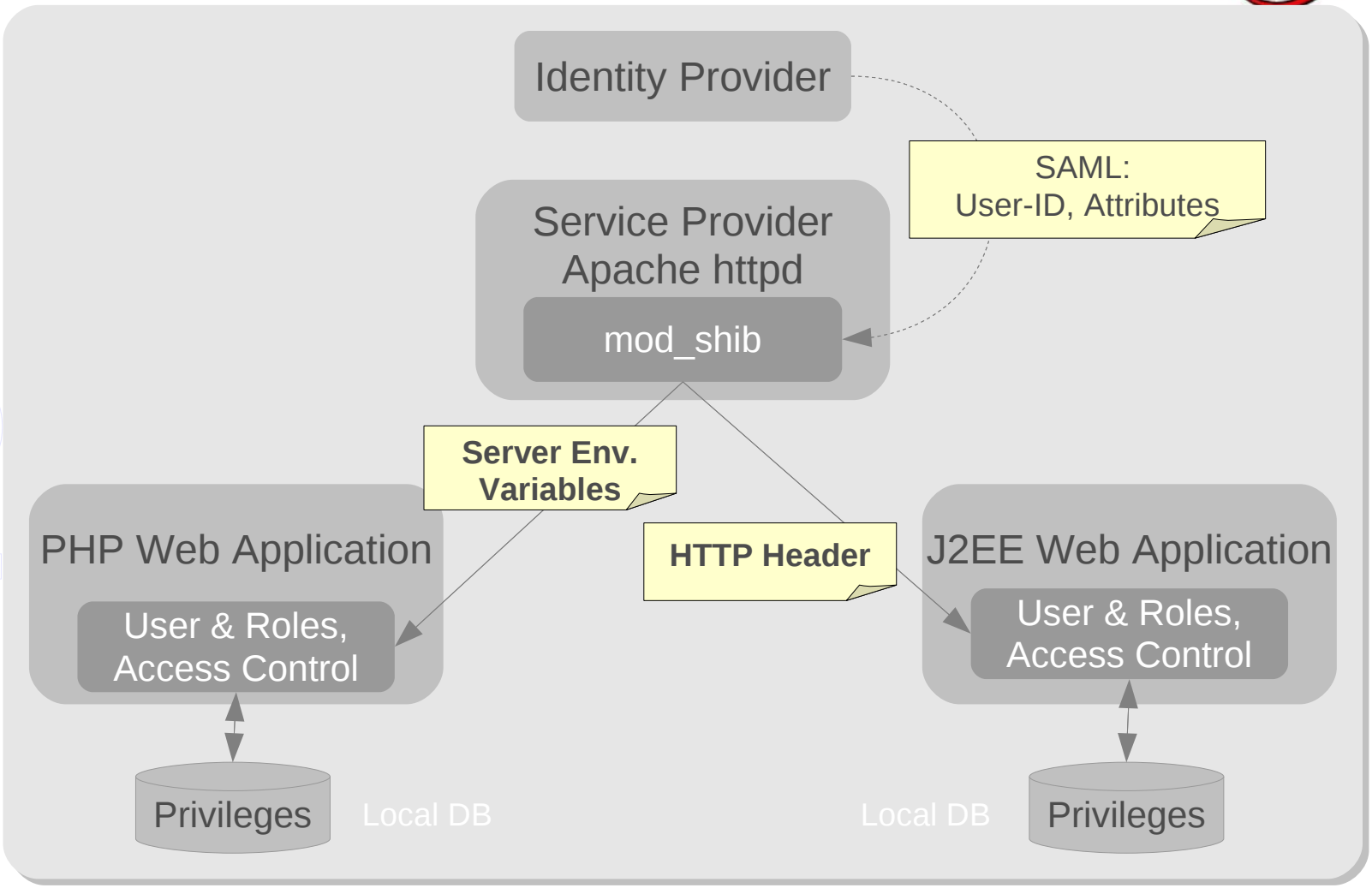
Classical Web Application Authorization



Shibbolized Web Application



Attributes Propagation



Shibboleth “Lazy Session” / “Passive Mode”

- Shibboleth SP (Apache Httpd) Configuration
  https://spaces.internet2.edu/display/SHIB2/NativeSPApacheConfig

```
<Location /secure/passive >

     AuthType            shibboleth
     ShibRequireSession  Off


     ShibUseEnvironment  On

     ShibUseHeaders      On


     Require             shibboleth

</Location>
```

use Shibboleth, don't enforce login

set attributes to
- server environment variables
- http headers

Shibboleth enabled, but does not block

Demo

Shibboleth “Lazy Session” / “Passive Mode”

- Shibboleth is enabled as in Standard (Shib.) Session only:
- no Login *enforced !*
  that means:
- User logged-in:
  - Attributes are retrieved and propagated as usual
- User *not* logged-in:
  - Attributes are still “propagated”, *but are empty / hold no values*
  - as soon as a Shib. Session (Login) exists => Attributes filled

MediaWiki – Shibboleth Extension

- http://www.mediawiki.org/wiki/Extension:Shibboleth_Authentication
- move code to <mediawiki_home>/extensions/ShibAuthPlugin.php
- edit LocalSettings.php

```
### SHIBBOLETH INTEGRATION
require_once('extensions/ShibAuthPlugin.php');
$shib_WAYF = "Login";
$shib_WAYFStyle = "DS";
$shib_Https = true;
$shib_LoginHint = "login via mpgaai";
$shib_AssertionConsumerServiceURL = "/Shibboleth2.sso";
$shib_RN = ucfirst(strtolower($_SERVER['sn']));
$shib_email = strtolower($_SERVER['mail']);
$wgHooks['ShibUpdateUser'][] = 'ShibUpdateTheUser';
function ShibUpdateTheUser($existing, &$user) {
    global $shib_email;
    global $shib_RN;
    $user->setEmail($shib_email);
    $user->setRealName($shib_RN);
    return true;
}
$shib_UN = str_replace("_","-", $_SERVER['persistentID']);
```

Demo

Demo – Simple Http Header AuthN/Z

- Demo on Language Archive Tools
  (MPI f. Psycholinguistics, Nijmegen)
  - Anonymous Access
  - SSO Login
  - Show Session- & Attributes Information
  - Attribute Based Authorization
  - SSO to Modular Web Application
  - “SLO”, Logout from Local SP Session
  - API to retrieve Session- & User Information (Attributes)
- Integration
  add filter in `web.xml`, set config-file location, use filter for whole web app
- Configuration
  see own configuration file `shhaa.xml`

Enable SHHAA Filter in Web Application

- web.xml

```
<web-app>

    <context-param>
        <param-name>ShhaaConfigLocation</param-name>
        <param-value>/WEB-INF/shhaa.xml</param-value>
    </context-param>

    <!--  filter configs -->
    <filter>
        <filter-name>AAIFilter</filter-name>
        <filter-class>de.mpg.aai.shhaa.AuthFilter</filter-class>
    </filter>

    <!-- filter mappings -->
    <filter-mapping>
        <filter-name>AAIFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

    <!-- context listener -->
    <listener>
        <listener-class>
            de.mpg.aai.shhaa.config.ConfigContextListener
        </listener-class>
    </listener>
```

OIOSAML – Java Service Provider

- www.softwareborsen.dk/projekter/softwarecenter/brugerstyring/oio-saml-java
- Easy Initial Setup & Configuration via GUI
- Web Application Filter – Easy to Integrate in Your Web Application
- API to retrieve Session- & User Information (Attributes)
- Compatible with Shibboleth Federation
  after just slight configuration modifications
- Provides *Own* Discovery Service

Enable OIOSAML Filter in Web Application

- web.xml

```
<context-param>
     <param-name>oiosaml-j.home</param-name>
     <param-value>/opt/mpgaai/oiosaml/conf/</param-value>
 </context-param>
 <listener><listener-class>
     dk.itst.oiosaml.sp.service.session.SessionDestroyListener
 </listener-class></listener>

  <servlet>
      <servlet-name>SAMLDispatcherServlet</servlet-name>
      <servlet-class>
              dk.itst.oiosaml.sp.service.DispatcherServlet
      </servlet-class>
      <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
      <servlet-name>SAMLDispatcherServlet</servlet-name>
      <url-pattern>/saml/*</url-pattern>
  </servlet-mapping>

 <filter>
     <filter-name>LoginFilter</filter-name>
     <filter-class>dk.itst.oiosaml.sp.service.SPFilter</filter-class>
 </filter>
 <filter-mapping>
     <filter-name>LoginFilter</filter-name>
     <url-pattern>/sp/*</url-pattern>
 </filter-mapping>
```

Configure OIOSaml.java

This page allows you to configure OIOSaml.java for your system. For security reasons, this configuration can be run only once. If possible, the configuration files will be written automatically to '/opt/mpgaai/oiosaml/conf/' (oiosaml-j.home).

Entity ID

https://saml.idp.rzg.mpg.de

Protocol locations:

These are autodiscovered based on the url you've used for this page. Make sure you're using the official url.

Receive SAML Artifact response	https://idp.rzg.mpg.de/oiodemo/saml/SAMLAssertionConsumer
Receive SAML POST response	https://idp.rzg.mpg.de/oiodemo/saml/SAMLAssertionConsumer
Initiate single logout	https://idp.rzg.mpg.de/oiodemo/saml/SAMLAssertionConsumer
Receive single logout response	https://idp.rzg.mpg.de/oiodemo/saml/LogoutServiceHTTPRedirectResponse
Receive single logout request	https://idp.rzg.mpg.de/oiodemo/saml/LogoutServiceHTTPRedirect
Receive SOAP single logout request	https://idp.rzg.mpg.de/oiodemo/saml/LogoutServiceSOAP

Identity provider metadata

IdP metadata file

Browse...

Configuration and metadata for this service provider

Keystore with private key for signing requests and responses. Either a pkcs12 file or a Java keystore.

Browse...

☐ Create new self-signed keystore? (only for testing, and only when not uploading a keystore above)

Keystore password	
Organisation Name	
Organisation URL	
Technical email contact address	
Enable Artifact consumer?	☐
Enable Redirect consumer?	☐
Enable SOAP Single Logout?	☐
Support OCES Attribute Profile?	☐

Even if this is not checked, any attributes will be accepted, but no AttributeConsumer will be added to the SP metadata.

Configure system

OIOSAML with Shibboleth Federation

- www.gmjjavadesigns.com/gmjd/entry/how_to_integrate_oiosaml_java
- access initial configuration site via “production” URL
  => URL is used for metadata generation
- Metadata: OIOSAML uses separate files for each IdP
- on import Shibboleth IdP metadata:
  add missing XML namespace information
  (some prefixes not recognized)
- turn off Name-ID encryption at Shibboleth IdP -
  in relying-party.xml change Profile config to:

```
<ProfileConfiguration xsi:type="saml:SAML2SSOProfile"
    encryptNameIds="never"
    ...
```

- modify OIOSAML default config - set in oiosaml-sp.properties:

```
oiosaml-sp.assurancelevel=0
oiosaml-sp.nameid.allowcreate = false
oiosaml-sp.nameid.policy = transient
```

Thanks & Discussion

Questions, Discussion...

- Thank You for Your Attention -