



Enabling SAML 2.0 in a wiki

Anders Lund (UNINETT)
Andreas Åkre Solberg (UNINETT)

FEIDE Software used

- Dokuwiki

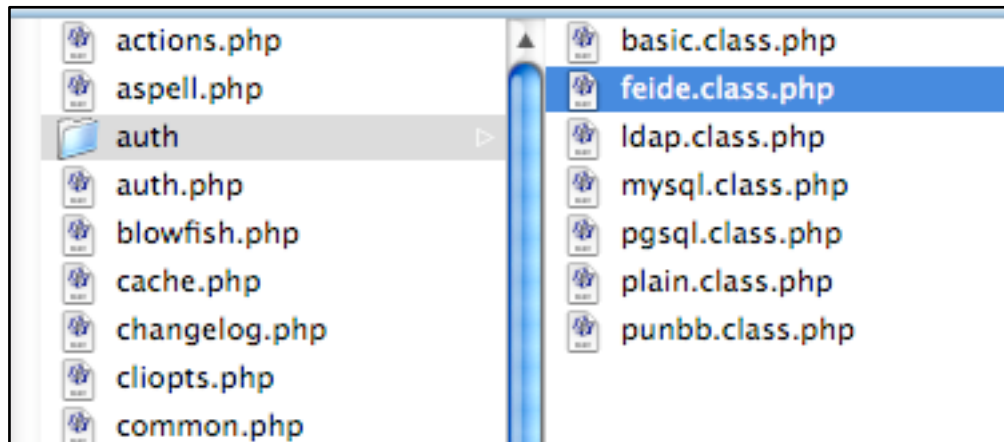
<http://wiki.splitbrain.org/wiki:dokuwiki>

- OpenSSO PHP Extension (lightbulb)

<https://lightbulb.dev.java.net/>

FEIDE Dokuwiki

Pluggable authentication modules



Supports ACL lists, and is using groups for authorization.

FEIDE OpenSSO PHP

A pure PHP5 implementation of a SAML 2.0 SP. Extremely simple installation and configuration.

Implemented as proof of concept.
Not feature-rich.

Opensourced from Sun, modified by Feide.



OpenSSO Metadata

Feide Meta data

```
$idpMetadata = array( "sam.feide.no" =>  
    array( "SingleSignOnUrl" => "https://sam.feide.no/amserver/SSORedirect/metaAlias/idp",  
          "SingleLogoutUrl" => "https://sam.feide.no/amserver/IDPSloRedirect/metaAlias/idp",  
          "certFingerprint" => "3a:e7:d3:d3:06:ba:57:fd:7f:62:6a:4b:a8:64:b3:4a:53:d9:5d:d0" ) );
```

Service Meta data

```
$spMetadata = array( "/sp" =>  
    array(  
        "assertionConsumerServiceURL" => "https://www.ufisa.uninett.no/lightbulb/AssertionConsumerService.php",  
        "issuer" => "www.ufisa.uninett.no",  
        "spNameQualifier" => "http://www.ufisa.uninett.no" ) );
```

OpenSSO meta data is in a simple format, less verbose than standard SAML 2.0 meta data format. Most importantly: endpoints urls, entity id and cert.-info.



Loading Metadata at Feide

SAML 2.0 Meta data for service

```
<EntityDescriptor
  xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
  entityID="www.ufisa.uninett.no">
  <SPSSODescriptor
    AuthnRequestsSigned="false"
    WantAssertionsSigned="false"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://www.ufisa.uninett.no/lightbulb/SingleLogoutService.php"
    />
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
    <AssertionConsumerService isDefault="true" index="0"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://www.ufisa.uninett.no/lightbulb/AssertionConsumerService.php"/>
  </SPSSODescriptor>
</EntityDescriptor>
```

Contains the same info in standard SAML 2.0 meta data format.



Implementing an authentication module

A dokuwiki authentication module identifies whether the user is logged in or not and returns either `true` or `false`. If `true` it associates the authenticated user with a list of groups the user is member of, and also sets a username and a mail address.





Implementing an authentication module

Load OpenSSO.php in the authmodule:

```
// Loading SAML library
require_once('../openssophp/config/config.php');
require_once('../openssophp/lib/saml-lib.php');

require_once('../openssophp/spi/sessionhandling/' . $LIGHTBULB_CONFIG['spi-sessionhandling'] . '.php');
```

Set the OpenSSO SSOinit and logout URL in a variable

```
// URL to return user to after authentication. Will be this page :D
$return_url = self::URL();

// URL initiating SSO with lightbulb, contains some configuration parameters.
$ssoinit_url = $LIGHTBULB_CONFIG['baseurl'] . "spSSOInit.php?" .
    "RelayState=" . urlencode($return_url);

// Logout URL. Also a lightbulb service with some parameters and a return url.
$log_out_url = $LIGHTBULB_CONFIG['baseurl'] . "spSLOInit.php?" .
    "RelayState=" . urlencode($return_url);
```



Implementing an authentication module

Redirect to OpenSSO SSOinit URL if local session cookie does not exist.

```
if (!isset($spi_sessionhandling_getUserID() )) {  
    header("Location: " . $ssoinit_url, true);  
    exit;  
}
```

When a user does not have a local session at the service, she is redirected to the Feide IdP with SAML 2.0 authentication request (this is done by OpenSSO php). After successful authentication the user is sent back to OpenSSO php with a response, and the OpenSSO php library will set a session cookie for you.

When a user is authenticated, you can get a userid through a OpenSSO method:

```
$userid = spi_sessionhandling_getUserID();
```



Dynamic group membership

Retrieve attributes from OpenSSO php

```
$token = spi_sessionhandling_getResponse();  
$attributes = getAttributes($token);
```

Generate dynamic group membership based on attributes:

```
$decomposedID = explode("@", $attributes['eduPersonPrincipalName']);  
$organization = $decomposedID[1];  
if (isset($organization)) {  
    $groups[] = preg_replace("/^[^a-zA-Z0-9]/", "X", "org-" . $organization);  
}  
  
$affiliations = explode("_|_", $attributes['eduPersonAffiliation']);  
foreach ($affiliations AS $affiliation) {  
    $groups[] = preg_replace("/^[^a-zA-Z0-9]/", "X", "affiliation-" . $affiliation . "-" . $organization);  
}
```

In addition add personal group memberships from a file:

```
include($conf['groupfile']);  
if (isset($customgroups[$user])) {  
    $groups = array_merge($groups, $customgroups[$user]);  
}
```



Returning from the auth module

After retrieving attributes and dynamic group membership generation, we set name, mail and groups readable for dokuwiki internals and **return true**.

```
$USERINFO['name'] = $attributes['eduPersonPrincipalName'];  
$USERINFO['mail'] = $attributes['mail'];  
$USERINFO['grps'] = $groups;  
return true;
```





Access Control List

We configure access control of the wiki, using the dynamic groups.

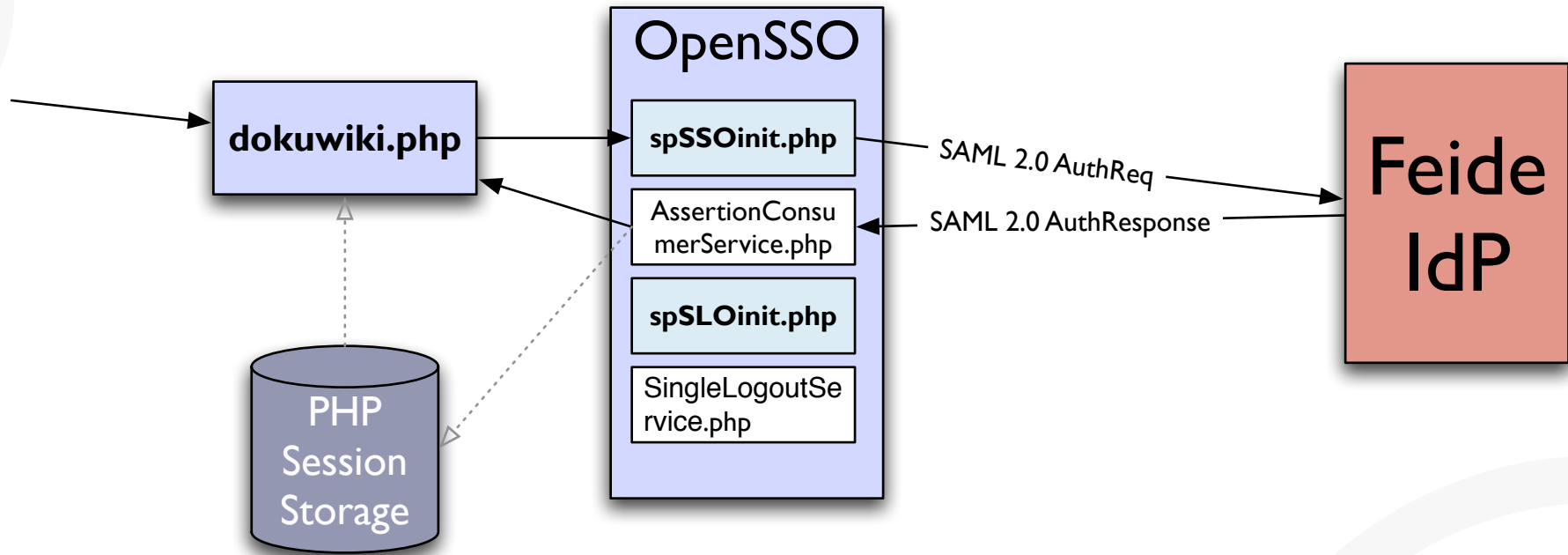
```
# none 0
# read 1
# edit 2
# create 4
# upload 8

# wikipage @group permissions
* @ALL 0 # No access
* @affiliationXemployeeXuninettXno 1 # employee read all
* @affiliationXhospitantXuninettXno 1 # students read all
:employee @affiliationXemployeeXuninettXno 15 # employee write one wiki-page
* @ufisa 16 # custom group "ufisa" write all pages
```

The auth module requires no local users at the wiki to map against. But optionally users can be configured custom group membership in a separate file.

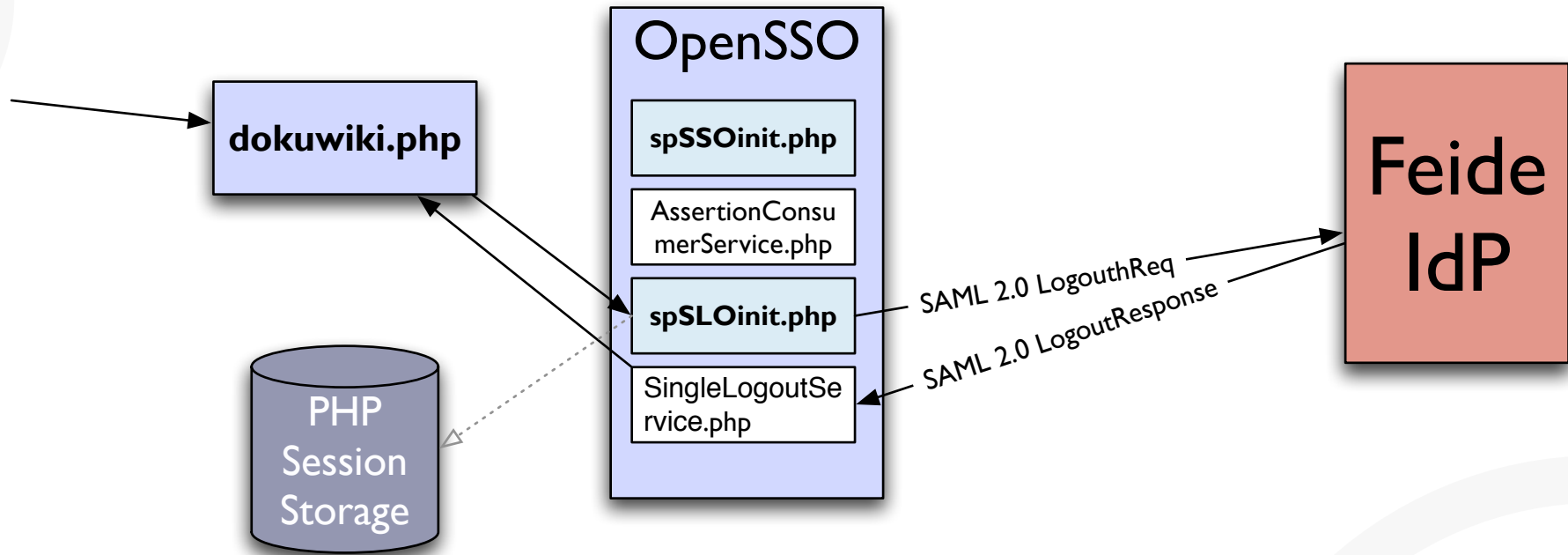


Login sequence





Logout sequence



FEIDE

?



UNI  **NETT** 