Federated Wikis

Andreas Åkre Solberg
andreas@uninett.no

UNINETT

Wikis in the beginning

...in the beginning wikis were wide open.

Great! - But then the spammers arrived.

Password protected wikis

Create yet another account, with yet another password. And registrations is open, so basicly anyone can register and anonymously terrorize the wiki.

FEIDE

Introducing...

UNINETT

Why?

Federated wikis:

- does not require registration
(convenient for user)
- works with Single-Sign-On
(convenient for user)
- Can be anonymous, but trackable!
Wiki admin sets the degree of anonymity.
- Can use **trusted attributes** to perform access control!

UNINETT

FEIDE Software used

- Dokuwiki

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

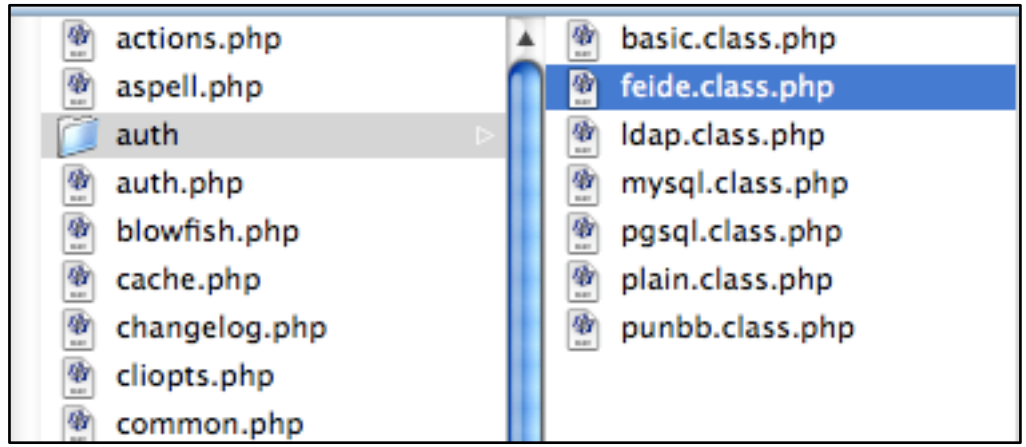
- simpleSAMLphp

http://rnd.feide.no/simplesamlphp

UNINETT

FEIDE Dokuwiki

Pluggable authentication modules



Supports ACL lists, and is using groups for authorization.

UNINETT

simpleSAMLphp

A native full PHP5 implementation of a **SAML 2.0 SP**. Extremely simple installation and configuration.

- **Install** (drop the folder)
- **Configure** (setup SAML 2.0 metadata)
- Test the examples, and **run** it with your application.

BTW: It also supports SAML 2.0 IdP, Shibboleth 1.3 SP, Shibboleth 1.3 IdP, bridging, Radius/LDAP/SQL backends, OpenID Provider, OpenID bridging, eduGAIN ++.

UNINETT

simpleSAMLphp configuration

SAML 2.0 IdP: Feide

```
21      /*
22       * SAML 2.0 SP Configuration for the Feide demo wiki
23       */
24      'urn:mace:feide.no:services:no.uninett.wiki-demowiki' => array(
25          'host'                          =>  'demowiki.feide.no',
26          'spNameQualifier'               =>  'demowiki.feide.no',
27          'NameIDFormat'                  =>  'urn:oasis:names:tc:SAML:2.0:nameid-format:transient',
28          'ForceAuthn'                    =>  'false'
29      ),
30
```

SAML 2.0 SP: Meta data for the wiki

```
44      /*
45       * Metadata for Feide's production environment.
46       */
47      'sam.feide.no' =>  array(
48          'SingleSignOnService'    =>  'https://sam.feide.no/amserver/SSORedirect/metaAlias/idp',
49          'SingleLogoutService'    =>  'https://sam.feide.no/amserver/IDPSloRedirect/metaAlias/idp',
50          'certFingerprint'        =>  '3a:e7:d3:d3:06:ba:57:fd:7f:62:6a:4b:a8:64:b3:4a:53:d9:5d:d0',
51          'base64attributes'       =>  true
52      )
```

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

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 accociates 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

In the **DokuWiki auth module**, load simpleSAMLphp

```
/* Load simpleSAMLphp configuration and metadata */
$config = SimpleSAML_Configuration::getInstance();
$metadata = new SimpleSAML_XML_MetaDataStore($config);
$session = SimpleSAML_Session::getInstance();
```

If session is not valid, then redirect to simpleSAMLphp for initializating a SAML 2.0 Authentication Request

```
/*
 * If session is not set or not valid, then we redirect to the
 * simpleSAMLphp SSO initialization page.
 */
if (!isset($session) || !$session->isValid() ) {
    header('Location: /simplesaml/saml2/sp/initSSO.php?RelayState=' .
        urlencode(SimpleSAML_Utilities::selfURL()));
    exit(0);
}
```

UNINETT

Implementing an authentication module

Next, user returns to the same page (remember the `RelayState` parameter), but is not catched by the `if (not authenticated)` section. Now we know the user is **authenticated**. We set user ID and mail attribute.

```
/* Retrieve attributes from the session. */
$attributes = $session->getAttributes();

/* Set the DokuWiki user ID from eduPersonPrincipalName and also the mail variable */
$user =  preg_replace("/[^a-zA-Z0-9]/", "X", $attributes['eduPersonPrincipalName'][0]);
$mail = $attributes['mail'][0];
```

UNINETT

Dynamic group membership

We generates some dynamic groups based on SAML 2.0 attributes:

```
if (is_array($attributes['eduPersonAffiliation']) ) {
    foreach ($attributes['eduPersonAffiliation'] AS $affiliation) {
        $groups[] = preg_replace("/[^a-zA-Z0-9]/", "X", "affiliation-" . $affiliation . "-" .
        $organization);
    }
}
if (isset($attributes['eduPersonOrgUnitDN'][0])) {
    $groups[] = preg_replace("/[^a-zA-Z0-9]/", "X", "orgunit-" .
        $attributes['eduPersonOrgUnitDN'][0] . "-" . $organization);
}
```

Resulting group membership for andreas@uninett.no:

- orgXuninettXno
- affiliationXemployeeXuninettXno
- affiliationXmemberXuninettXno
- orgunitXouXSUXouXTAXouXUNINETTXouXorganizationXdcXuninettXdcXnoXuninettXno

UNINETT

Custom groups

Sometimes you have local groups at a service, that can not be generated dynamically from attributes at the IdP, right?

Let's make a custom groups file (`conf/customgroups.php`):

```
$customgroups = array(
    'andreasXuninettXno'    => array('ufisa', 'feidecore', 'dame',
                                'norgridadmin', 'norstoreadmin', 'gigacampusadmin'),
    'catoXuninettXno'       => array('feidecore'),
    'andersXuninettXno'     => array('feidecore', 'dame'),
    'heidirXuninettXno'     => array('extabc'),
```

And load the custom groups of the user into the Dokuwiki auth module:

```
include($conf['groupfile']);
if (is_array($customgroups[$user])) {
    foreach ($customgroups[$user] AS $group) {
        $groups[] = $group;
    }
}
```

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**.

```
/*
 * Set variables into the global USERINFO, to be readable for DokuWiki
 */
$USERINFO['name'] = $user;
$USERINFO['mail'] = $mail;
$USERINFO['grps'] = $groups;

/*
 * We are done in the auth module, let's return back to DokuWiki to
 * process the page.
 */
return true;
```

UNINETT

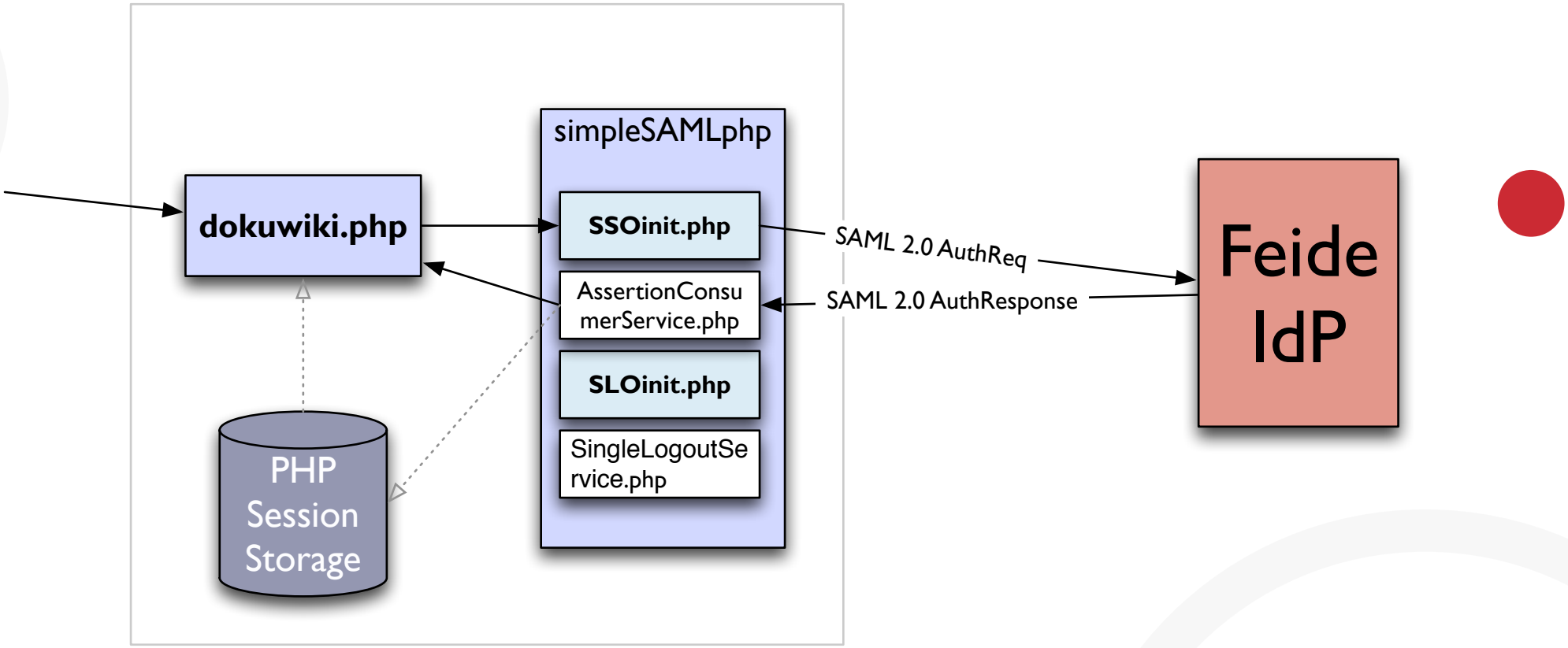
Access Control List

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

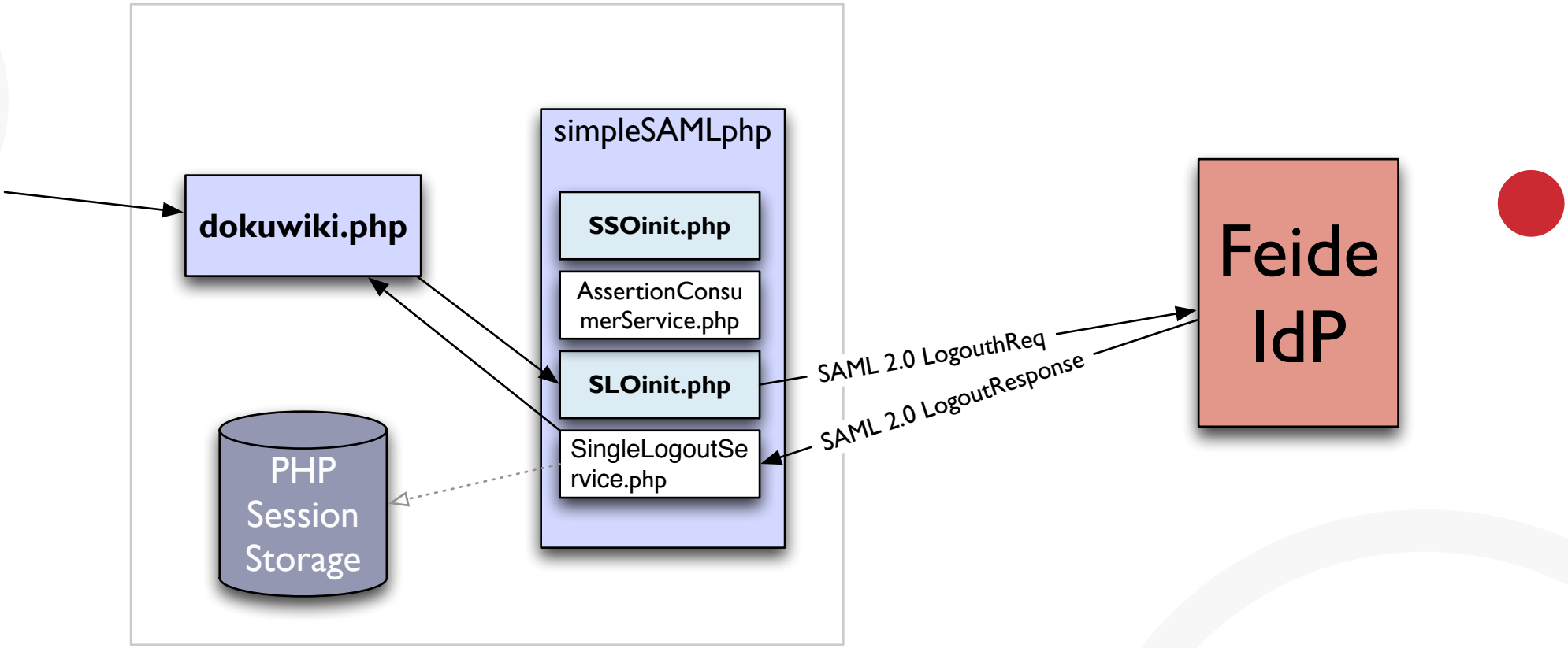
```
# none   0
# read   1
# edit   2
# create 4
# upload 8

# 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
```

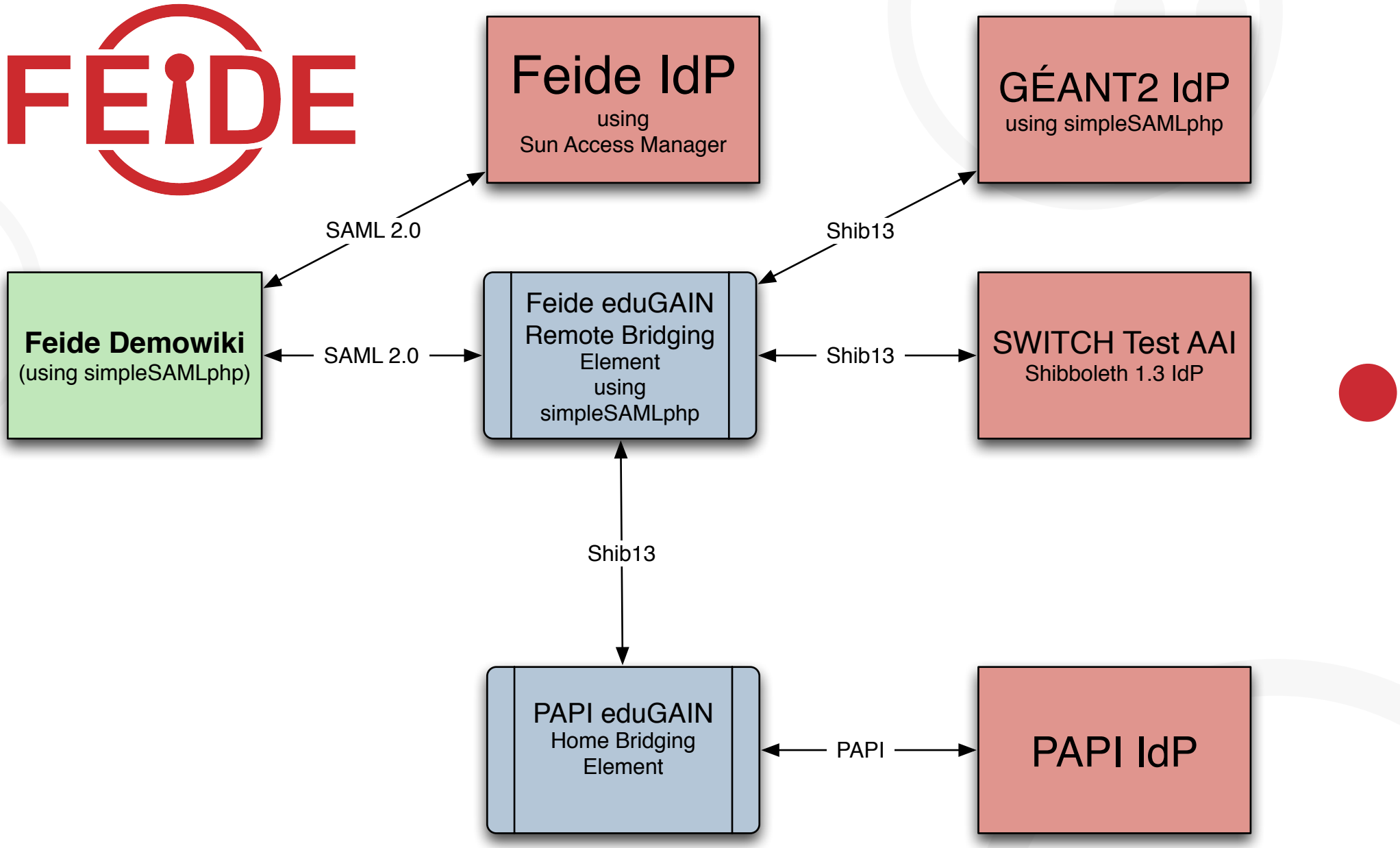
The auth module requires no local users at the wiki to map against. But optionally users can be configured custom group membership in a separate file.

Login sequence



Logout sequence




Feide IdP
using
Sun Access Manager
GÉANT2 IdP
using simpleSAMLphp
SAML 2.0
Shib13
Feide Demowiki
(using simpleSAMLphp)
SAML 2.0
Feide eduGAIN
Remote Bridging
Element
using
simpleSAMLphp
Shib13
SWITCH Test AAI
Shibboleth 1.3 IdP
Shib13
PAPI eduGAIN
Home Bridging
Element
PAPI
PAPI IdP

Feide RnD

Read more about other projects

http://rnd.feide.no

(feel free to subscribe to the RSS)

FEIDE

?

UNINETT