

eduGAIN Access Check architecture

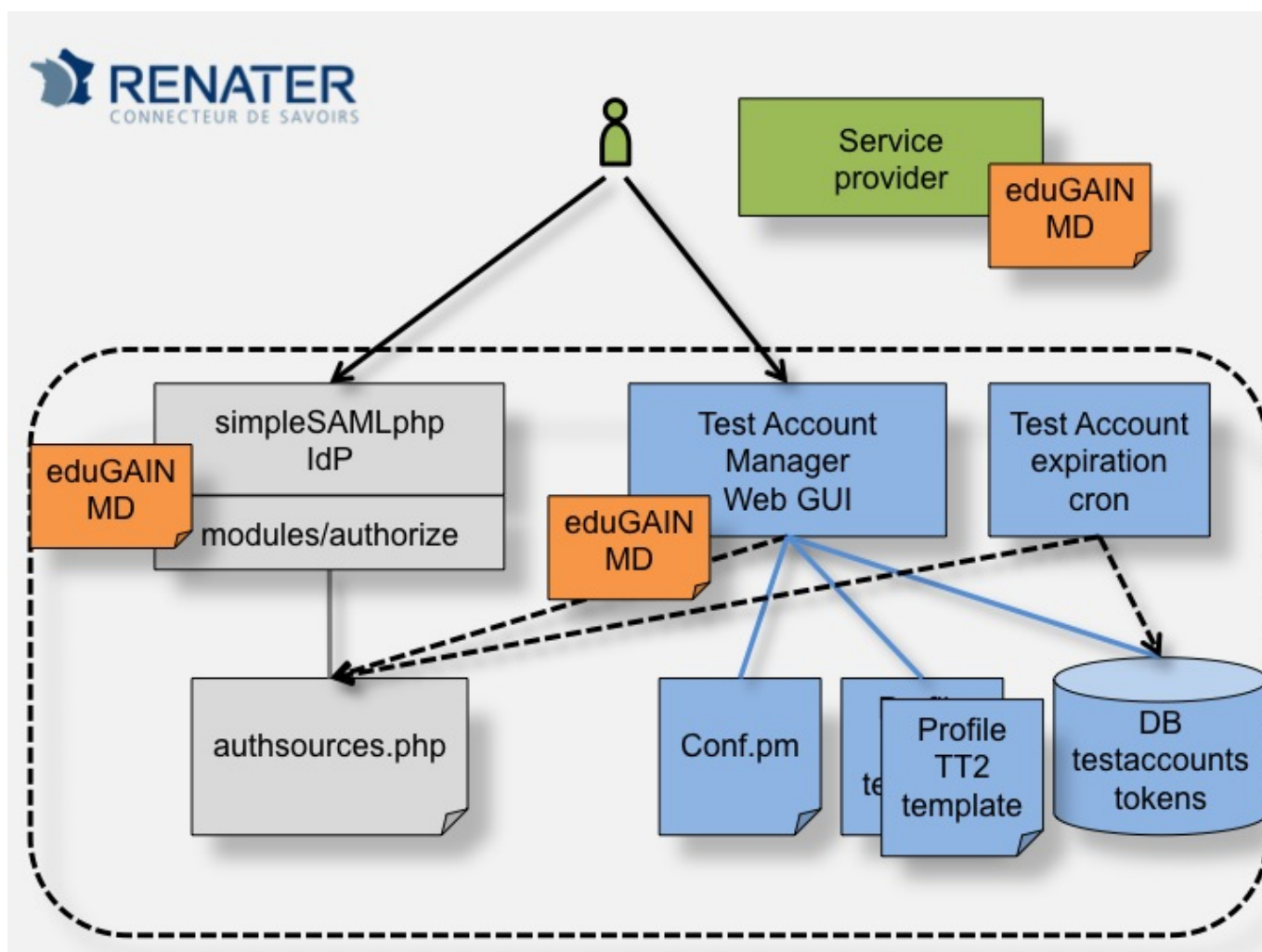
The Access Check IdP is in eduGAIN metadata and therefore can be used by SP admins to validate the behaviour of their federated service in the production environment. Only SP admins have the right to create and use test accounts at the Access Check IdP. The test accounts can be used for authentication only when accessing the own service, which also implies that user attributes are only released to the own SP. While creating a test account, the SP admin can select a user profile (e.g. student, researcher, teacher) with an associated set of user attributes.

Copyright and intellectual property

Copyright © GEANT

This software was developed by RENATER. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 238875 (GÉANT).

Overview



Technical components

- eduGAIN Access Check Identity Provider
 - implemented with simpleSAMLphp
 - registered in eduGAIN
 - coupled with the eduGAIN Test account manager
- eduGAIN Test account manager
 - application developed by RENATER
 - a web GUI to create test accounts
 - a cron script to expire test accounts

eduGAIN Access Check Identity Provider

simpleSAMLphp is configured to act as an Identity Provider and is registered in eduGAIN interfederation.

configuration of test accounts

simpleSAMLphp loads its user accounts from a flat file included in config/authsources.php:

```
include "/opt/testidp/IdPAccountManager/conf/valid-accounts.php";
```

The included valid-accounts.php file is automatically generated by the eduGAIN Test account manager, every time test accounts are created/removed.

For each test account, valid-accounts.php file provides the user ID, password hash and a set of user attributes, derived from a test account profile:

```
<?php
// template for a PhP configuration file loaded in simpleSamlPhp
authsources.php file
$validTestAccounts = array (
    'authcrypt:Hash',

    'user319:{SHA256}YvNJqmd3VFjqvqDYcyTcXcr8zZdhkuhALTbyMrgJtF4=' => array(
        'cn' => 'John Kleinman',
        'displayName' => 'John Kleinman',
        'eduPersonAffiliation' => array('member','student'),
        'eduPersonPrincipalName' => '319@dev-edugain.renater.fr',
        'eduPersonScopedAffiliation' => array(
            'member@dev-edugain.renater.fr','student@dev-edugain.renater.fr'),
        'eduPersonTargetedID' =>
            'https://dev-edugain.renater.fr/simplesaml/saml2/idp/metadata.php!https://test.federation.renater.fr/test/ressource!X622UR2A7PGluVhATobB0rMz+Ys=',
        'mail' => 'john.kleinman@dev-edugain.renater.fr',
        'schacHomeOrganization' => 'dev-edugain.renater.fr',
        'schacHomeOrganizationType' =>
            'urn:schac:homeOrganizationType:int:university',
        'uid' => '319',

        'associatedSP' => "https://test.federation.renater.fr/test/ressource",
    ),

    'user320:{SHA256}3cb3hdyg2VYHiVqlaWXVukp95VG80MjYPp2SUKUFH8g=' => array(
        'cn' => 'Peter Smith',
        'displayName' => 'Peter Smith',
        'eduPersonAffiliation' => array('member','faculty'),
        'eduPersonPrincipalName' => '320@dev-edugain.renater.fr',
        'eduPersonScopedAffiliation' => array(
            'member@dev-edugain.renater.fr','student@dev-edugain.renater.fr'),
        'eduPersonTargetedID' =>
            'https://dev-edugain.renater.fr/simplesaml/saml2/idp/metadata.php!https://test.federation.renater.fr/test/ressource!X622UR2A7PGluVhATobB0rMz+Ys=',
        'mail' => 'peter.smith@dev-edugain.renater.fr',
        'schacHomeOrganization' => 'dev-edugain.renater.fr',
        'schacHomeOrganizationType' =>
```

```
'urn:schac:homeOrganizationType:int:university',  
    'uid' => '320',  
  
    'associatedSP' => "https://test.federation.renater.fr/test/ressource",  
),
```

configuration to restrict login for each SP

- see [simpleSAMLphp authentication processing filters documentation](#)

Each test account is bound to a SP and should be used to login at this SP only. That behavior is the main specificity to the eduGAIN Access Check IdP compared to a standard IdP.

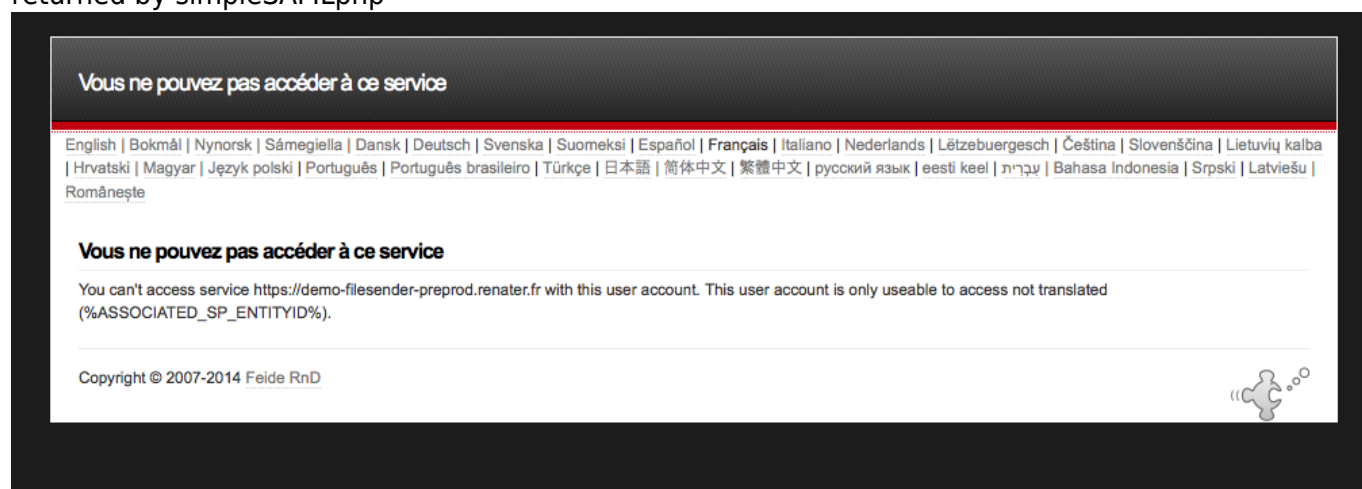
This behavior has been added through an authentication processing filter using the **restrictLogin** simpleSAMLphp module developed by RENATER. The **restrictLogin** simpleSAMLphp is not distributed with simpleSAMLphp. Here it is as a tar.gz: [restrictlogin_ssp_module.tar.gz](#)

The authentication processing filter is configured in config/config.php file:

```
60 => array(  
    'class' => 'restrictLogin:RestrictLogin',  
    'sp_entityid_attr' => 'associatedSP',  
),
```

The processing filter makes use of a special user attributes: associatedSP. This user attributes is provisionned by the account manager and refers to the entityID of the SP that is associated to each test user account.

If the test account is used to access a SP different than the one it is bound, an error message is returned by simpleSAMLphp



eduGAIN Test account manager

The eduGAIN Test accounts manager is the application responsible for creating, removing test accounts. It has been developed by RENATER.

implementation

The application is written in Perl using the following specific CPAN modules:

- Template: template toolkit is used to write templates (mail, web, test account).
 - Check [the Template toolkit documentation](#).
- Rose::DB::Object: a Perl ORM (Object Relational Mapper) used to automatically generate the RDBMS code.
 - Check [Rose::DB::Object documentation](#)
 - RDBMS supported by the library are: Pg, mysql, SQLite, Informix, Oracle
- XML::LibXML: an XML parser used to parse SAML metadata

The web GUI of the application uses the following frameworks:

- [jQuery](#)
- [jQuery UI](#)
 - <http://jqueryui.com/autocomplete/> autocomplete widget provides autocompletion in the SP selection menu
- [jQuery-Steps](#): a wizard plugin
- [jQuery validation](#): a form validation plugin

versioning

The code is versionned via a dedicated SVN repository on RENATER's forge:

<https://sourcesup.renater.fr/projects/accountmanager/>.

SVN repository is not public.

the database

The application uses a mySQL database to store data related to:

- validation tokens
- test accounts

Here is the database structure:

```
CREATE TABLE `authenticationtokens` (  
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
  `token` VARCHAR(50) NOT NULL,  
  `email_address` VARCHAR(200) NOT NULL,  
  `sp_entityid` VARCHAR(200) NOT NULL,  
  `creation_date` INT(11) DEFAULT NULL,
```

```
PRIMARY KEY (`id`),  
UNIQUE KEY `token_2` (`token`),  
KEY `token` (`token`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=63 ;  
  
CREATE TABLE `testaccounts` (  
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,  
  `user_password_hash` VARCHAR(50) NOT NULL,  
  `creation_date` INT(11) DEFAULT NULL,  
  `expiration_date` INT(11) DEFAULT NULL,  
  `account_profile` VARCHAR(100) NOT NULL,  
  `sp_entityid` VARCHAR(250) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=371 ;
```

Perl code to use the database is handled by the ORM (Rose::DB::Object).

logs and notifications

The application keeps track of accesses and errors in a dedicated log file.

The application admins get email notified whenever a significant error occurs.

configuration

A main configuration file allows to configure the application:

[| conf/Conf.pm](#)

```
package Conf;  
  
our %global = (  
  
    ## Code version  
    'version' => '0.1',  
  
    ## Name of the application used in web pages, mail notices  
    'app_name' => 'eduGAIN Test Account Manager',  
  
    ## URL of the application  
    'app_url' => 'https://dev-edugain.renater.fr/accountmanager',  
  
    ## Validity period of test accounts, in days  
    'accounts_validity_period' => 7,  

```

```
## Scope used by the Test IdP
'idp_scope' => 'dev-edugain.renater.fr',

## EntityID of the IdP
'idp_entityid' =>
'https://dev-edugain.renater.fr/simplesaml/saml2/idp/metadata.php',

## Name of the IdP
'idp_displayname' => 'eduGAIN Access Check',

## Root simpleSamlPhp directory
'root_ssp_dir' => '/opt/testidp/simplesamlphp',

## Root test account manager directory
'root_manager_dir' => '/opt/testidp/IdPAccountManager',

## Database type refers to a Perl Database Driver name
## However only a subset of existing DBDs are supported by
Rose::DB::Object:
## Pg, mysql, SQLite, Informix, Oracle (DBD names are case
sensitive)
'database_type' => 'mysql',

## Database hostname
'database_host' => 'localhost',

## Database_name
'database_name' => 'idp_account_manager',

## Database username
'database_user' => 'idpadmin',

## Database user password
'database_password' => 'XXX',

## Log file for the manager
'log_file' => '/opt/testidp/IdPAccountManager/log/manager.log',

## Log level : debug, info, trace, notice, error
'log-level' => 'info',

## email address to contact admins
'admin_email' => 'olivier.salaun@renater.fr',

## email address to ask for support
'support_email' => 'support@geant.net',

## Development feature
## Protection to prevent notifications during test dev phases
## Notify only admin_email above
'dev_no_mail_outside' => 1,
```

```
## Development feature
## hard-coded list of contactPersons
## these email addresses will be added to the list of contacts for
any SP
'dev_sp_contact' =>
'olivier.salaun@renater.fr,mehdi.hached@renater.fr,lukas.haemmerle@switch.ch,thomas.baerecke@switch.ch,florio@terena.org,ann.harding@switch.ch,thomas.lenggenhager@switch.ch,mikael.linden@csc.fi,marco.malavolti@garr.it,marialaura.mantovani@garr.it,pempe@dfn.de,Mandeep.Saini@dante.net,simona.venuti@garr.it',
  #'dev_sp_contact' =>
'olivier.salaun@renater.fr,mehdi.hached@renater.fr,lukas.haemmerle@switch.ch,thomas.baerecke@switch.ch',

## From field use by the account manager
'notice_from' => 'testidpaccountmanager@geant.net',

## federation metadata local copy path
#'federation_metadata_file_path' =>
'/opt/testidp/IdPAccountManager/conf/edugain-md.xml',
  'federation_metadata_file_path' =>
'/opt/testidp/IdPAccountManager/conf/renater-test-metadata.xml',

## valid account profiles
'account_profiles' => ['student1','teacher1'],
);
```

test accounts and test account profiles

Account creation

Each SP admin gets a set of test accounts created after he has completed the test account wizard



eduGAIN Test Account Manager -

1. Select your Service Provider

2. Send email challenge

3. Validate challenge

4. Get test accounts

Select your Service Provider

This Test Identity Provider allows you to create test accounts with different profiles to validate the behaviour of your own Service Provider registered in eduGAIN inter-federation. Note that only a Service Provider administrator can create accounts here.

Please select your Service Provider below

Previous

Next

The test account wizard follows the following steps:

1. SP admin selects his SP from a dropdown list,
2. SP admin selects his email address (or one that can be used to contact him) among the contactPersons declared for this SP in eduGAIN metadata,
3. SP admin receives an authentication token at this email address. User pastes the token in the wizard,
4. token is checked (and deleted from the database). Test accounts gets created. Credentials for the test accounts are provided to the SP admin in the wizard.
5. SP admin can see the detailed attribute profiles

Account details

Below is the list of user attributes associated to this test account. After a successful authentication process these attributes are transmitted via the SAML protocol from the Test Identity Provider to your Service Provider.

- cn: John Kleinman
- displayName: John Kleinman
- eduPersonAffiliation: member,student
- eduPersonPrincipalName: 371@dev-edugain.renater.fr
- eduPersonScopedAffiliation: member@dev-edugain.renater.fr,student@dev-edugain.renater.fr
- eduPersonTargetedID: https://dev-edugain.renater.fr/simplesaml/saml2/idp/metadata.php!https://shibb-dev.univ-nancy2.fr/sp!X622UR2A7PG1uVhATobBOrMz+Ys=
- mail: john.kleinman@dev-edugain.renater.fr
- schacHomeOrganization: dev-edugain.renater.fr
- schacHomeOrganizationType: urn:schac:homeOrganizationType:int:university
- uid: 371

If you a customized test account with additionnal user attributes, you should contact support@geant.net

Ok

Account credentials

A password is generated for each account. This password is provided to the SP admin on the web page after the account gets created

Get test accounts

Your identity has been checked successfully. Test accounts with different profiles have been created for you, see details below. Note that these accounts will automatically expire in 7 days. Note also that these test accounts can only be used to login on your SP <https://www.portal.euromonitor.com/shibboleth>.

1. account profile: student1
user name: user369
user password: Y-<2w=1=%
[show account details](#)
2. account profile: teacher1
user name: user370
user password: 2Nh3pAcx:
[show account details](#)

You can now use these accounts to login at your federated service. To do so, you should select "eduGAIN Test Account Manager" in your service discovery service menu. You should keep a record of these user names and passwords since we don't provide any mechanism to remind you the credentials. If you forget the credentials you can however ask for new test accounts via this service. Thank you for using the eduGAIN Test Account Manager

Password is **not** sent via email.

Password cannot be changed.

Password is not stored in clear form; only a SHA256 hash is stored in the database.

Account profiles

Each test account is created using a test account template. Test account templates are TT2 files. Here is a sample test account template:

[| templates/accountProfiles/student1.tt2](#)

```
[% account_sp_entityid = account.get('sp_entityid') %]
[% account_id = account.get('id') %]

[% SET account_profile = {
  'uid' => "${account_id}",
  'eduPersonAffiliation' => ['member', 'student'],
  'eduPersonScopedAffiliation' => ["member@${conf.idp_scope}",
"student@${conf.idp_scope}"],
  'displayName' => 'John Kleinman',
  'cn' => 'John Kleinman',
  'mail' => "john.kleinman@${conf.idp_scope}",
  'eduPersonPrincipalName' => "${account_id}@${conf.idp_scope}",
  'eduPersonTargetedID'
=> "${conf.idp_entityid}!${account_sp_entityid}!X622UR2A7PG1uVhATobB0rMz
+Ys=",
  'schacHomeOrganization' => "${conf.idp_scope}",
  'schacHomeOrganizationType' =>
'urn:schac:homeOrganizationType:int:university',
```

```
} %]
```

Account profiles in use are listed in **account_profiles** configuration parameter:

```
## valid account profiles  
'account_profiles' => ['student1','teacher1'],
```

 **To do...** SP admins could add attributes to a profile, change attribute values

Account expiration

A dedicated script removes expired test accounts from the database. This script should be run daily via the crontab.

Expiration period of accounts is configured in the configuration file (default is 7 days):

| [conf/Conf.pm](#)

```
## Validity period of test accounts, in days  
'accounts_validity_period' => 7,
```

From:

<https://wiki-intranet.renater.fr/> - **INTRANET du GIP**

Permanent link:

https://wiki-intranet.renater.fr/dt/epa/activites/federation/services/test_idp_geant/architecture

Last update: **2015/03/12 13:30**

