Authentication in Galaxy : let's use what is out there - (National) Identity Providers

Nikolay Vazov

University Center for Information Technologies
University of Oslo

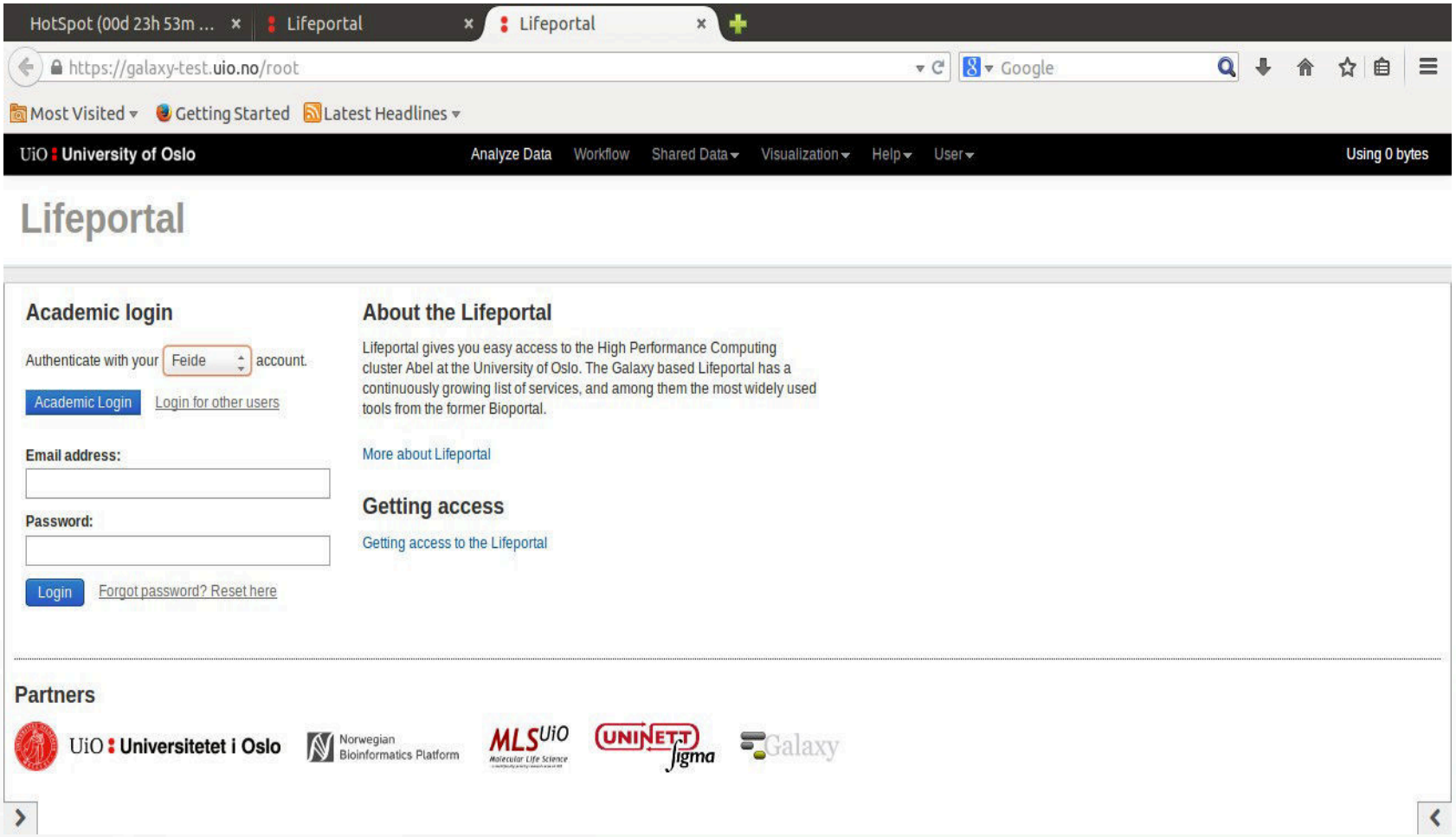
2nd Swiss Galaxy Workshop, October 2014

Authentication challenges

- Galaxy on clusters
  - Requires appropriate accounting techniques operating over a complex user database
  - Security issues : delegate authentication to authorized bodies (e.g. local LDAPs via authorized identity providers)
- Reusability and Reproducibility
  - Collaboration between researchers from different institutions requires flexible and secure user database management
  - Possibility of SSO from trusted identity providers

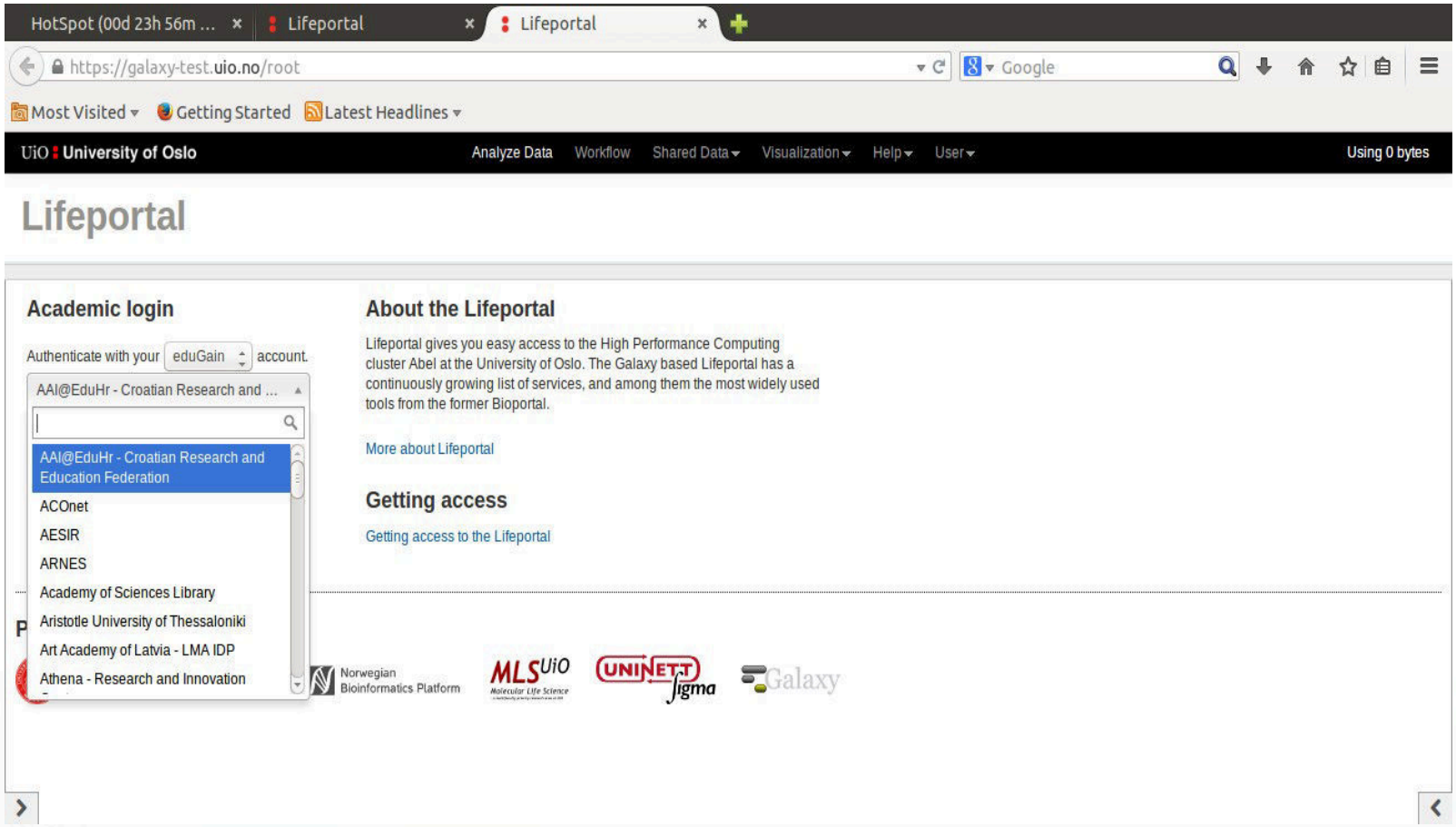
Example of the Lifeportal?

require_login=True



Example of the Lifeportal?

require_login=True



Identity Providers solution

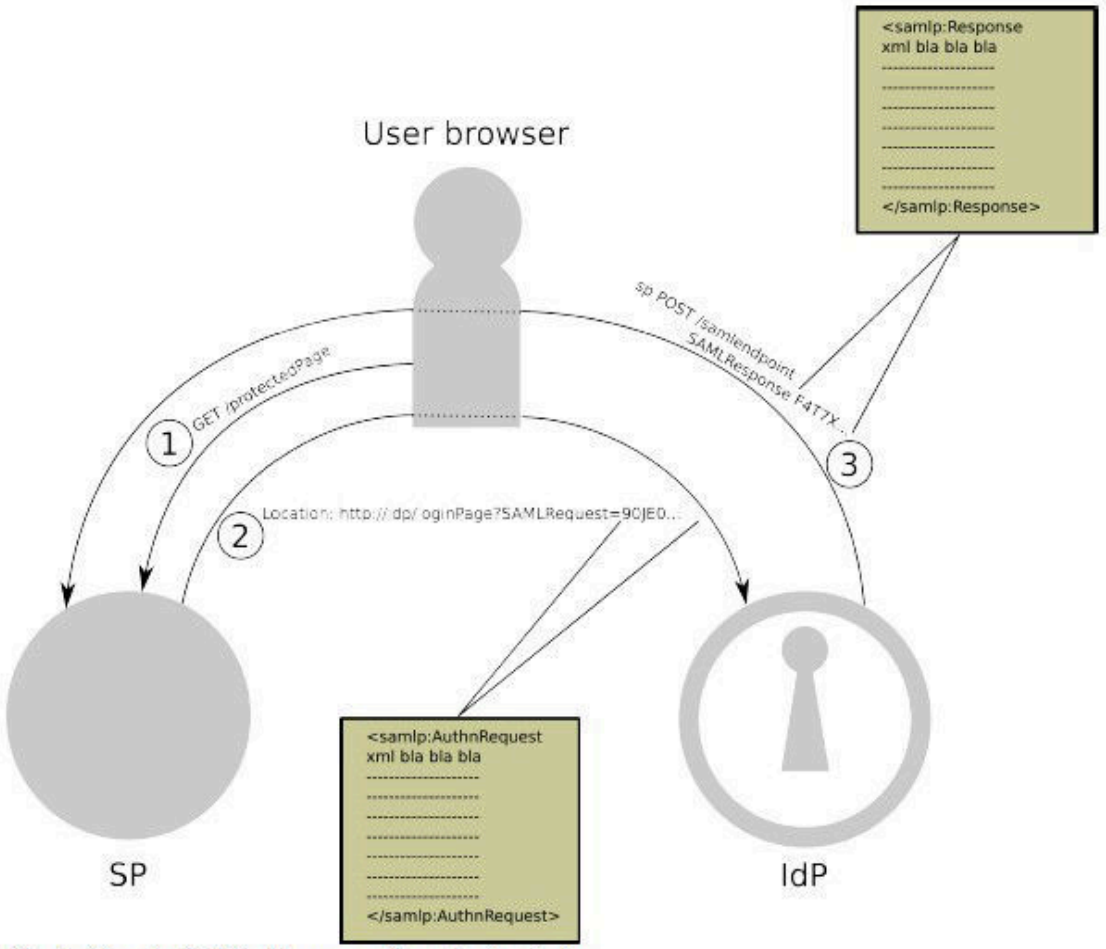


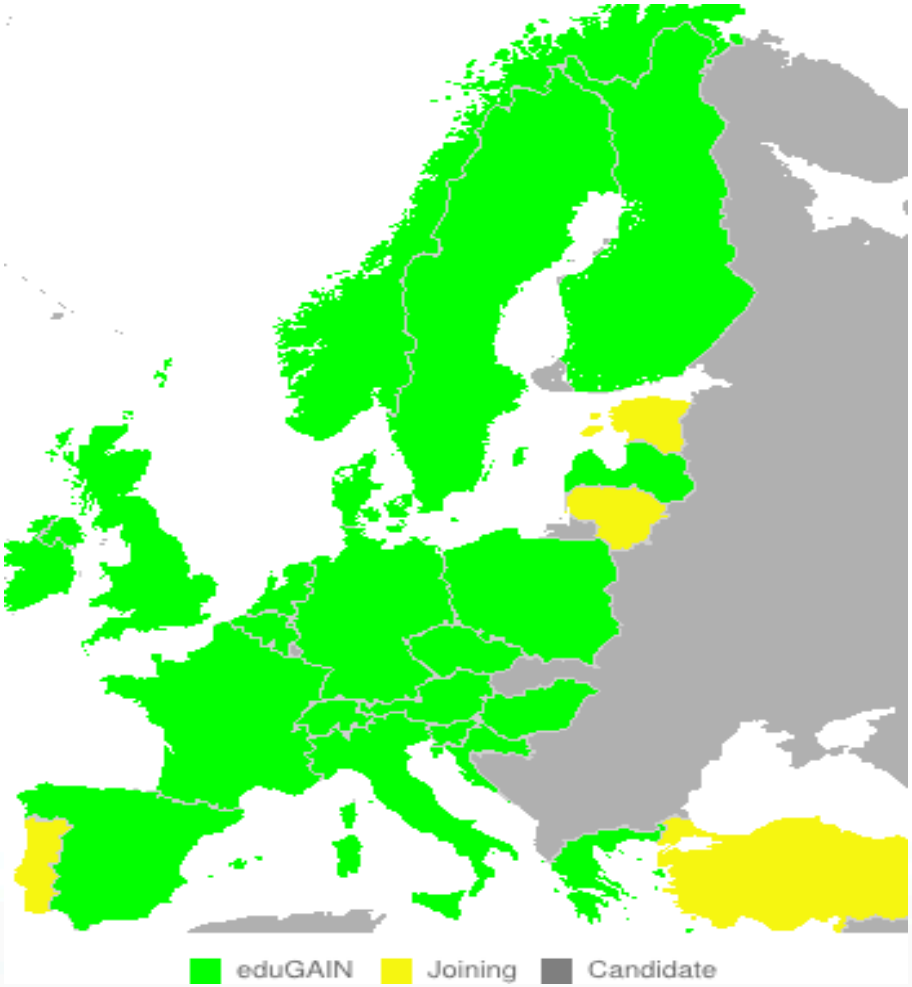
Illustration 1: SAML Message flow during login

So, let's delegate the task to ...

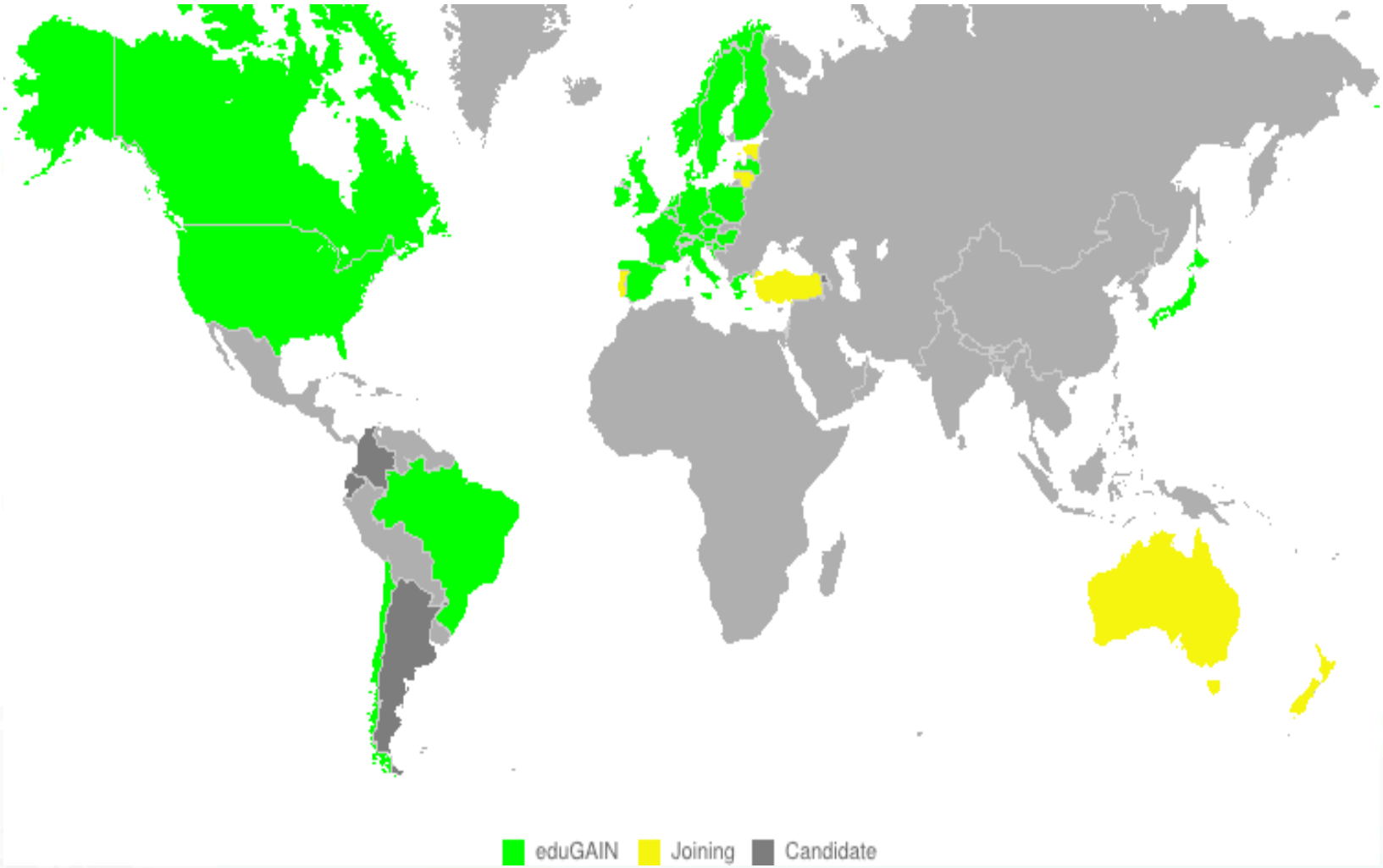
eduGain

- 528 institutions per today
- The eduGAIN service is intended to enable the trustworthy exchange of information related to identity, authentication and authorisation between the GÉANT (GN3plus) Partners' federations. The eduGAIN service will deliver this through co-ordinating elements of the federations' technical infrastructure and a policy framework controlling the exchange of this information. The initial goal is to enable Pan-European Web Single Sign On (Web SSO) to both GÉANT services and to those provided by other communities represented by, or associated with, the GN3plus Partners.

eduGain members



eduGain members



Enable a Galaxy instance to use external authentication

- Non-trusted Identity providers
  - OpenID

- Trusted : FEIDE (Norway), eduGain (Géant, includes FEIDE)

1. Use the existing galaxy “infrastructure”
   - OpenID

2. Install and configure the necessary packs
   - Pysaml (Roland Hedberg)

3. Make the changes in the Galaxy tree
   - Easy part : create the new files/directories
   - Difficult part : modify the Galaxy code

All we need is 1 additional pack

pysaml2-2.0.1beta-py2.6

- *tar zxvf pysaml2-2.0.1beta-py2.6.tar.gz*
- *cd pysaml2-2.0.1beta-py2.6*
- *python setup.py install*

Configure your service provider (SP)
(based on pysaml package and located *outside* Galaxy)

1. Prepare the keys/certificates

   generate your own or order valid ones for production instances

2. Configure your Service Provider (SP)

- create an sp_conf.py file
- generate sp metadata file (XML)
- send it to the IdP

- get the IdP metadata file (XML)

  /home/galaxy/idp/IdP-metadata.xml
  /home/galaxy/sp/SP-metadata.xml

3. Copy the directory "attributes"

http://bazaar.launchpad.net/~pysaml2maint/pysaml2/main/files/head:/example/sp

Example of *sp_conf.py* file

```
from saml2 import BINDING_HTTP_REDIRECT
from saml2 import BINDING_HTTP_POST
from saml2.saml import NAME_FORMAT_URI

BASE= "https://lap.hpc.uio.no/"

CONFIG = {
    "entityid" : "urn:mace:feide.no:services:no.uio.hpc.lap",
    "description": "LAP (Language Analysis Portal) server at UiO",
    "service": {
        "sp":{
            "name" : "LAP_UIO",
            "endpoints":{
                "assertion_consumer_service": [(BASE+"user/feide_auth",BINDING_HTTP_POST)],
                "single_logout_service" : [(BASE+"user/logout_feide_user",BINDING_HTTP_REDIRECT)],
            },
            "required_attributes": ["surname", "givenname",
                                    "edupersonaffiliation", "mail"],
            "optional_attributes": ["title"],
            },
    },
    "debug" : 1,
    "key_file" : "/etc/pki/tls/private/lap.hpc.uio.no.key",
    "cert_file" : "/etc/pki/tls/certs/cert-4297-lap.hpc.uio.no.pem",
    "attribute_map_dir" : "/home/laportal/attributemaps",
    "metadata" : {
       "local": ["/home/laportal/idp/idp-feide.no.xml", "/home/laportal/idp/mds.edugain.org.xml"],
    },
        # -- below used by make_metadata --
    "organization": {
        "name": "Galaxy Language Analysis Portal UiO",
        "display_name": [("LAP at UiO","en")],
        "url":"https://lap.hpc.uio.no",
    },
    "contact_person": [{
        "given_name":"Nikolay",
        "sur_name": "Vazov",
        "email_address": ["n.a.vazov@usit.uio.no"],
        "contact_type": "technical",
        },
    ],
    "xmlsec_binary":"/usr/bin/xmlsec1",
    "name_form": NAME_FORMAT_URI
}
```

Galaxy code changes for eduGain

8 files edited / 5 new

1. Declaration of the eduGain auth method
2. Loading the auth method information at boot
3. Processing the auth method information : authentication and logging in
4. Visualization issues

Declaration of the eduGain auth method

1. *universe_wsgi.in*i - Enable authentication via eduGain.
2. *edugain_conf.xml* – define all providers
3. *edugain/edugain.xml* – define urls for redirection
4. */lib/galaxy/edugainid* and */lib/galaxy/edugain/providers.py* – reuse OpenID

Loading the auth method information at boot

1. */lib/galaxy/config.py* – enable eduGain
2. */lib/galaxy/app.py* – load the configuration from file
3. */lib/galaxy/web/framework/__init__.py* - edit list of functions accessible by the web interface in require_login mode
4. */lib/galaxy/model/__init__.py* – manage eduGain passwords – IdPs do not communicate us the password, we store info about the IdP in clear text

Processing the auth method information : authentication and logging in

1. */lib/galaxy/webapps/galaxy/controllers/user.py* – main changes are here : new pysaml user class to process the request/response communication with the IdPs; login functions
2. *~/additional_python_galaxy_packs/IdPselector.py* - it returns a dictionary : keys - nameEntityID (url), values - the display name in user friendly format

Visualization (makos and js)

1. */lib/galaxy/templates/user/login.mako* – editing dropdowns
2. */lib/galaxy/templates/user/confirm registration.mako* - add this file as an intermediate template. If allows the user autenticated by eduGain, to accept/deny further login into Galaxy
3. *../galaxy-dist/static/scripts/galaxy.base.js* - adding the dropdown for eduGain members

Additional system files

1. Cron to extract the latest edugain metadata file
2. Cron to control the edugain metadata file for validity

Reusability and reusability again

- Fits the main philosophy of Galaxy
- Does not rule out the existing methods:
  - “Local” login preserved
  - Common Galaxy user database
  - Allows for integration of other authentication methods like LDAP

For more details

Full description of eduGain integration into Galaxy

Thank you