



Training on the EOSC-hub AAI

The service provider perspective



eosc-hub.eu

Dissemination level: Public

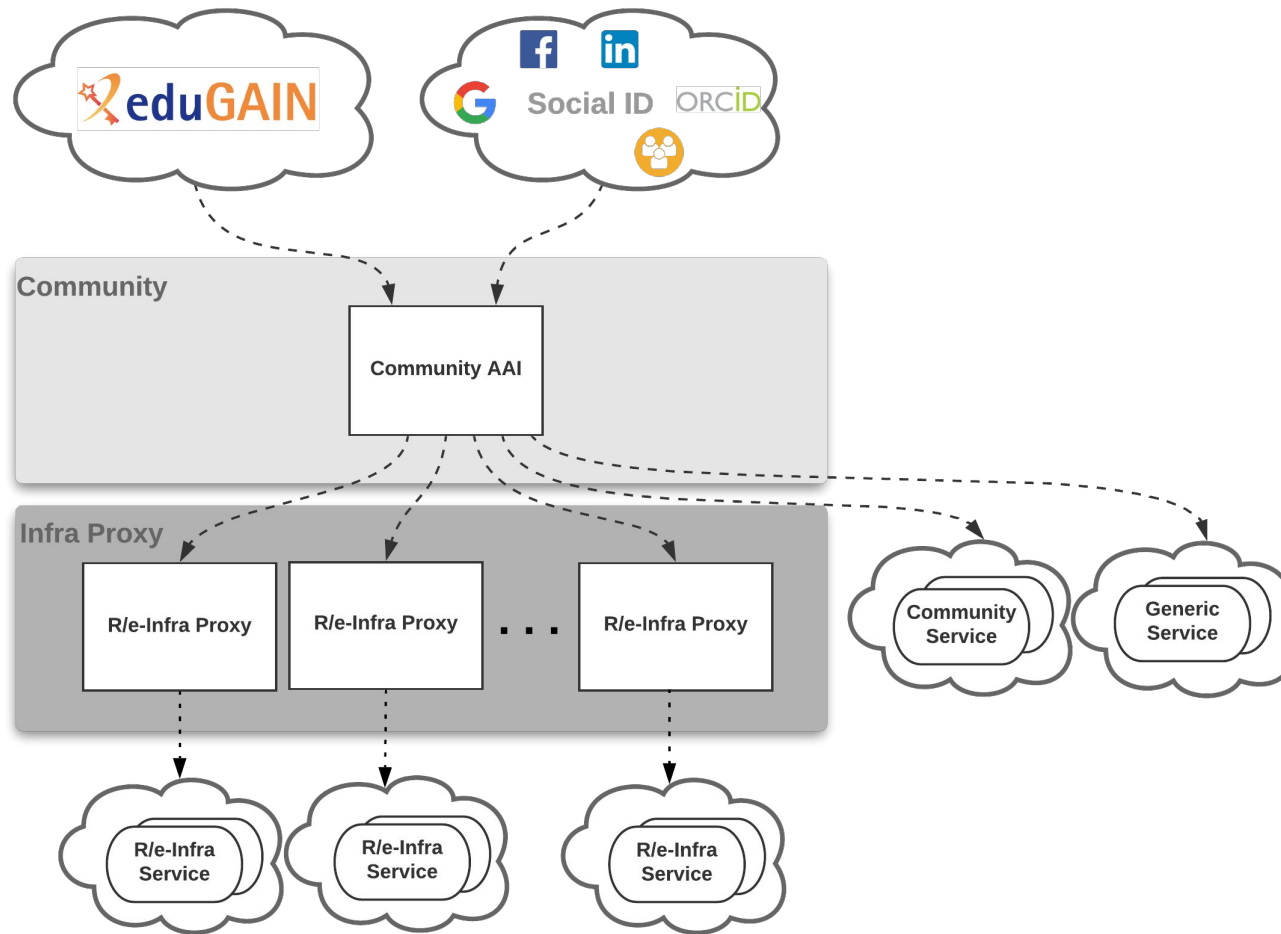


@EOSC_eu

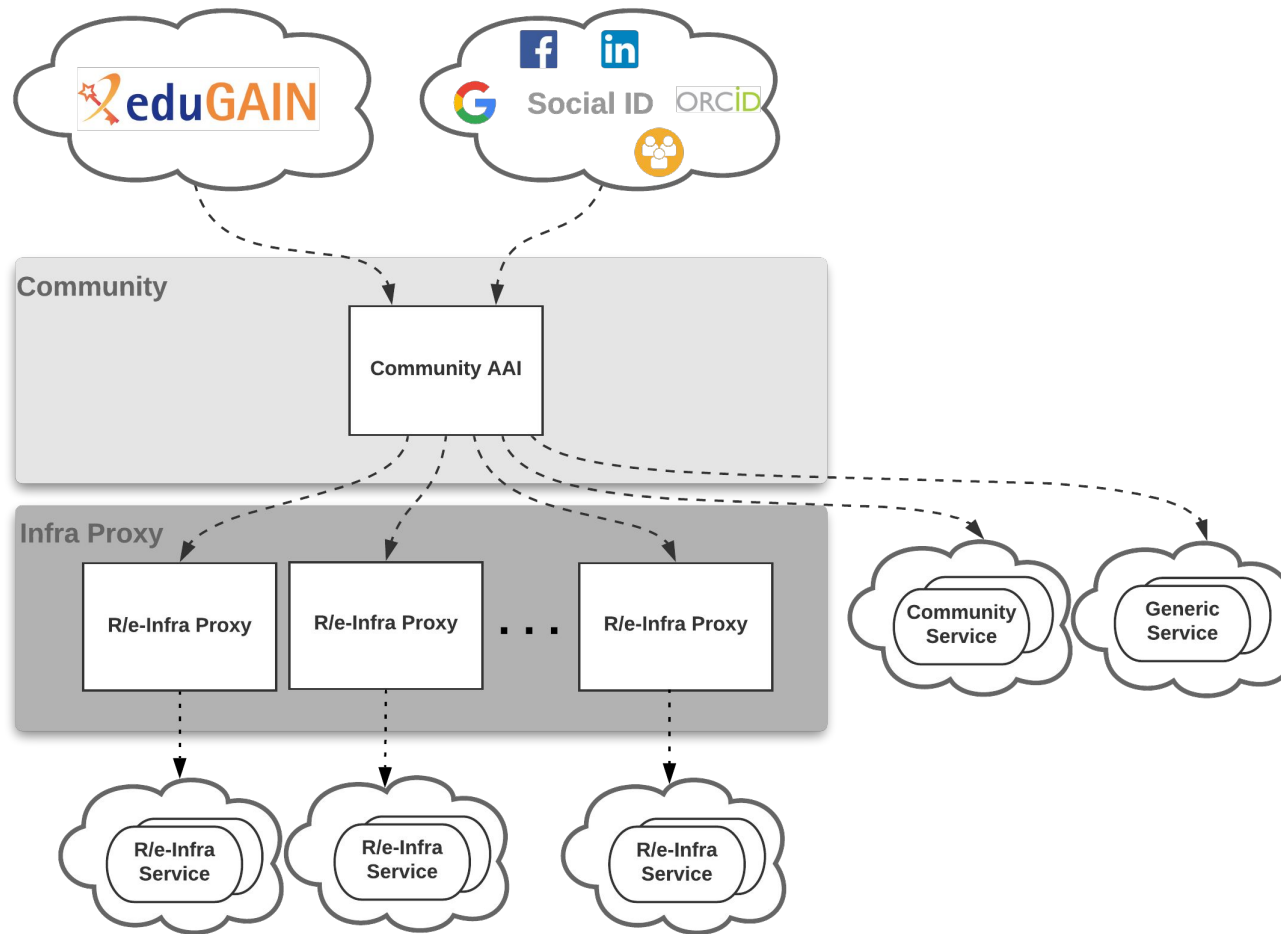


EOSC-hub receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777536.

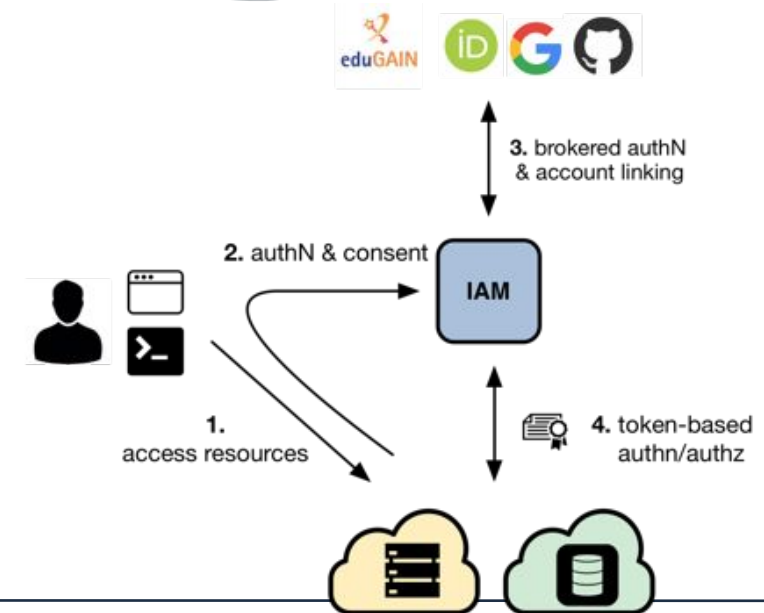
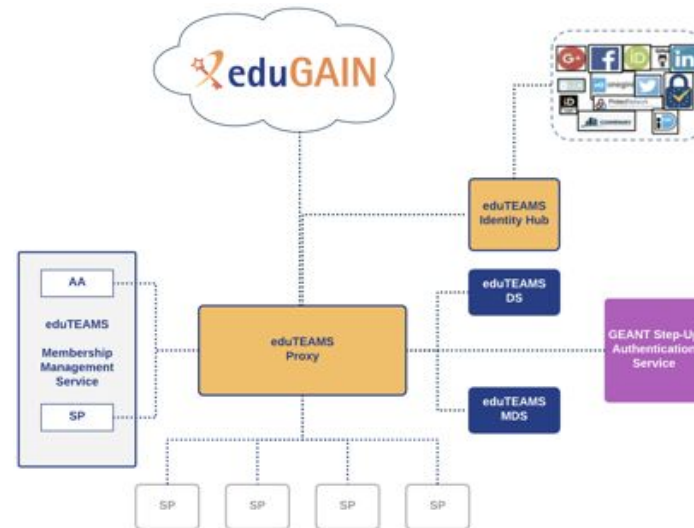
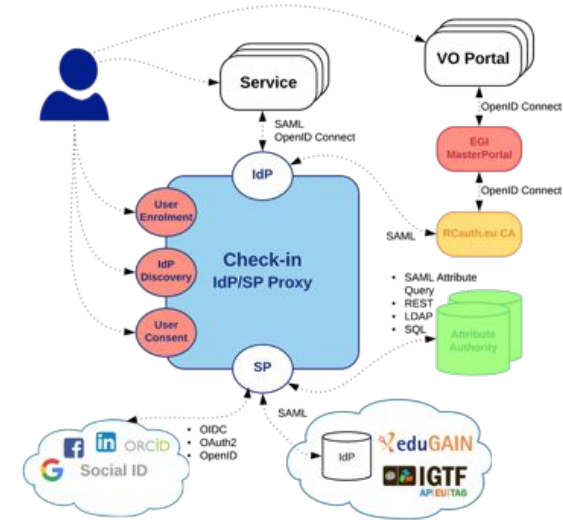
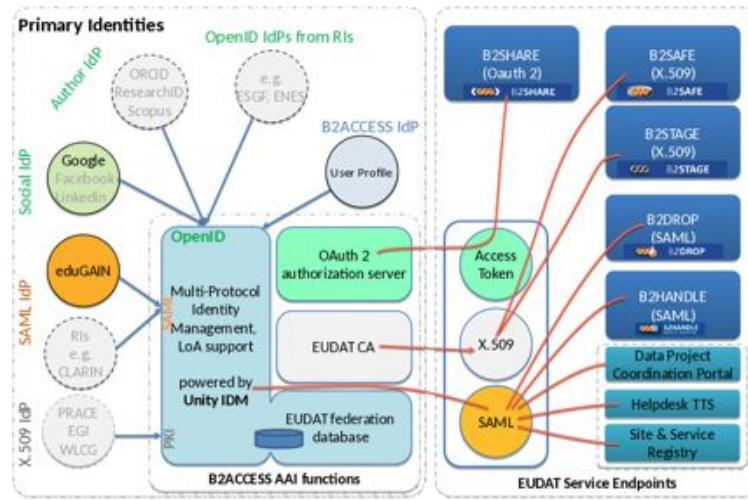
- Overview of the different EOSC-hub AAI services
- Service Provider integration flows - demos
- EOSC-hub AAI status & interoperability roadmap

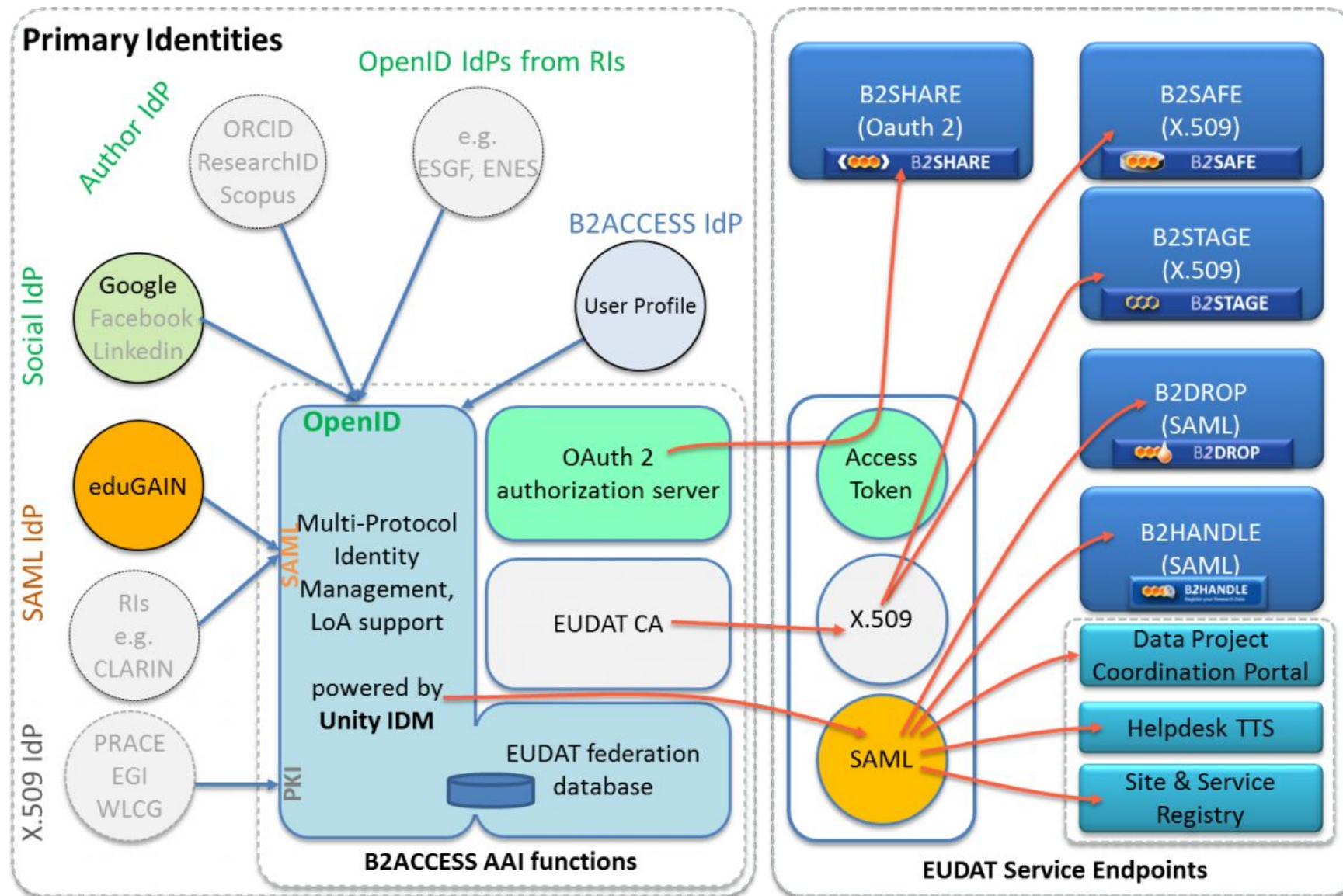


- Implementation of the AARC BPA “Community-first” approach:
 - Researchers register once with their Community AAI
 - Researchers always sign in via their Community AAI using their academic/social credentials for accessing:
 - **Community-specific** services
 - **Generic services** (e.g. RAuth.eu Online CA)
 - General-purpose **R/e-Infra services**



- R/e-Infra proxy serves as a single integration point for services
- No need to run an IdP Discovery Service on each service
- Services get consistent/harmonised user identifiers and accompanying attribute sets from different IdPs/AAs that can be interpreted in a uniform way for authorisation purposes





- Attribute management and translation
- Hierarchical group management on dedicated interface
 - Delegate subgroups to dedicated managers
- OIDC SP registration in self-service
- User registration
 - Self-service
 - Invitations
- Enquiries for requests to existing users
- Dedicated endpoints to sensitiv services

Identity and Access Management solution that makes it easy to secure access to services and resources



Components:

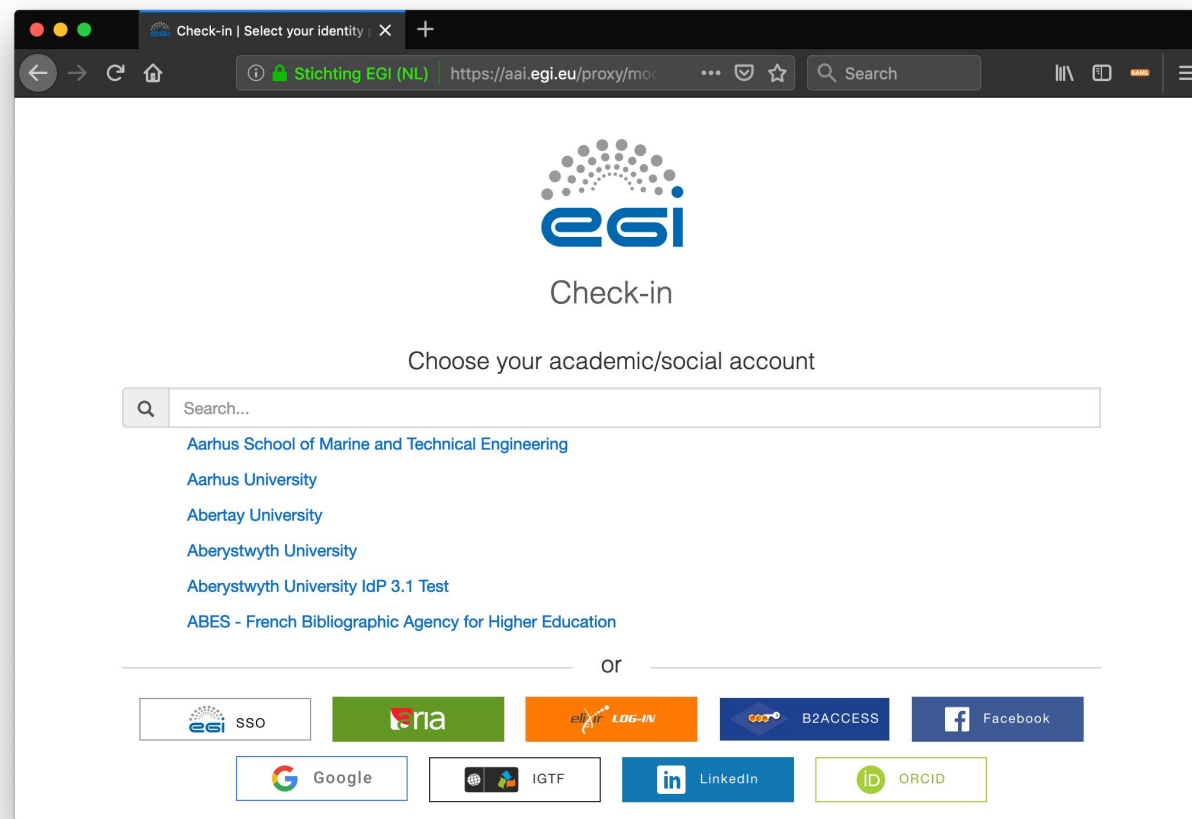
- IdP/SP Proxy
- User enrolment & group management
- IdP Discovery
- Token Translation

Documentation

- Usage guide
- Integration guides

<https://wiki.egi.eu/wiki/AAI>

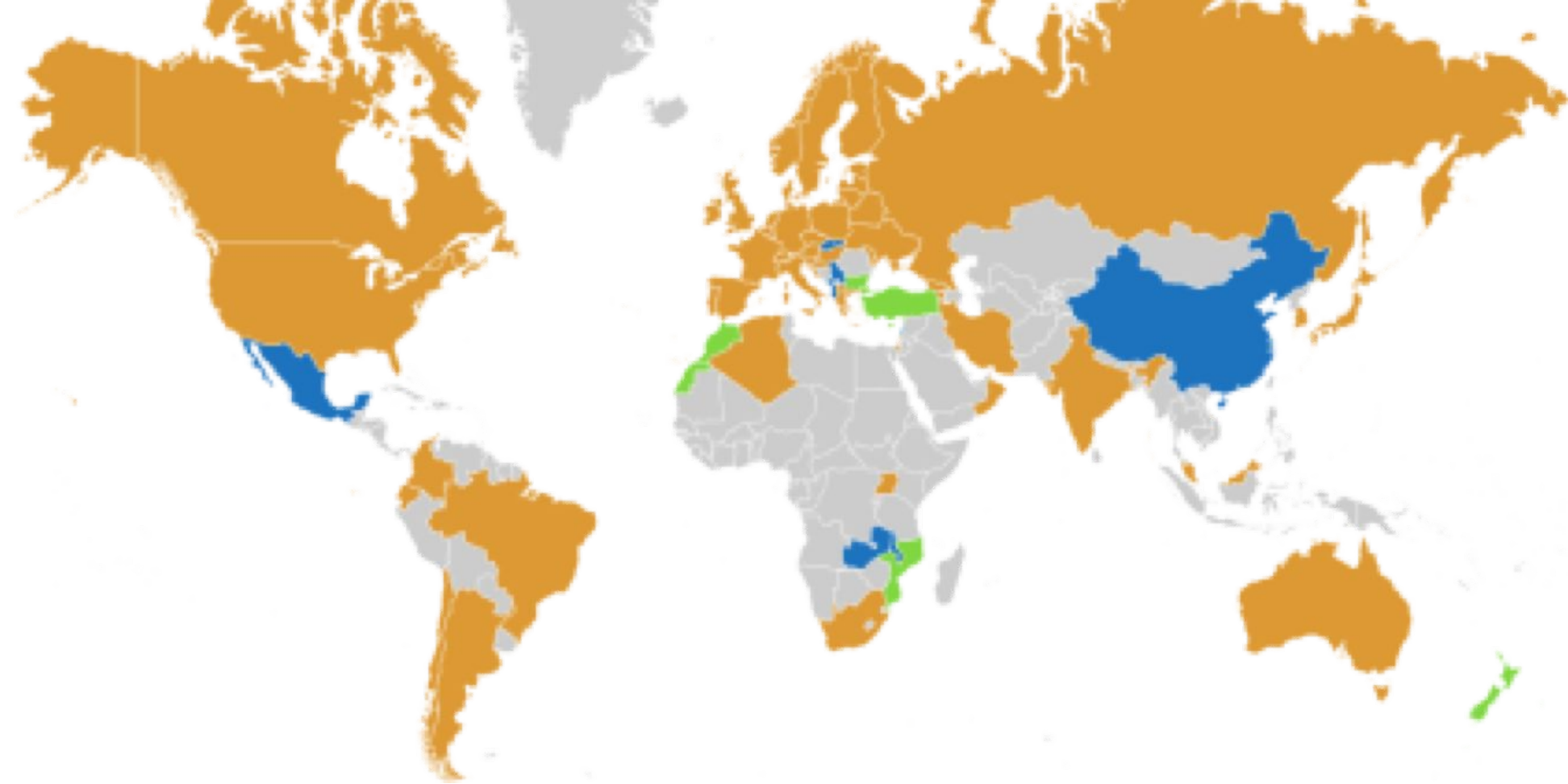
- Support for SAML/OpenID Connect Identity Providers
 - eduGAIN
 - Social media
 - ORCID
- Ability to create enrolment flows specific to a community's requirements
- Support for organising users in hierarchical groups
- Ability to associate certificate and ssh key information to researcher's federated identity
- Ability to enrich researcher's identity with community-specific attributes
- Direct (de)provisioning of information into an LDAP directory or VOMS
- Multipath delegation via OAuth 2.0 Token Exchange
 - Support for attenuation of rights/scopes

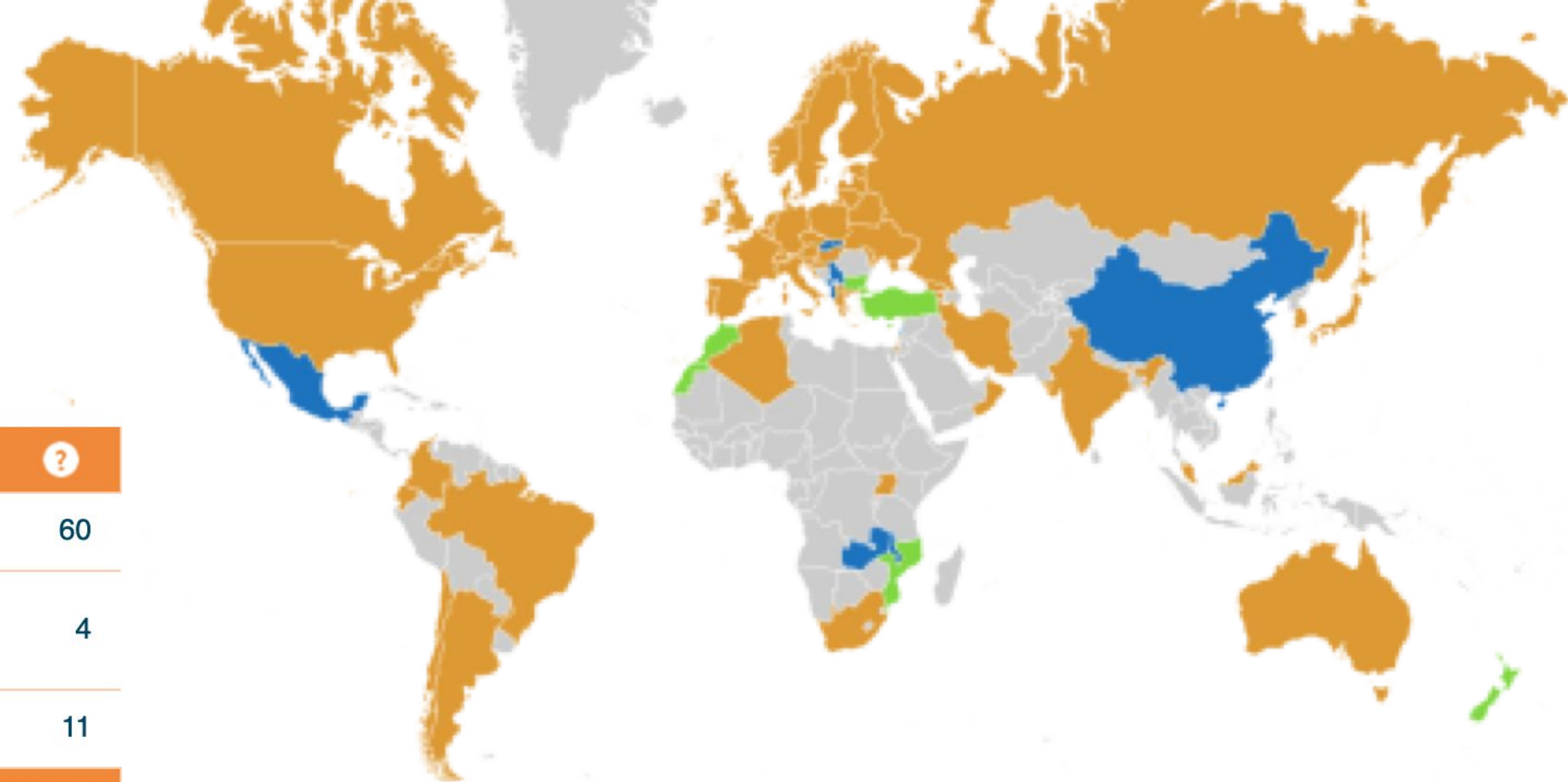


eduTEAMS

Virtual Teams Made Easy







Federations in eduGAIN ?	
--------------------------	--

Members	60
Voting-only Members	4
Candidates	11

Entities in eduGAIN ?	
-----------------------	--

All entities	5482
IdPs	3060
SPs	2422
Standalone AAs	4



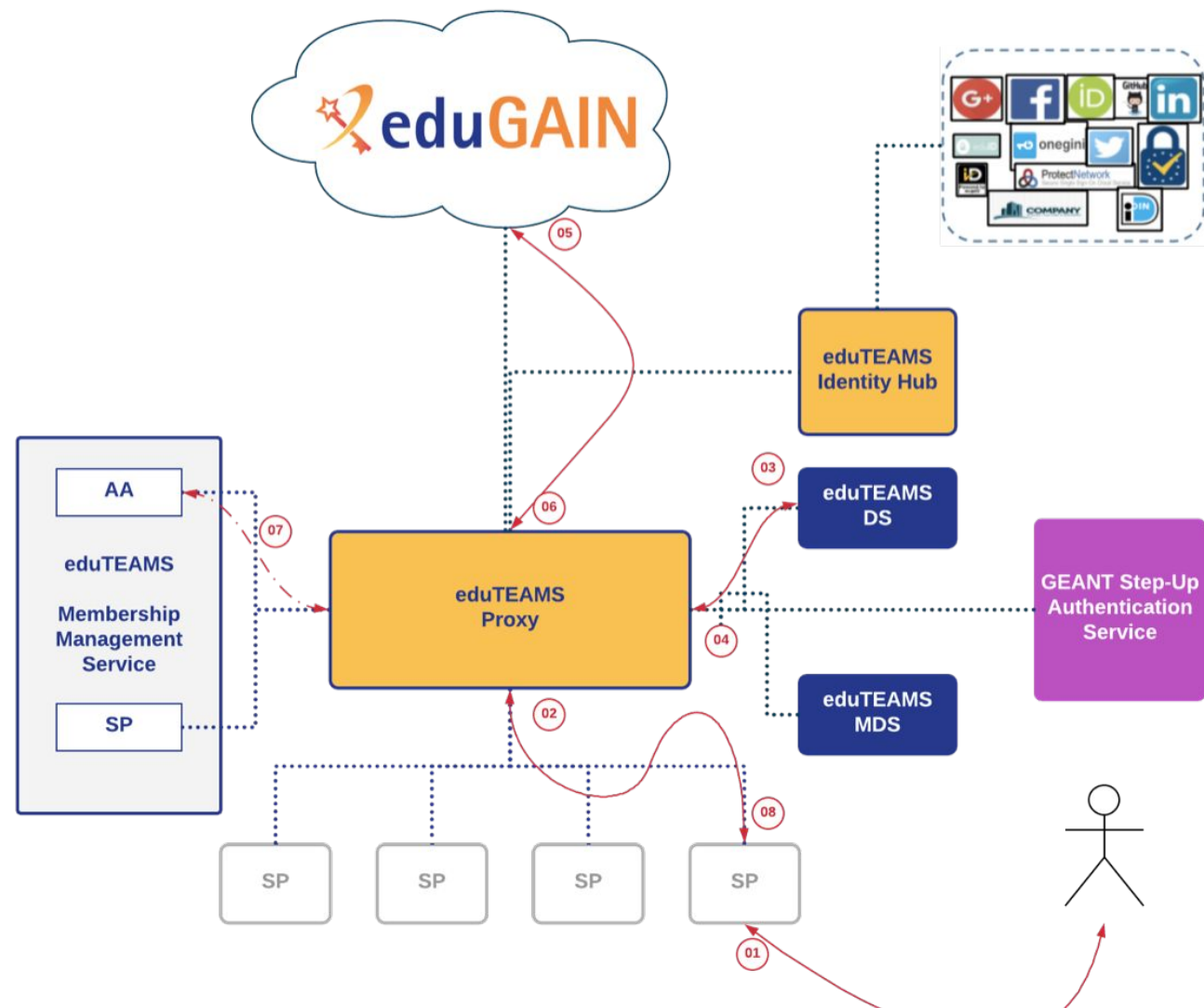
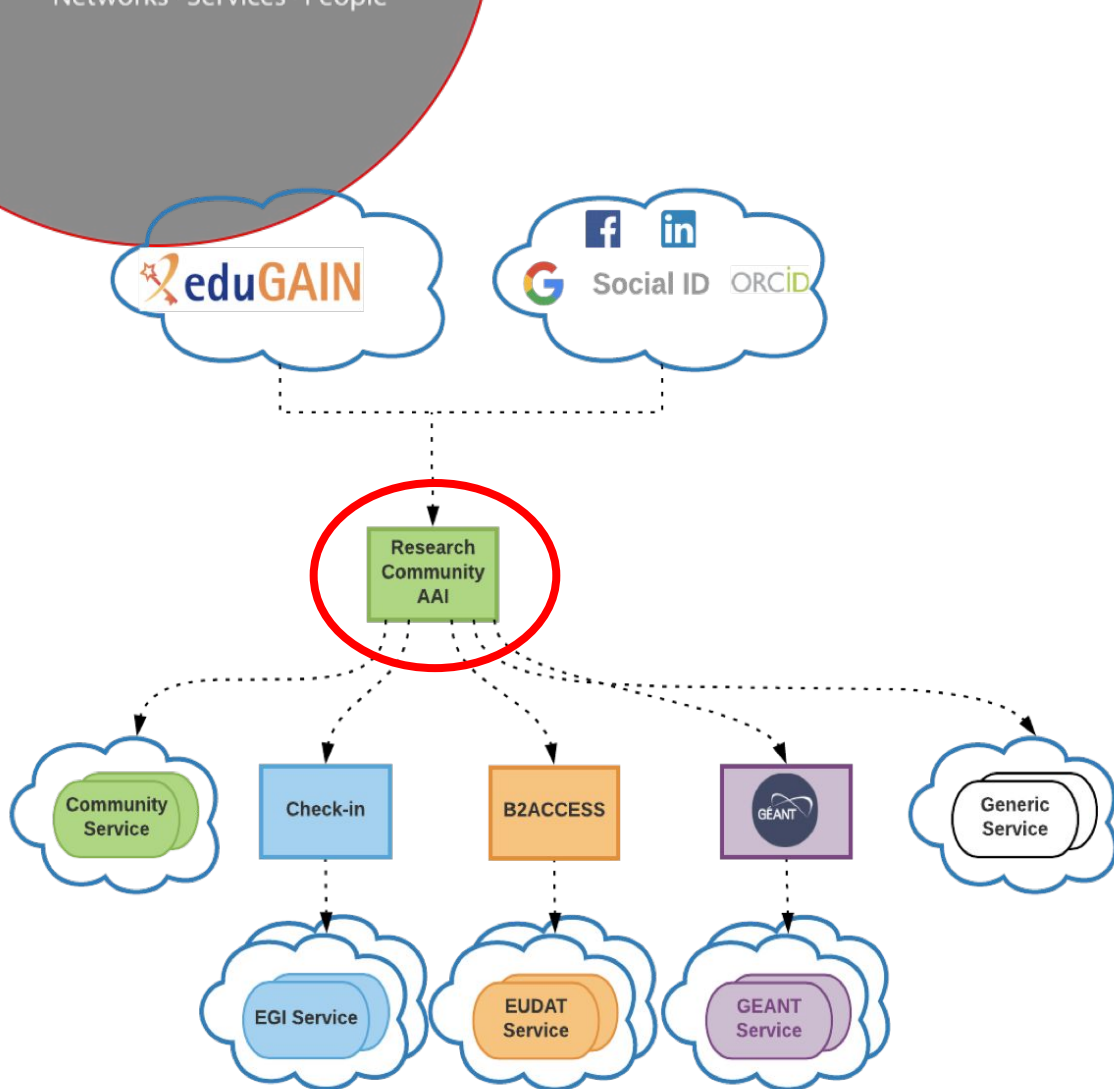
A service to enable use of
federated identity
management in research &
academic collaborations

- Full implementation AARC BPA
- Single- and multi-tenant options
- Sustainability and strategic partnerships

Components

- Proxy & Identity Hub
- Membership Management service
- Discovery Service
- Metadata Service
- Token Translation Service
- Second Factor Authentication (Pilot!)

- SAML & OIDC Support for SPs and IdPs
- Flexible management of group, roles and access rights
- User enrollment flows
- Account Linking



Service Managers

- A central point for the community to manage its user membership, to connect Identity Providers and Service Providers and to define and apply access and sharing policies
- Secure access and sharing of common resources and services
- Secure user authentication and identification
- Group and role management
- Virtual Organizations management
- **Options to manage VOs >>**

Users

- Sign in to services with existing identities via eduTEAMS
- First class support of eduGAIN Identity Providers
- Support for the Research and Scholarship entity category, Code of Conduct and Sirtfi to support scalable authorisation
- Support for a wide range of external Identity Providers, such as ORCID and Google
- Support for web and non-web based services - access to HTTP APIs
- Account linking

CREATE VIRTUAL ORGANISATIONS

The eduTEAMS Membership Management Services (MMS) provide you with the ability to create VOs, manage these VOs, invite users to collaborate, manage registration flows, organise user to groups and assign them roles and resource entitlements as needed within the collaborations. The eduTEAMS Platform has first class support of three of the most popular membership and group management systems. Choose the one that best fits your needs and start using eduTEAMS.



COmanage

Fully customizable user flows
Complex organizational structures
(support for hierarchical groups,
roles, etc.)
Rich support for a range of data
sources and targets

HEXAA

Very lightweight and intuitive user
interface
Service focused with built-in
support for service entitlements
Hook functionality, announcement
functionality

Perun

Feature rich, more control for the
service owners
Enhanced privacy controls
Sophisticated provisioning / de-
provisioning

eduTEAMS Service

- Shared platform that can be used by small - medium communities and the long tail of science
- Managed and operated by GEANT
- eduTEAMS branding & eduTEAMS community identifier
- eduTEAMS service policies
- Connected to EOSC (GEANT, EGI and EUDAT services)
- Onboarding of community specific services



eduTEAMS Service

- Shared platform that can be used by small - medium communities and the long tail of science
- Managed and operated by GEANT
- eduTEAMS branding & eduTEAMS community identifier
- eduTEAMS service policies
- Connected to EOSC (GEANT, EGI and EUDAT services)
- Onboarding of community specific services

eduTEAMS Dedicated

- Dedicated, white-label service offering, specific to a community
- Managed by the community, operated by GEANT
- Community branding & community specific identifier
- Community managed policies
- Can be connected to EOSC (GEANT, EGI and EUDAT services)
- Onboarding of community specific services

eduTEAMS Service

- Shared platform that can be used by small - medium communities and the long tail of science
- Managed and operated by GEANT
- eduTEAMS branding & eduTEAMS community identifier
- eduTEAMS service policies
- Connected to EOSC (GEANT, EGI and EUDAT services)
- Onboarding of community specific services

eduTEAMS Dedicated

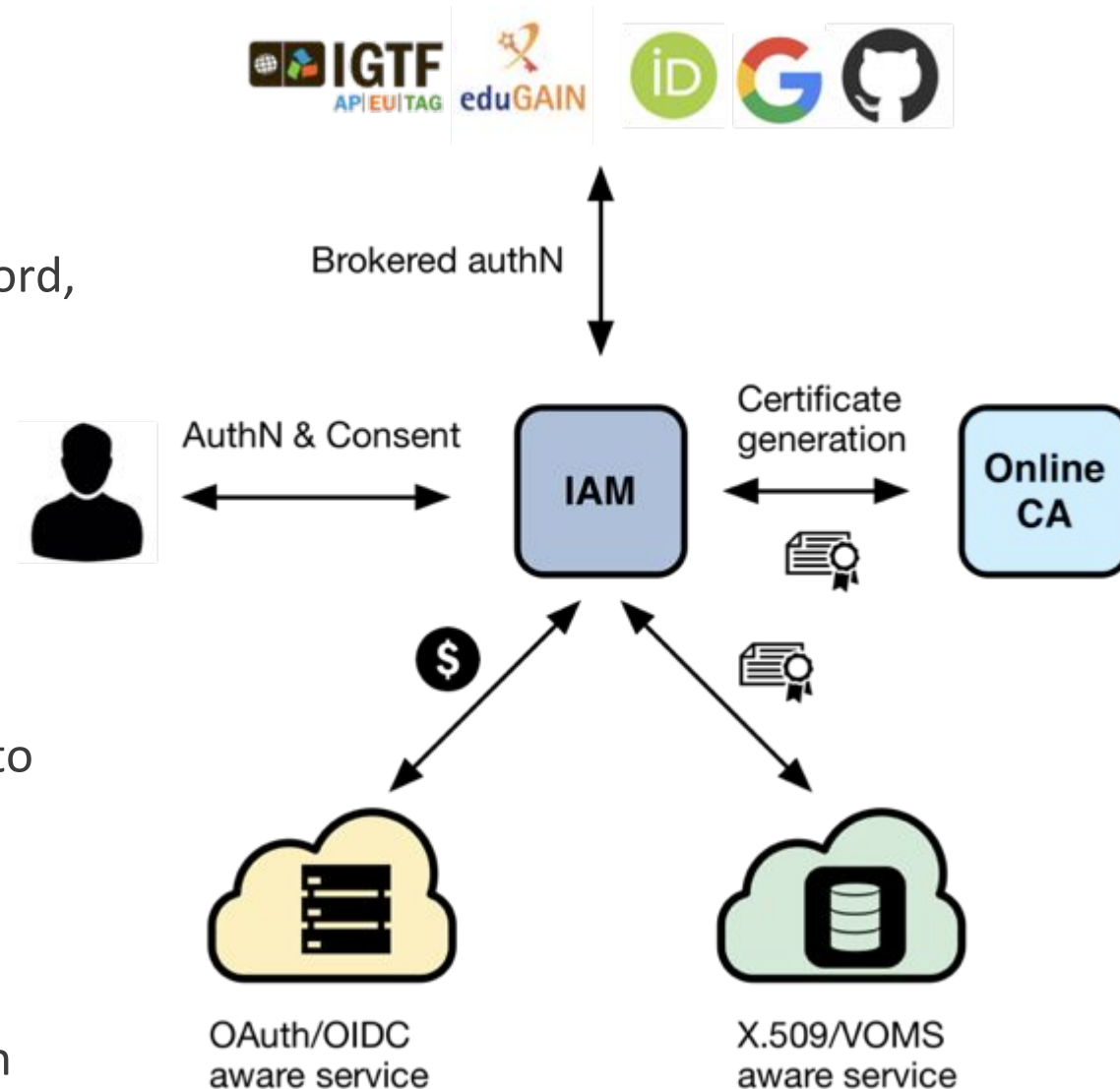
- Dedicated, white-label service offering, specific to a community
- Managed by the community, operated by GEANT
- Community branding & community specific identifier
- Community managed policies
- Can be connected to EOSC (GEANT, EGI and EUDAT services)
- Onboarding of community specific services

eduTEAMS Bespoke

- Bespoke solution with tailor-made functionality
- Ownership model depended on the solution, operated by GEANT
- Consultancy, development and hosting of the service.

Features

- **Flexible authentication support**
 - SAML, X.509, OpenID Connect, username/password, ...
- **Account linking**
- **Registration service** for moderated and automatic user enrollment
- **Enforcement of AUP acceptance**
- **Easy integration** in off-the-shelf components thanks to **OpenID Connect/OAuth**
- **VOMS support**, to integrate existing VOMS-aware services
- **Self-contained**, comprehensive AuthN/AuthZ solution

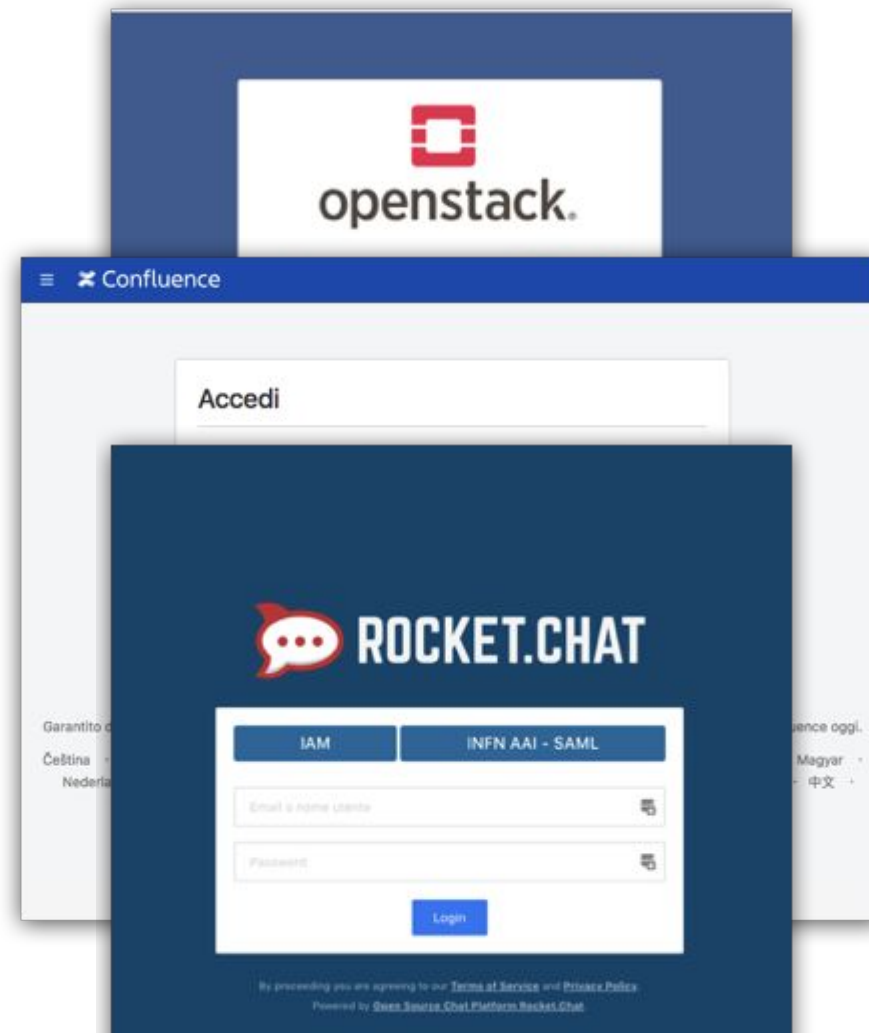


Easy integration with services

Standard OAuth/OpenID Connect enable **easy integration** with off-the-shelf services and libraries.

We have successfully integrated IAM with minimal effort with:

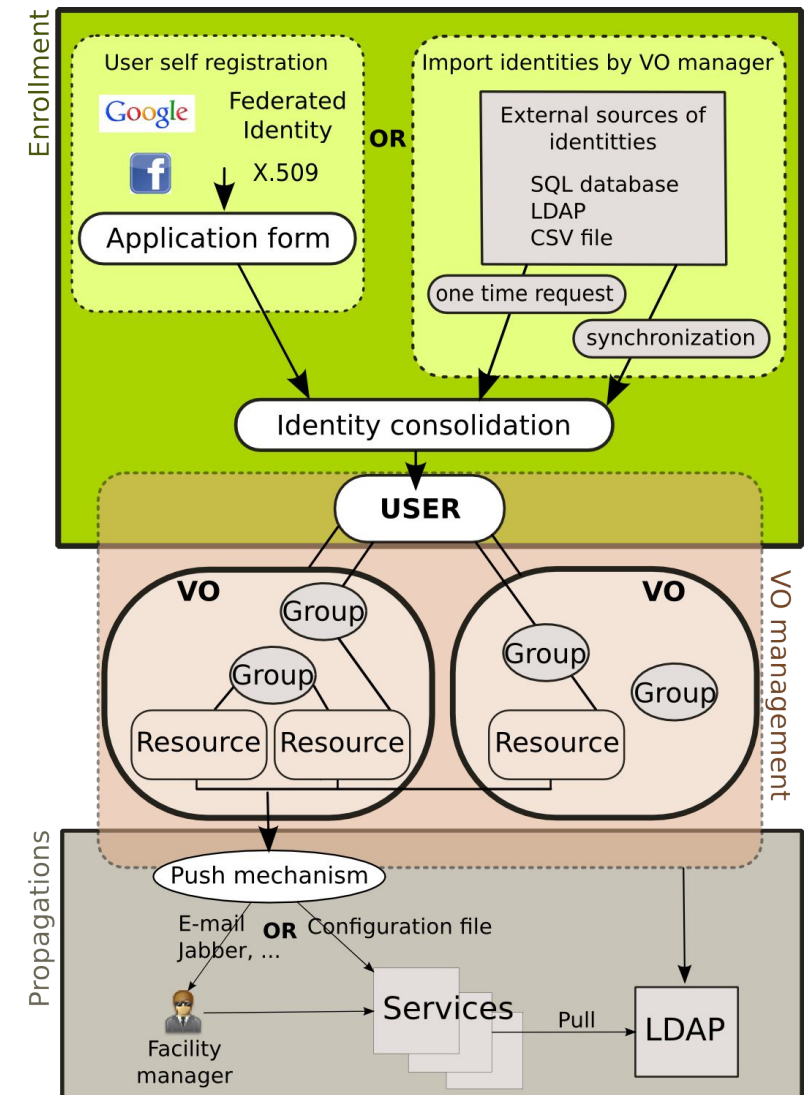
- Openstack
- Atlassian JIRA & Confluence
- Kubernetes
- Moodle
- Rocketchat
- Grafana
- JupyterHub



Useful references

- IAM @ GitHub: <https://github.com/indigo-iam/iam>
- IAM documentation: <https://indigo-iam.github.io/docs>
- WLCG AuthZ WG Demos:
<https://indico.cern.ch/event/791175/attachments/1806605/2948665/demos.mp4>
(IAM starts at minute 46)
- IAM in action video: <https://www.youtube.com/watch?v=1rZlvJADOnY>
- Contacts:
 - andrea.ceccanti@cnafr.infn.it
 - enrico.vianello@cnafr.infn.it
 - indigo-aai.slack.com

- Feature rich identity and access management
- Manages
 - Users, identities and attributes
 - VOs, groups and registrations
 - Access control for services
 - Provisioning / deprovisioning
- Strong support for rights delegation
- Designed to fit in existing infrastructure
- Doesn't do
 - Authentication, proxy
 - Credential storage



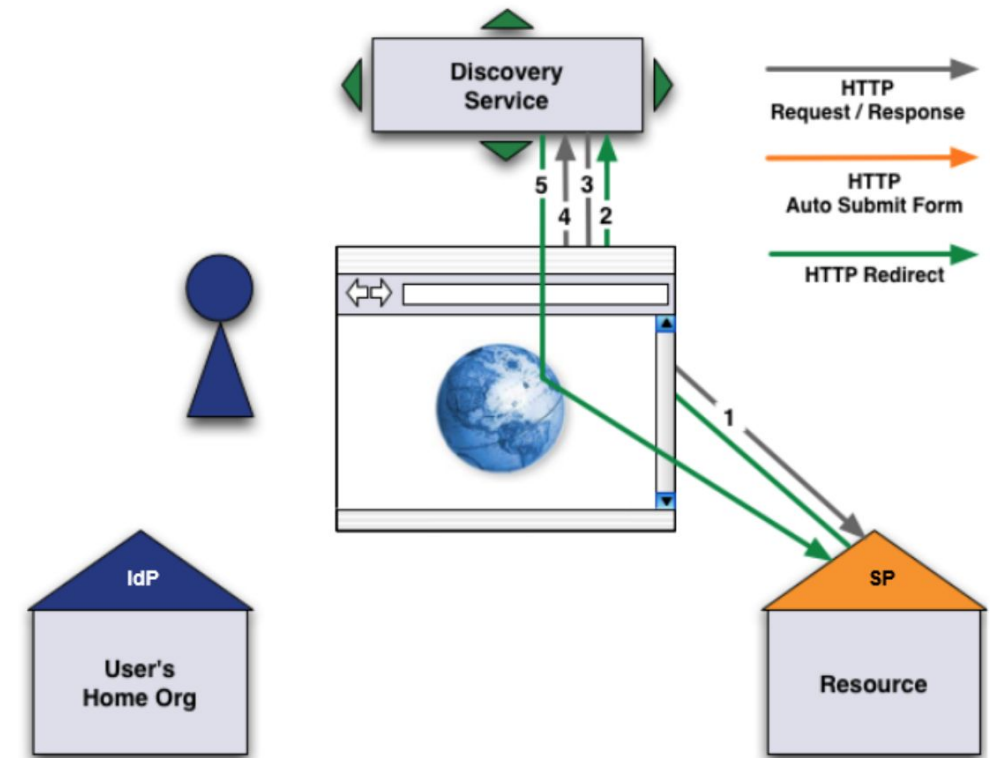
- Deployments
 - ELIXIR AAI
 - BBMRI AAI
 - Life-science AAI pilot
 - Czech national e-infrastructure
- In EOSC-hub
 - Part of eduTEAMS service
 - Part of EGI Check-in service
- OpenSource
 - <https://github.com/CESNET/perun>
- <https://perun-aai.org/>

Service Provider integration flows - demos

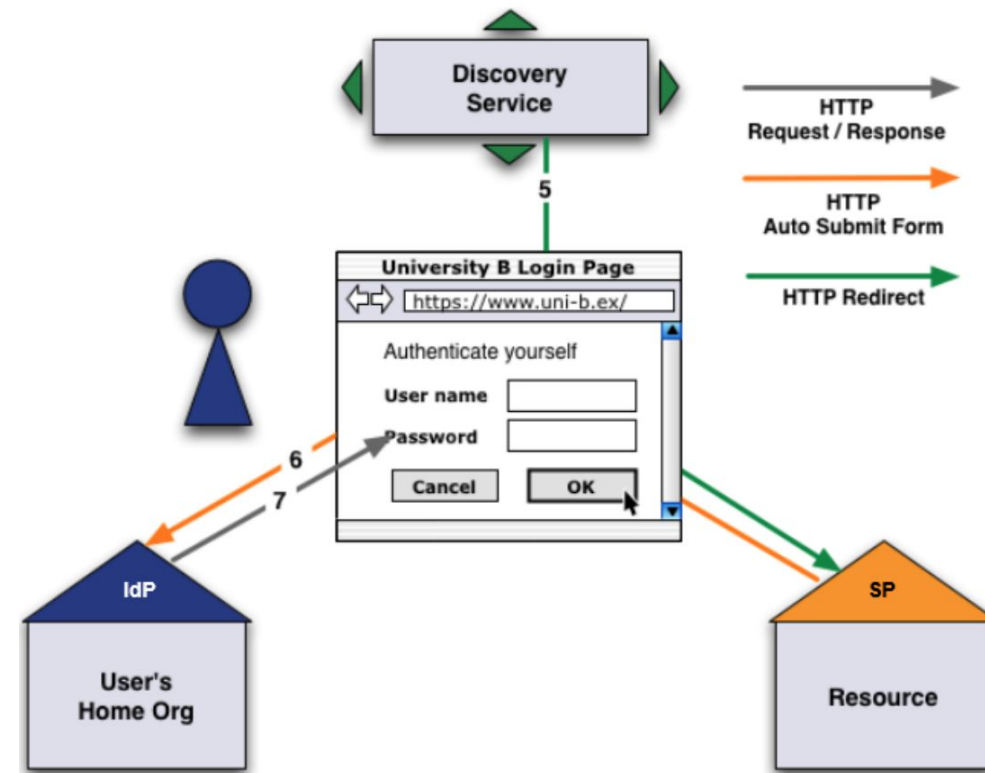
Use case: “*Web-based single sign-on (SSO)*”

- SAML 2.0
- OpenID Connect
- General recipe: SPs need to provide the following information to their IdP:
 - name of the service
 - short description
 - list of required user attributes
 - privacy statement
 - technical contact
 - security contact
 - protocol-specific endpoints

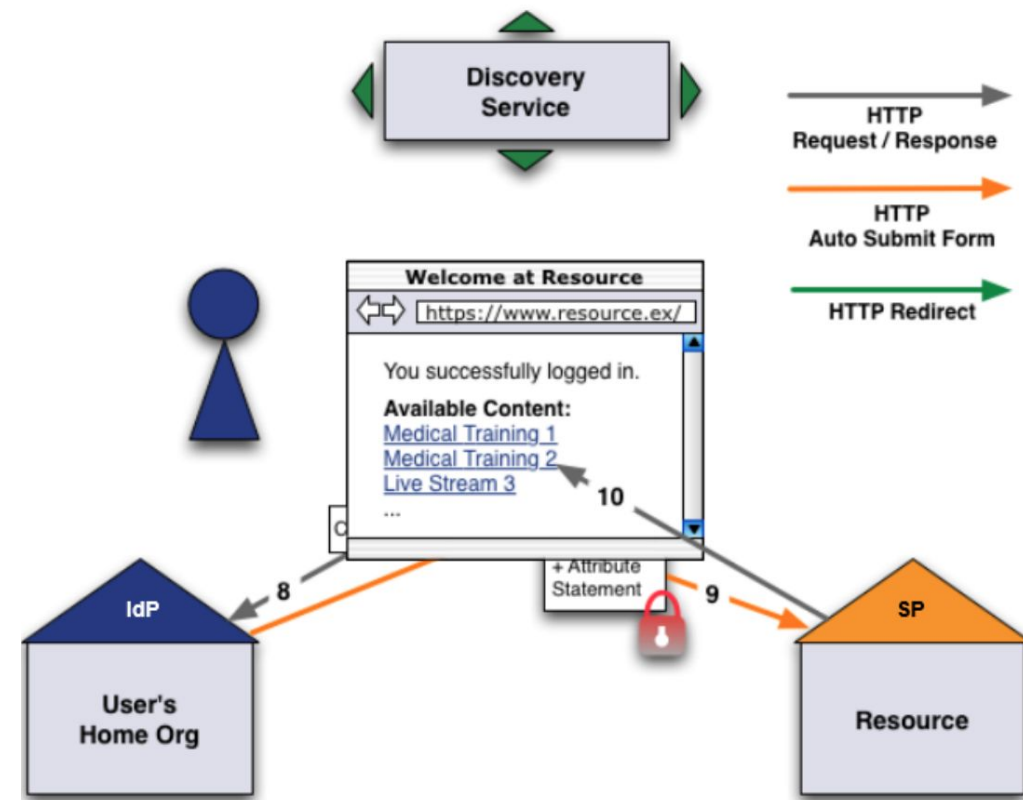
1. The user opens a web browser and accesses the Service Provider
2. The user is redirected to the Discovery Service by the Service Provider. Consequently, the web browser sends a new request to the Discovery Service
3. The Discovery Service responds with the web page that allows the user to select an Identity Provider
4. On the Discovery Service page, the user submits the Identity Provider selection
5. The Discovery Service sends a redirect to the SP return destination, including the IdP selection



5. The browser is redirected to the Service Provider by the Discovery Service
6. The session initiator of the Service Provider creates an authentication request and returns it within an auto-submit-post-form to the browser. The browser posts the SAML AuthN Request automatically to the Identity Provider
7. The Identity Provider checks the authentication request. Because the user hasn't been authenticated, the Identity Provider sends a redirect to the appropriate login page (usually: Username/Password)



8. The user types their username and password credentials and submits them to the Identity Provider
9. The Identity Provider verifies the credentials. If authentication succeeds, the IdP issues an assertion for the SP and returns it within an autosubmit-post-form to the browser. The web browser automatically posts the SAML Assertion to the Service Provider with the use of JavaScript. The Service Provider processes the SAML assertion including the authentication and attribute statements
10. Finally, the Service Provider starts a new session for the user and redirects the user to the previously requested resource. Now, the user is authenticated and gains access to the resource



SAML authentication relies on the use of metadata

- You as a SP and the R/e-Infra proxy (IdP) need to exchange metadata in order to know and trust each other
- Metadata include information such as the location of the service endpoints that need to be invoked, as well as the certificates that will be used to sign SAML messages.
- Format based on the XML-based SAML 2.0 specification.
- Can be automatically generated by all major SAML 2.0 SP software solutions (e.g. Shibboleth, SimpleSAMLphp, and mod_auth_mellon)
- **Important: You need to make your metadata available over HTTPS using a browser-friendly SSL certificate**

```
<?xml version="1.0"?>
<md:EntityDescriptor xmlns:md="urn:oasis:na
  <md:SPSSODescriptor protocolSupportEnumer
    <md:Extensions>
      <mdui:UIInfo xmlns:mdui="urn:oasis:na
        <mdui:DisplayName xml:lang="en">RCI
        <mdui:Description xml:lang="en">RCI
      </mdui:UIInfo>
    </md:Extensions>
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.c
        <ds:X509Data>
          <ds:X509Certificate>MIIDPzCCAiegA
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.c
        <ds:X509Data>
          <ds:X509Certificate>MIIDPzCCAiegA
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:SingleLogoutService Binding="urn:oa
    <md:AssertionConsumerService Binding="u
    <md:AssertionConsumerService Binding="u
    <md:AssertionConsumerService Binding="u
    <md:AssertionConsumerService Binding="u
  </md:SPSSODescriptor>
  <md:ContactPerson contactType="technical"
```

- Shibboleth-SP
 - Implements the SP component in a SAML communication:
 - i.e. protects a web application behind it and deals with SAML communication
 - There are two parts:
 - A background service or "daemon" process (shibd)
Keeps states, evaluates protocol messages
 - A Webserver module (mod_shib)
Protects Locations/Directories, defines Access Rules
 - Available through:
 - Linux Debian/RPM packages, Binaries for Windows, OSX
- SimpleSAMLphp
- mod_auth_mellon

eduPersonUniqueid	urn:oid:1.3.6.1.4.1.5923.1.1.1.13	uniqueid@scope
givenName	urn:oid:2.5.4.42	John
sn	urn:oid:2.5.4.4	Doe
displayName	urn:oid:2.16.840.1.113730.3.1.241	John Doe
mail	urn:oid:0.9.2342.19200300.100.1.3	john.doe@example.org
eduPersonEntitlement	urn:oid:1.3.6.1.4.1.5923.1.1.1.7	<NAMESPACE>:group:<GROUP>[:<SUBGROUP>]...[:role=<ROLE>]#<GROUP-AUTHORITY> <NAMESPACE>:res:<RESOURCE>[:<CHILD-RESOURCE>]...[:act:<ACTION>[,<ACTION>]...]#<AUTHORITY>
eduPersonScopedAffiliation	urn:oid:1.3.6.1.4.1.5923.1.1.1.9	faculty@example.org

- Introduction to OpenID Connect/OAuth2
- Web-based SSO - OpenID Connect Authorization code flow
- OAuth2 Device code flow
- Supporting multiple OPs - ESACO


Token exchange allows to request and obtain access/refresh tokens from a client B using the token of client A

- Multipath delegation
- Attenuation of rights/scopes


<https://tools.ietf.org/html/draft-ietf-oauth-token-exchange-16>

OAuth2 - Token Exchange Example

Service





EGI Check-in Demo Client
OIDC Client for demonstration purposes
▼ More Info



Access Token

eyJraWQiOiJvaWRjliwiYW:

 Copy

 Revoke

Permissions: read your affiliation within a domain, read your unique identifier, log in using your identity, read your email address, read your rights to resources, read your basic profile info
Issued: Today at 2:59 PM
Expire: Today at 3:59 PM

An access token has been issued for client “EGI Check-in Demo Client” with the following scopes:

- openid
- profile
- email
- eduperson_entitlement
- eduperson_scoped_affiliation
- eduperson_unique_id



EGI Check-in Demo Token Exchange Client

OIDC Client for demonstration purposes

Grant Types

- ☒ authorization code
- ☐ client credentials
- ☐ password
- ☐ implicit
- ☐ redelegation
- ☒ token exchange
- ☐ device

OAuth2 Token Exchange Example

Using the access token from client “EGI Check-in Demo Client” as subject token, the user is going to exchange it to claim a new access token from client “EGI Check-in Demo Token Exchange Client”, which has the token exchange grant type enabled. The scope set of the new token can be the same or less than the subject token.

Note: Exchanged tokens have the same type (an access token for an access token and a refresh token for a refresh token).

```
$ export client_B_id=...
```

```
$ export client_B_secret=...
```

```
$ export access_token_A=...
```

```
$ curl -u "${client_B_id}":"${client_B_secret}"
```

```
-X POST "https://aai-dev.egi.eu/oidc/token"
```

```
-d "grant_type=urn:ietf:params:oauth:grant-type:token-exchange&
```

```
subject_token=${access_token_A}&
```

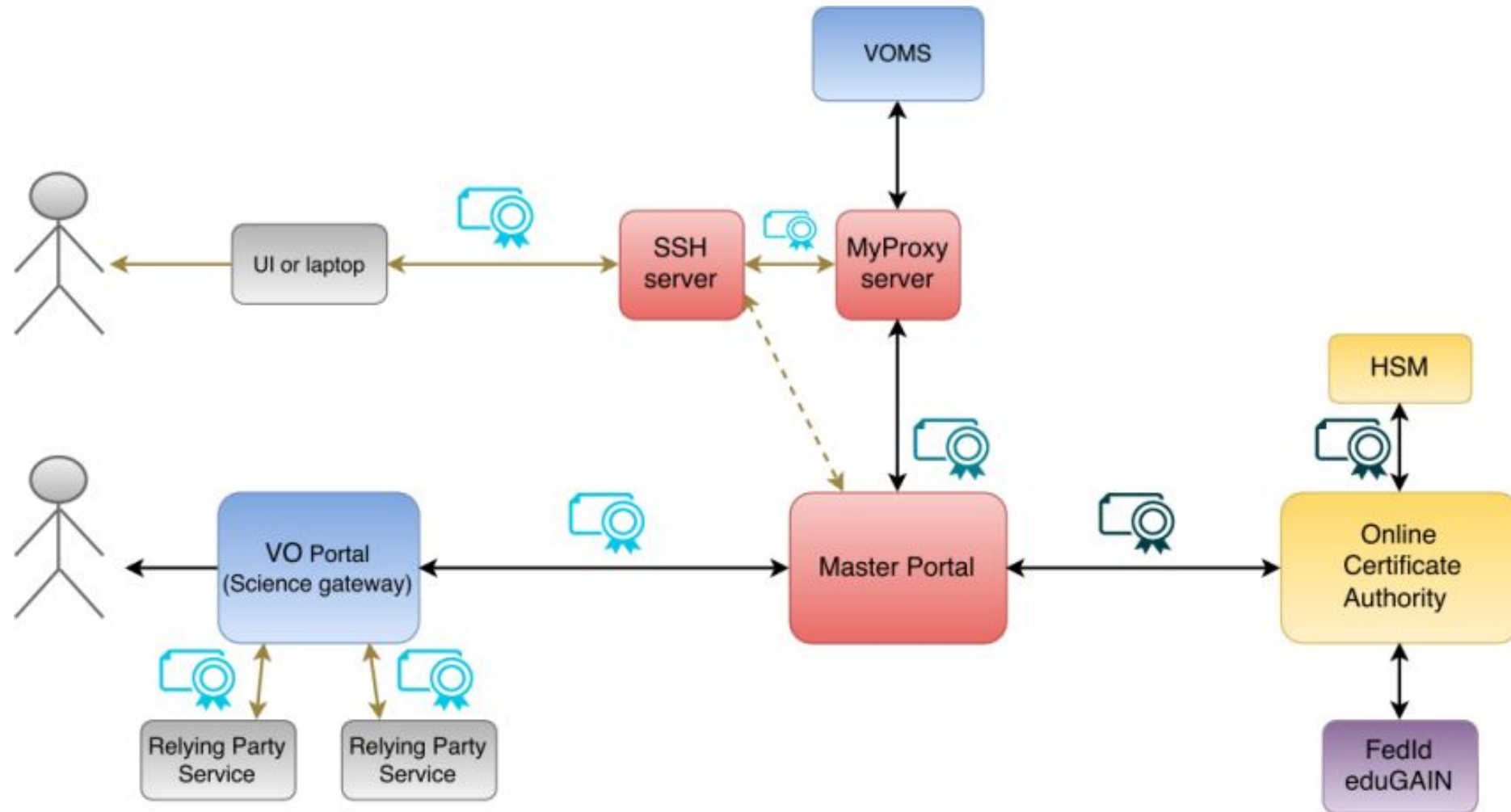
```
subject_token_type=urn:ietf:params:oauth:token-type:access_token&
```

```
scope=openid%20profile%20email"
```

39

Use case: *“Implementing a VO portal (Science Gateway) to support access to services relying on X509 authentication using proxy certificates from the RCauth.eu online CA”*

- VO portals do not directly contact the RCauth CA, but use an intermediate service, a so-called Master Portal, which handles most of the complexity
- Master Portal is an OpenID Connect Provider, with an integrated protected endpoint for obtaining proxy certificates



Register a new client at a Master Portal at the /register endpoint:

- E.g. for the EGI development instance
 - <https://masterportal-pilot.aai.egi.eu/mp-oa2-server/register>
- or the EGI production instance
 - <https://aai.egi.eu/mp-oa2-server/register>

Welcome to the Master Portal Client Registration Page

This page allows you to register your client with the Master Portal that supports the OIDC/OAuth 2. To get your client approved, please fill out the form below. Your request will be evaluated for approval. For more information, please make sure you read the [Registering a Client with an OAuth 2 server](#) document.

Client Name:

Contact email:

Home URL:

Refresh Token lifetime: (in seconds - leave blank for no refresh tokens.)

Callback URLs:

Put your callbacks here, one per line.

- *NOTE: Make sure to store the `client_id` and `client_secret` in a secure place.*
- In order to get the client approved, send an email to the administrator of the Master Portal to request client approval. For EGI use [EGI CheckIn Support](#)

Registration Successful!

Here is your client identifier:

`myproxy:oa4mp,2012:/client_id/807cb841d200a6f097e64d87d45a7ff`

Here is your client secret:

`UCqOa-sdSBIHRKOaM_h_huU2O6Vl2gTpoOphYBIciG3I5PmFEc93erH_EQ70ZAlUX0IeBv2uU0fbFwBmV5Ya-9ZDMfidLiDbA1SfH9in_QoI9tfa48HmOW18ubcOLTPHgMtrJ5G8PnNuq0hQB3E6daRXwSqe9V6O14C7jRwI7KlUR2_-XyzStn-XvilKKm5E1Tx7HRb7VSB9mp9tQ1VULgG3uMX7l88v5c3sNxVVJ_dVMNgIKfBfkm-VqjKkFzYr27hFf5L3Ak-jlc9jatETzkmKOq56r7r7UTA4Rwt-`

IMPORTANT NOTE: It is the client's responsibility to store the identifier and secret. Your client will need to use it as needed to identify itself. Please keep these in a safe location. If you lose the secret, you will have to re-register. Be sure you copy the secret without line breaks (which some browsers will insert) or you will get an invalid secret.

An administrator will contact you once your registration request is approved. You cannot use this identifier code until you have been approved by the administrator.

Obtaining a proxy certificate from the RCauth.eu CA via the Master Portal follows the standard OIDC Authorization Flow:

1. VO portal initiates the flow by sending the user (browser redirect) to the /authorize endpoint on the Master Portal. Parameters (see Client Requests Authorization)
 - a. Optionally the VO portal can redirect the user to a specific IdP by also sending an idphint parameter. This is a RCauth / MasterPortal extension (see also Master Portal Internals#The_IdP_Hint).
2. When the authorization flow succeeds, an authorization grant is sent via the browser as code parameter to the redirect_uri (see Authorization Response).
3. The VO portal uses the authorization grant to obtain an access_token, an id_token and optionally a refresh_token from the /token endpoint (back-channel to the OIDC provider). Parameters (see Access Token Request)

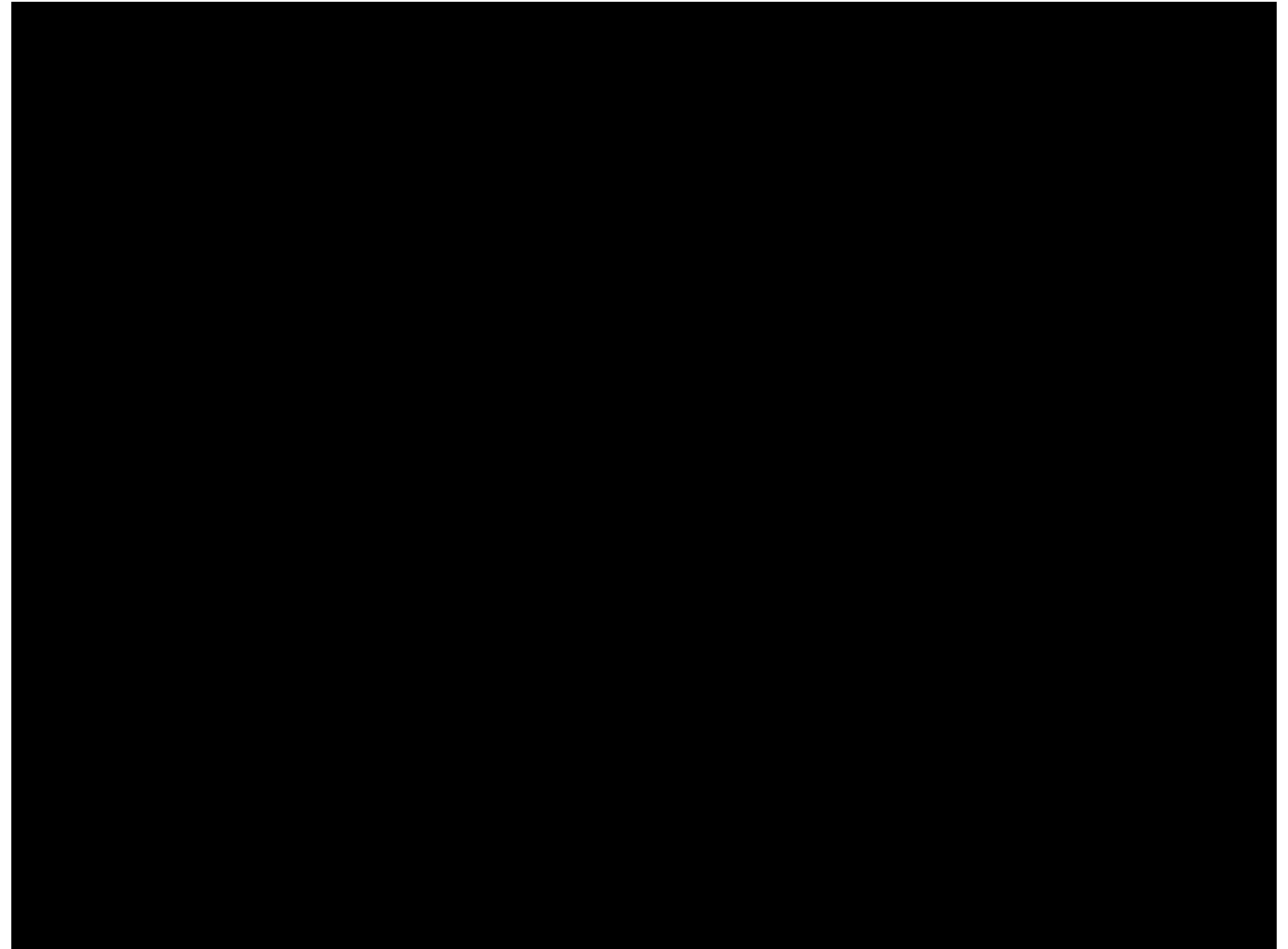
4. When successful, the response is a JSON including the `access_token`, `id_token` (and the `refresh_token` when configured). (see Access Token Response)
5. The VO portal can optionally access the `/userinfo` endpoint using the `access_token` (back-channel to the OIDC provider). Note that it can get the same information directly from the `id_token`.
Parameters (see UserInfo Request)
6. The VO portal can now obtain a proxy certificate from the `/getproxy` endpoint using the `access_token`, and authenticating using its `client_id` and `client_secret` (back-channel). Parameters (see OAuth for MyProxy GetProxy Endpoint):
7. The response will consist of the proxy certificate chain in a single PEM.

Getting proxy certificates from the command line using SSH key authentication:

- User uploads SSH public key to MasterPortal or to COmanage:
 - <https://aai.egi.eu/sshkeys/>
 - requires login via RCauth.eu to obtain username
-
- User makes sure MyProxy has a long-lived proxy, can use 'vo-portal'
 - <https://aai.egi.eu/vo-portal/>
 - Needs to do this ± once a week
- User obtains proxy certificate via dedicate SSH host:
 - `ssh proxy@ssh.aai.egi.eu > /tmp/x509up u$(id -u)`

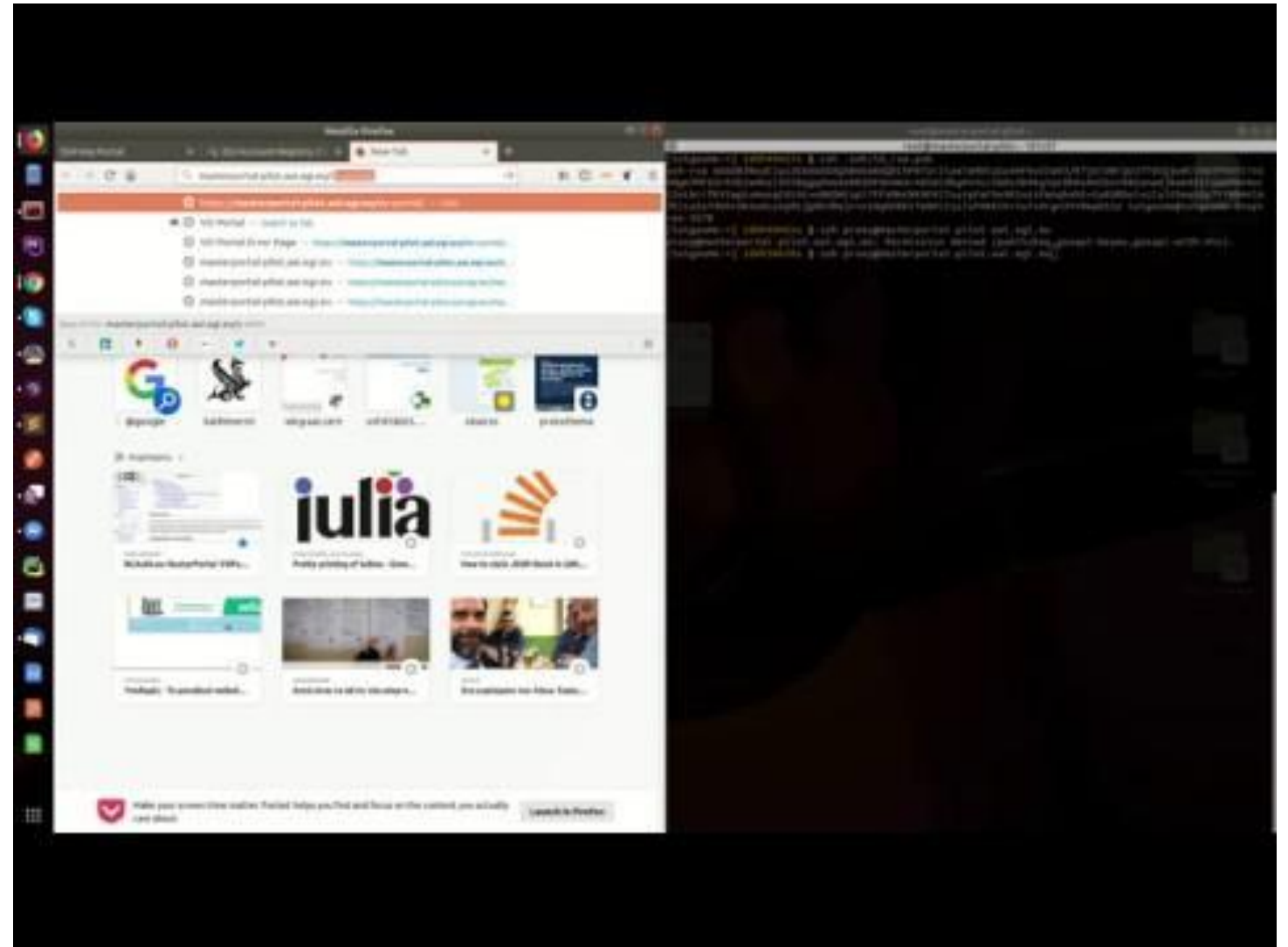
Demo:

- Obtaining proxy certificates through VO Portal (Science Gateway)



Demo:

- Obtaining proxy certificates from command line using SSH key authentication



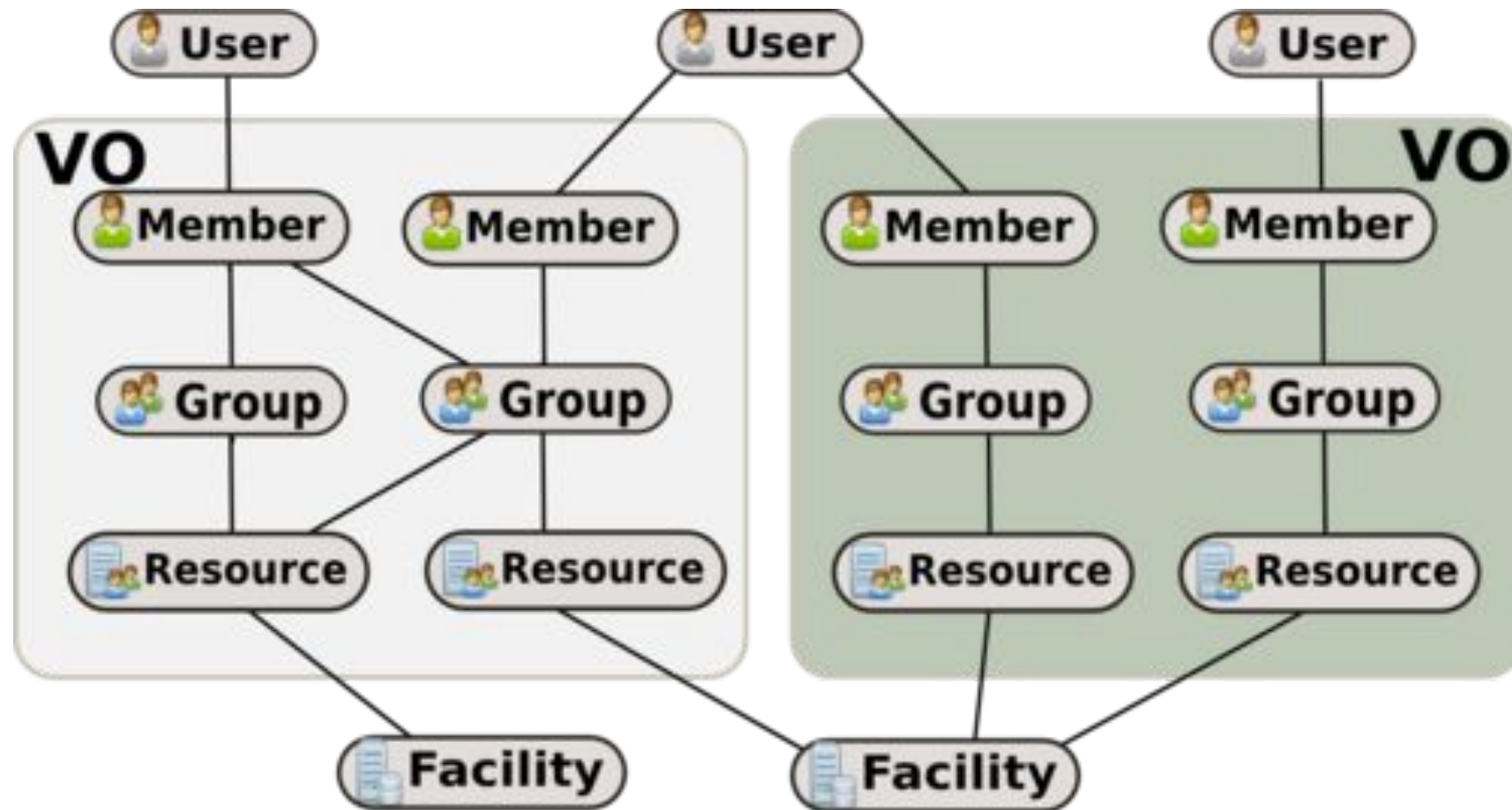
- <https://rcdemo.nikhef.nl/>
- <https://rcdemo.nikhef.nl/demogsiftp/>
- https://rcdemo.nikhef.nl/demobasic/oidc_getproxy_demo_source.php
- https://drive.google.com/open?id=1T_b4U3RgUI8IzjCq9gE1asqtHN6BefK (Full list of Demo videos)

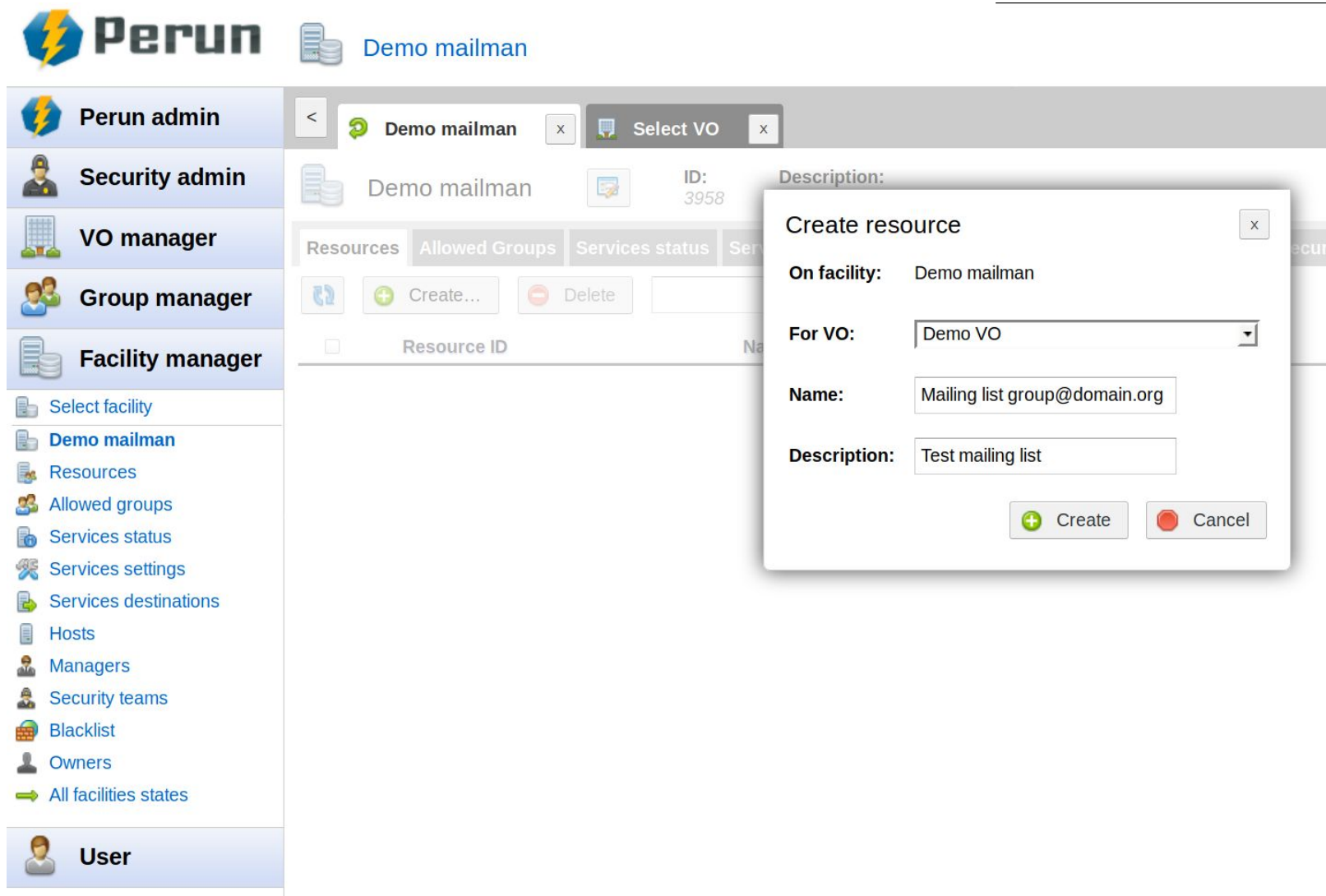
Use case: *“Provision or de-provision user account to services base on his/hers roles or group membership without any direct user interaction.”*

Example services: mailing lists, cloud management systems (OpenStack), unix account & ssh keys distribution, VOMS

- This demo will show (de-)provisioning in Perun system
- Using simple custom service connectors

Configuration within Perun





The screenshot displays the Perun web interface. On the left is a navigation sidebar with the following menu items: Perun admin, Security admin, VO manager, Group manager, Facility manager, Select facility, Demo mailman, Resources, Allowed groups, Services status, Services settings, Services destinations, Hosts, Managers, Security teams, Blacklist, Owners, All facilities states, and User. The main content area shows the 'Demo mailman' facility selected. A 'Create resource' dialog box is open in the foreground, containing the following fields: 'On facility' (set to 'Demo mailman'), 'For VO' (set to 'Demo VO'), 'Name' (set to 'Mailing list group@domain.org'), and 'Description' (set to 'Test mailing list'). The dialog has 'Create' and 'Cancel' buttons at the bottom right. In the background, the 'Resources' tab is active, showing a table with columns for 'Resource ID' and 'Name'. A 'Demo mailman' resource with ID '3958' is visible. The top of the interface includes the Perun logo and a 'Demo mailman' breadcrumb.

Create resource: Assign and configure services



x

Selected service:

mailman

☐ Show assigned Add RemoveFinish 

Already assigned: mailman

 Save Fill Remove

<input type="checkbox"/>	Attr ID	Name	Value	Description
<input checked="" type="checkbox"/>	2520	Maling list manager's e-mail	boss@domain.org	Maling list manager's e-mail.
<input checked="" type="checkbox"/>	1840	Name of mailing list	group	Name of the mailing list which is represented by this resource.



Demo VO

Name: [Mgr. Slávek Licehammer](#)

Role: PERUN / SECURITY ADMIN



Logout



Perun admin



Security admin



VO manager

 Select VO Demo VO Members Groups Resources Applications Application form Resource tags Resources state Settings Managers External sources

Group manager



Facility manager



User



Demo VO



Demo VO

ID:
3797Short name:
demo-vo

Overview

Members

Groups

Resources

Applications

Application form

Settings

Managers

External sources



Quick tools



Add member...

Add new member into your VO. Candidates can be searched for in VO's external sources or among user already existing in Perun.



Create service member...

Create new member which represent service account (account usually used by more users with separate login and password).



Invite member...

Invite person to become member of your Virtual organization.



Add manager...

Add new manager which can manage your VO in Perun.



Create group...

Create new group in your VO.




Add member to resource...


Add selected member to specific resource (grant some type of access to Facility resources).





Statistics


 **Perun**


Demo VO


Name: [Mgr. Slávek Licehammer](#)
Role: PERUN / SECURITY ADMIN  Logout


 **Perun admin**


 **Security admin**


 **VO manager**


 [Select VO](#)


 [Demo VO](#)


 [Members](#)


 [Groups](#)


 [Resources](#)


 [Applications](#)


 [Application form](#)


 [Resource tags](#)


 [Resources state](#)


 [Settings](#)

 [Managers](#)

 [External sources](#)

 **Group manager**




 **Facility manager**

 **User**

< Demo VO x

Demo VO ID: 3797 Short name: demo-vo

Overview Members Groups Resources Applications Application form Settings Managers External sources

  Delete  Filter

	Resource ID	Name	Facility	Tags	Description	Count: 1
<input type="checkbox"/>	11265	Mailing list group@domain.org	Demo mailman		Test mailing list	



Demo VO > Resources

Name: [Mgr. Slávek Licehammer](#)

Role: PERUN / SECURITY ADMIN



Logout



Perun admin



Security admin



VO manager

 [Select VO](#) [Demo VO](#) [Members](#) [Groups](#) [Resources](#) [Applications](#) [Application form](#) [Resource tags](#) [Resources state](#) [Settings](#) [Managers](#) [External sources](#)

Group manager



Facility manager



User



Demo VO



Mailing list group@domain...



Mailing list group@domain.org

ID:
11265Description:
Test mailing list[View facility details >>](#)

Assigned groups

Assigned services

Service settings

Group settings


Member settings

Tags



+ Add...

- Remove

 Filter

Group ID

Name

Description

Count: 0

Resource has no groups assigned.





Demo VO > Resources


Name: [Mgr. Slávek Licehammer](#)


Role: PERUN / SECURITY ADMIN





 **Perun admin**


 **Security admin**


 **VO manager**


 [Select VO](#)


 [Demo VO](#)


 [Members](#)


 [Groups](#)


 [Resources](#)


 [Applications](#)


 [Application form](#)


 [Resource tags](#)


 [Resources state](#)


 [Settings](#)

 [Managers](#)

 [External sources](#)

 **Group manager**

 **Facility manager**

 **User**

<

Demo VO x

Mailing list group@domain... x


>


✕


Mailing list group@domain.org ID: Description: [View facility details >>](#)

Assign group

x


 Add

 Cancel

 Filter

☐ Configure group(s) before assign

<input type="checkbox"/>	Group ID	Name	Description	Count: 3
<input type="checkbox"/>	12117	members	Group containing VO members for VO Demo VO	
<input checked="" type="checkbox"/>	12118	My first reserach group	Group for demo purposes	
<input type="checkbox"/>	12119	Other group	Another demo group	


Perun

Demo VO > Resources

Name: [Mgr. Slávek Licehammer](#)
Role: PERUN / SECURITY ADMIN

Logout

Perun admin

Security admin

VO manager

Select VO

Demo VO

Members

Groups

Resources

Applications

Application form

Resource tags

Resources state

Settings

Managers

External sources

Group manager

Facility manager

User

<

Demo VO

Mailing list group@domain...

>

✕

Mailing list group@domain.org

ID: 11265

Description: Test mailing list

[View facility details >>](#)

Assigned groups

Assigned services

Service settings

Group settings

Member settings

Tags

↺

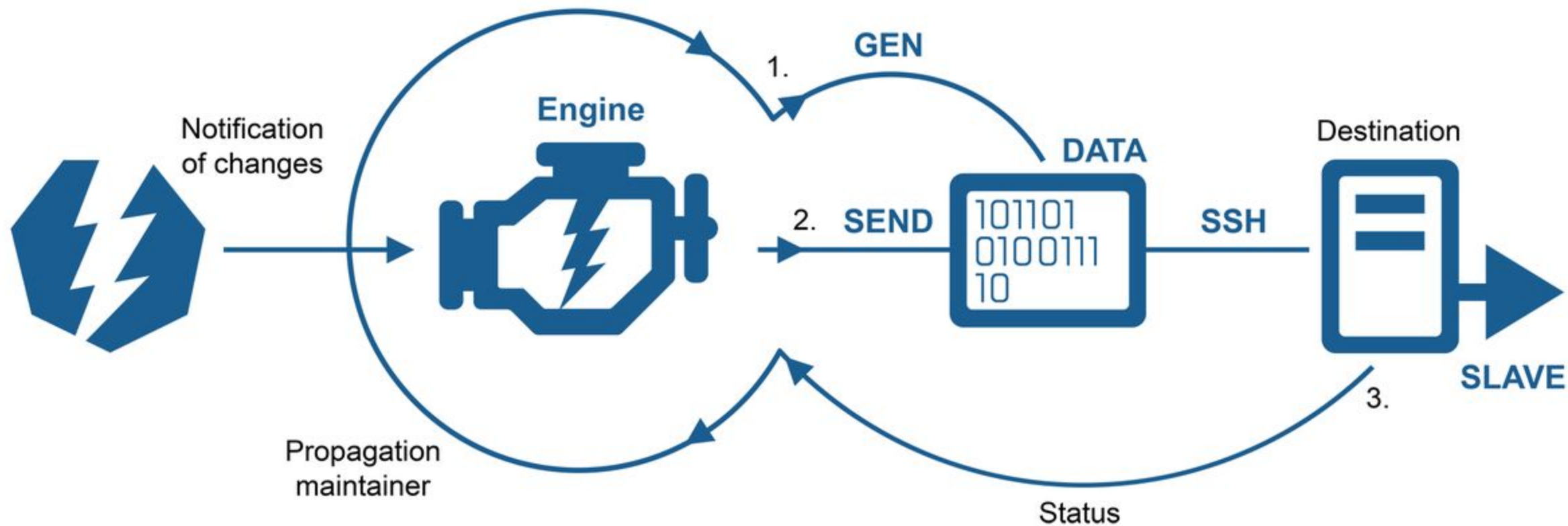
+ Add...

− Remove

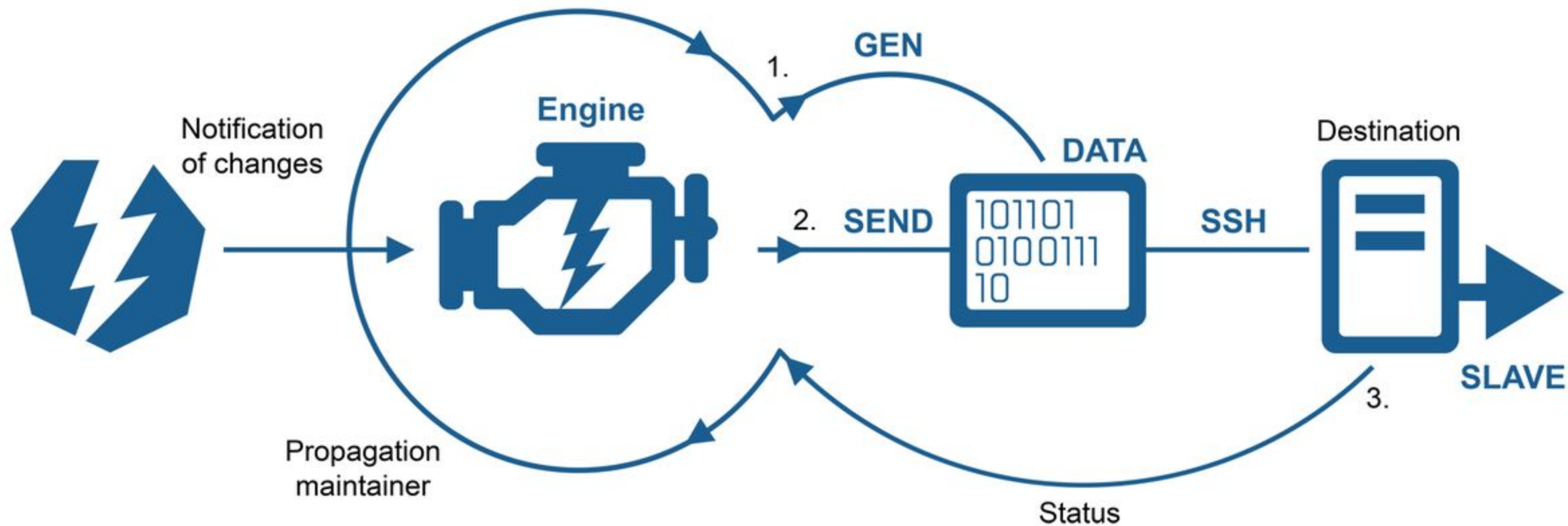
Filter

<input type="checkbox"/>	Group ID	Name	Description	Count: 1
<input type="checkbox"/>	12118	My first reserach group	Group for demo purposes	

Configuration on a service



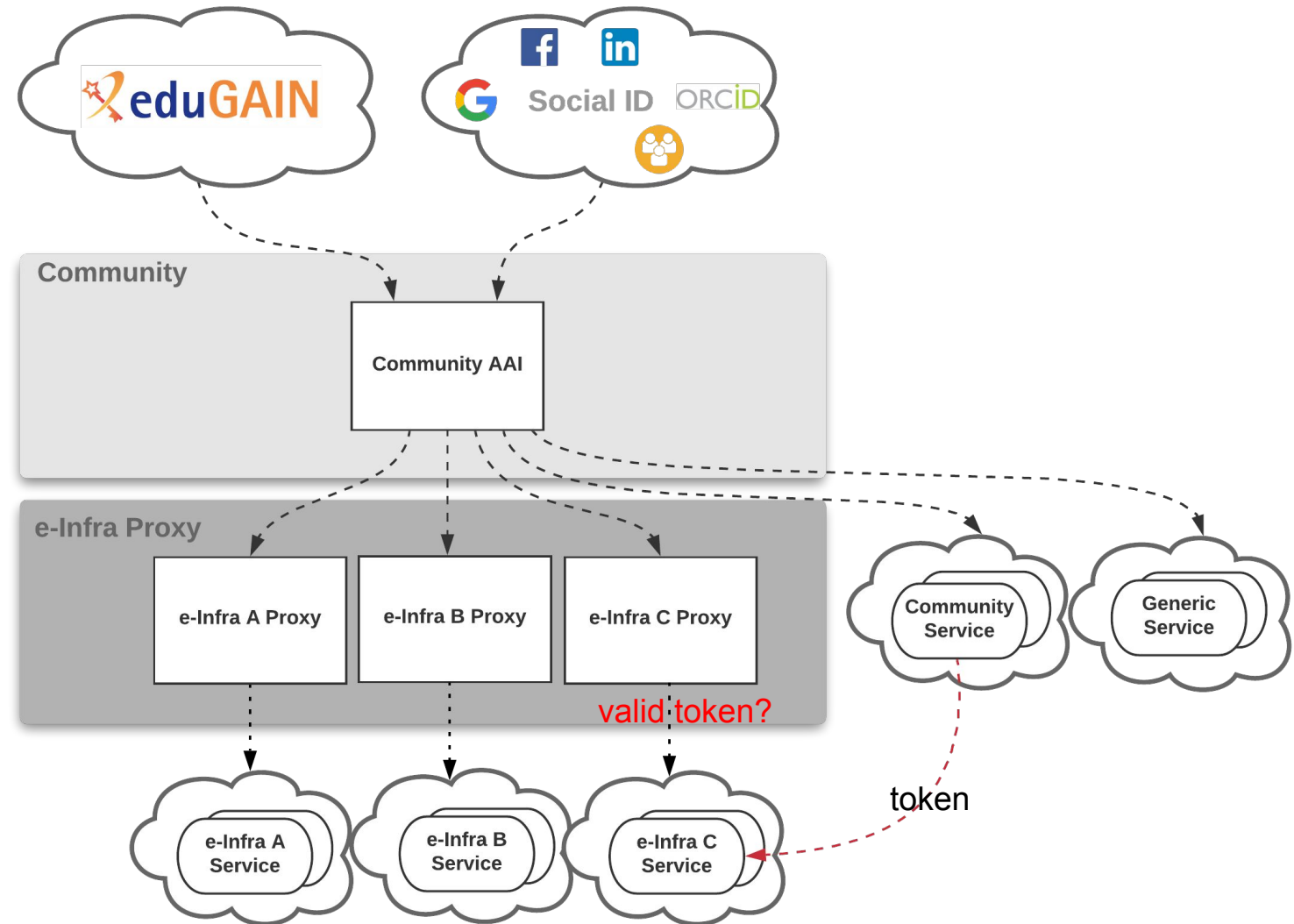
- Install Perun slave package for managing mailman
 - # apt-get install perun-slave-process-mailman
- Allow Perun to configure the service
 - # echo 'from="perun.cesnet.cz", command="/opt/perun/bin/perun" ssh-rsa AAAAB3NzaC... perun@cesnet.cz' >> ~/.authorized_keys



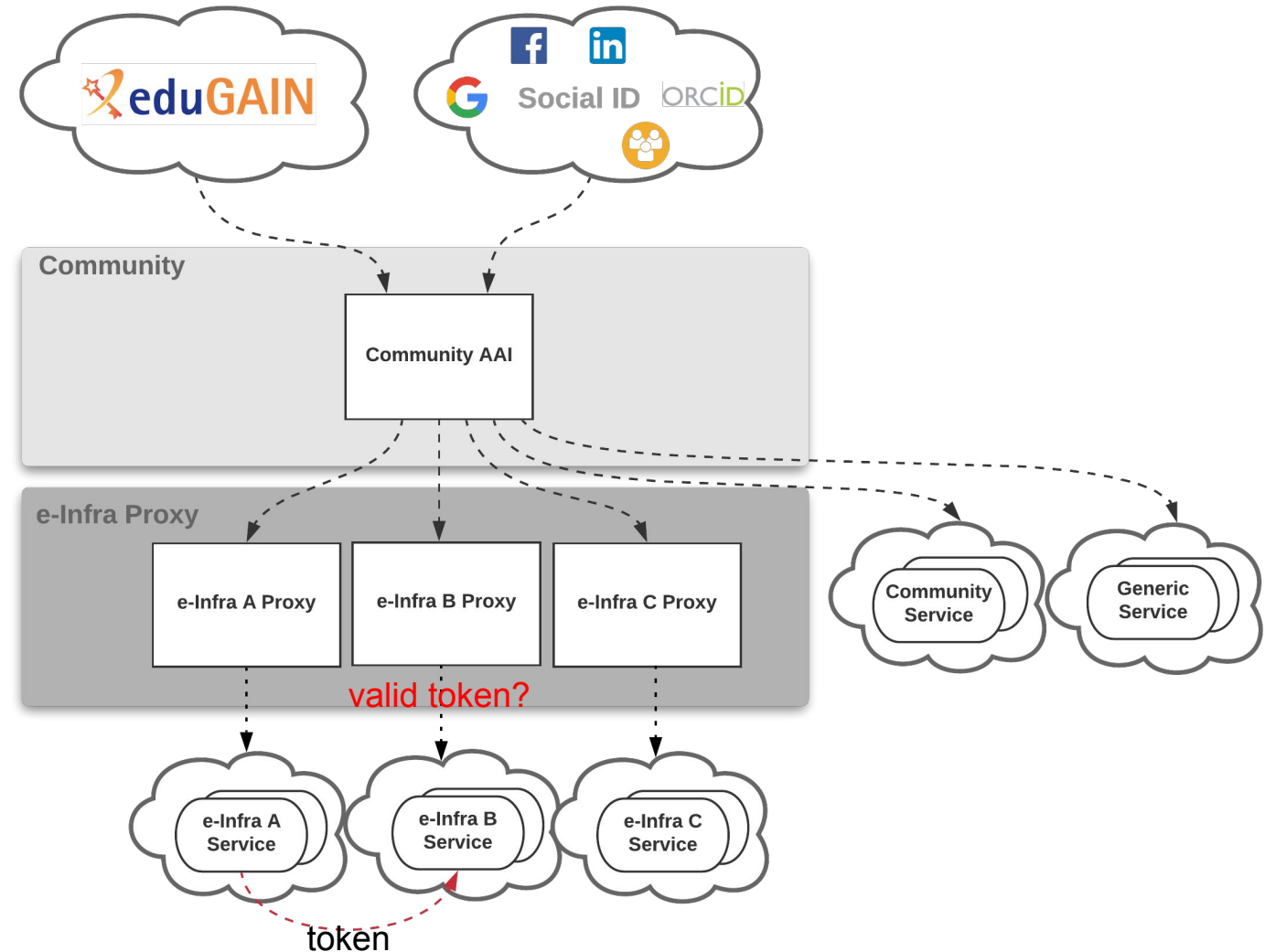
EOSC-hub AAI status & next steps

- **Multiple user registrations:** Users need to register with multiple AAI services due to differences in their Acceptable Use Policies (AUP).
- **Multiple IdP discovery steps:** As users access services protected by different AAls, they may need to select their identity provider (IdP)
 - **But** they don't need to enter their credentials again due to the Single Sign-On session in effect

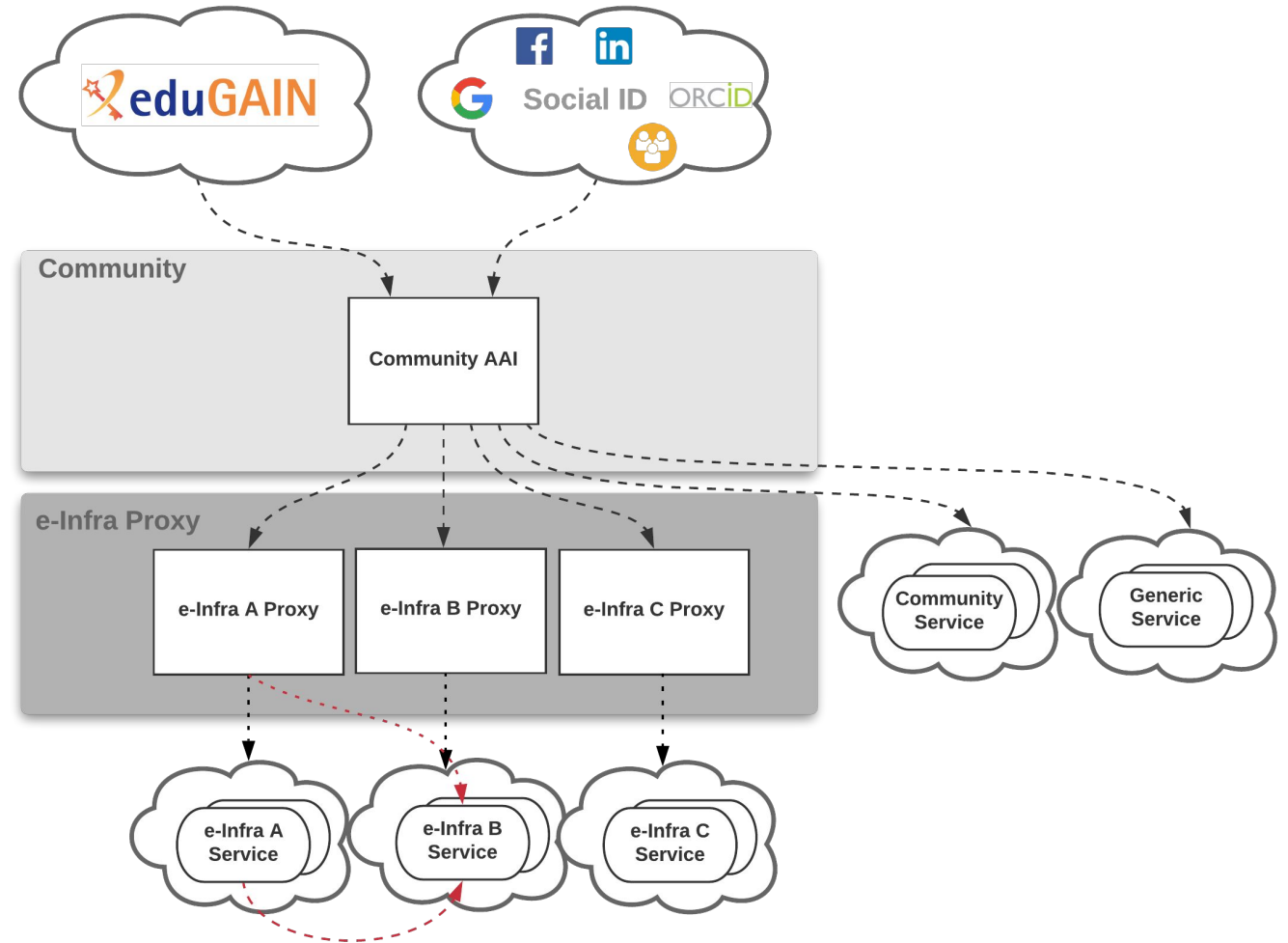
- **OAuth2 token validation:**
Existing implementations of OAuth2-based Authorisation Servers do not support the validation of tokens issued by a different Authorisation server.
- E.g. community service accessing e-Infra service on behalf of user



- **OAuth2 token validation:**
Existing implementations of OAuth2-based Authorisation Servers do not support the validation of tokens issued by a different Authorisation server.
- *E.g. Community service accessing e-Infra service on behalf of user*

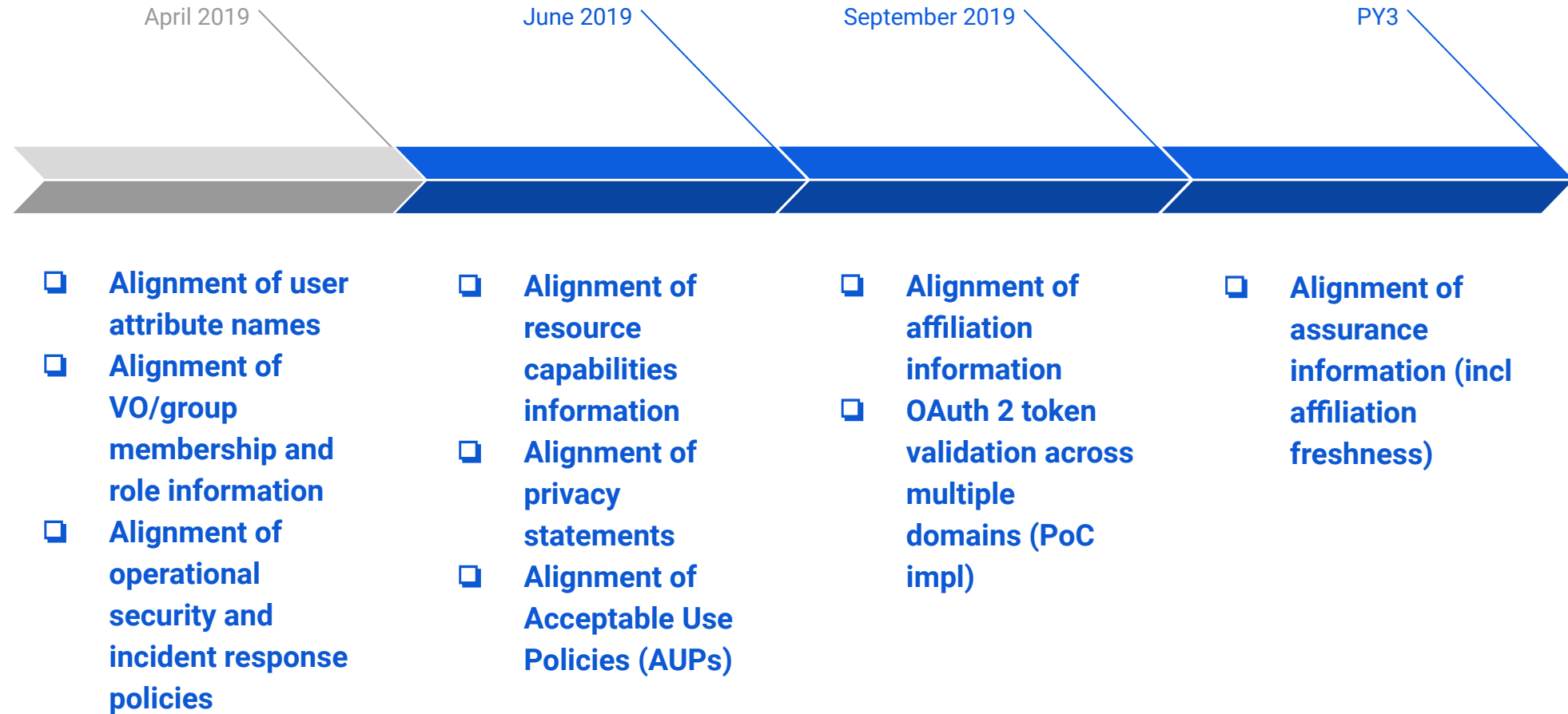


- **OAuth2 token validation workaround:** Services need to connect to different Authorisation Servers instead of relying on a single SP Proxy
... but
 - Requires additional integration effort from services
 - Cannot scale



	EOSC-hub AAI
Alignment of user attribute names	✓
Alignment of VO/group membership and role information	✓
Alignment of resource capabilities information	M18
Alignment of affiliation information	M21
OAuth 2 token validation across multiple domains (PoC impl)	M21
Alignment of assurance information (incl affiliation freshness)	PY3

	EOSC-hub AAI
Alignment of privacy statements	M18
Alignment of operational security and incident response policies	✓
Alignment of Acceptable Use Policies (AUPs)	M18



Please, take a few moments to provide us feedback about the training event:

<https://bit.ly/2leajPq>

Thank you for your attention!

Questions?



EOOSC-hub

Contact

 eosc-hub.eu  [@EOOSC_eu](https://twitter.com/EOSC_eu)



This material by Parties of the EOOSC-hub Consortium is licensed under a Creative Commons Attribution 4.0 International License.