EGI-Engage

Pilot services and best practices to enable federated AAI solutions released

M3.4

Date	06 June 2016
Activity	JRA1.1 Authentication and Authorization Infrastructure
Lead Partner	GRNET
Document Status	FINAL
Document Link	https://documents.egi.eu/document/2825

Abstract

This document presents the requirements, technical architecture, workflows and bundle of enabling technologies that have been defined, selected and tested to extend the current EGI AAI with new capabilities, this originating a new AAI service named “EGI AAI CheckIn”. The pilot has been successfully executed through the integration and testing of various technical solutions. The tangible outcome of this activity is a new set of AAI capabilities allowing easier access to EGI services through institutional user accounts that will be introduced into the EGI service catalogue after user validation.

The EGI-Engage project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 654142 http://go.egi.eu/eng

COPYRIGHT NOTICE

DELIVERY SLIP

	Name	*Partner/Activity*	*Date*
From:	Christos Kanellopoulos	GRNET/JRA1.1	6/06/2016
Moderated by:	Malgorzata Krakowian	EGI.eu/NA1	
Reviewed by	Alessandro Paolini Fernando Aguilar	EGI.eu/SA1 UNICANN/SA2	23/06/2016
Approved by:	AMB and PMB		30/06/2016

DOCUMENT LOG

Issue	*Date*	*Comment*	*Author/Partner*
v1	6/6/2016	Initial version submitted for external review	C.Kanellopoulos/GRNET Nicolas Liampotis/GRNET
v2	26/6/2016	Update after the external review	C.Kanellopoulos/GRNET
FINAL	29/6/2016	Typos fixed	D.Scardaci/EGI.eu

TERMINOLOGY

A complete project glossary is provided at the following page:
http://www.egi.eu/about/glossary/

ACRONYMS

Acronym	*Definition*
CSR	Certificate Signing Request
DS	Delegation Service
IdP	Identity Provider
LDAP	Lightweight Directory Access Protocol

LoA	Level of Assurance
OIDC	OpenID Connect
OA4MP	OAuth for MyProxy
SAML	Security Assertion Markup Language
TTS	Token Translation Service
VHO	Virtual Host Organization
VOMS	Virtual Organization Membership Service

Contents

Executive summary

Task JRA1.1 started its activity in May 2015 focussing on collecting requirements from EGI users and other stakeholders, in order to understand their short and medium-term requirements, and establishing relationships with AARC, GN4, EUDAT2020 and PRACE, in order to work together towards an interoperable AAI. A liaison with the AARC project has been established to adopt AAI policies, solutions and best practices defined at European level and deal with problems that require a larger scope to be resolved, such as the lack of globally unique identifiers, levels of assurance etc.
The outcome of this process has been a list of core requirements that drove the design of new EGI AAI:

- Obtain access EGI services with credential released by his/her home organisation enabling the support for National Federation via eduGAIN.
- Support 'homeless users', who cannot rely on a reliable institutional IdP.
- Release by Identity Providers (IdPs) of an identifier that uniquely identifies the user in the scope of that organization.
- Ability to extract attributes from different sources including community attribute providers.
- Ability to associate a Level of Assurance (LoA) to each identity in the EGI infrastructure.
- Availability of a persistent non-reassignable unique identifier for users in order to manage the accounting linking.
- Provisioning of credential translator mechanisms/token translator services (TTSs) is needed to hide the complexity of the new EGI AAI to the service providers.

By the end of the first year of JRA1.1, new components of the EGI AAI are fully functional in terms of core features and EGI started an on-boarding activity for scientific communities. The recent introduction of the pilot CILogon service, enables all users to access even the legacy non-web EGI Services through the EGI AAI.
By the end of Q2 2016 it is expected that the EGI AAI will join eduGAIN as Service Provider supporting the GÉANT Data Protection Code of Conduct and the REFEDS Research and Scholarship entity category. Through eduGAIN, EGI Services will automatically become available to more than 2000 Universities and Institutes that are connected to the 38 eduGAIN Federations. Complementary to this, users without an account on a federated Identity Provider will be able to use their Google, Facebook, LinkedIn and ORCID accounts to access EGI Services that do not require substantial level of assurance.
In parallel, we are working on the first phase of the pilot with the EGI Competence Centres in order to connect Research Infrastructures to the EGI AAI. This is an interactive process, which allows us to shape the EGI AAI exactly to the needs of our customer base.
In the third quarter of this year, we will continue with the second phase of the pilot, by the end of which we expect to have all the EGI scientific communities enabled by the services of

the EGI AAI. In addition, we will be introducing the new OpenID Connect interface, which will enable us to introduce new services in the EGI platforms in a faster and friendlier way.

1 Introduction

Tool name	EGI AAI CheckIn Service
Tool url	https://aai.egi.eu/proxy
Tool wiki page	https://wiki.egi.eu/wiki/AAI
Description	Provides Authentication and Authorisation, enabling user-friendly and secure federated access to EGI services.
Value proposition	The EGI AAI CheckIn Service enables research communities to access the EGI services without having to deal with X509v3 certificates. Researchers from home organizations that participate in one of the eduGAIN federations will be able to access the EGI services using the same credentials they are using at their home organization. Furthermore, the EGI AAI CheckIn Service supports user authentication with social media identities, enabling even those users who do not have a federated account at a home organization (such as many users that belong to the "Long Tail of Science"), to be able to access the EGI services in a seamless way without compromising the security of the EGI technical platforms. The EGI AAI CheckIn service can connect to existing community based AAIs and it can be offered as an "Identity Access Management as a Service" to those communities, which do not have or do not want to operate their own AAIs.
Customer of the tool	NGI; RI; Resource Providers
User of the service	All EGI users
User Documentation	https://wiki.egi.eu/wiki/AAI#Documentation
Technical Documentation	https://wiki.egi.eu/wiki/AAI#Documentation
Product team	GRNET
License	Apache 2.0 License
Source code	https://github.com/rciam

2 Service architecture

The AAI activity in EGI-Engage started in May 2015. During the first months of the project, we worked together with the AARC project in order to identify the requirements of the scientific communities. This work resulted in a set of guiding principles:

- Users should be able to access the EGI Services using the credentials they have got from their Home Organizations using eduGAIN when possible, but alternative methods should be available
- EGI should expect to receive at least an identifier that uniquely identifies the user coming from within the scope of the authentication source.
- Within the EGI environment, a user should have one persistent non-reassignable non-targeted unique identifier.
- EGI should define a set of minimum mandatory attributes, without which a user cannot exist within the EGI environment.
- EGI should attempt to retrieve these attributes from the user's Home Organization. If this is not possible, then an alternative process should exist in order to acquire and verify the missing user attributes.
- There should be a distinction (LoA) between self-asserted attributes and the attributes provided by the Home Organization/VO
- Access to the various services should be granted based on the VO/EGI roles the user has.
- EGI Services should not have to deal with the complexity of multiple IdPs/Federations/Attribute Authorities/technologies. This complexity should be handled centrally and should be hidden from the EGI Services.

2.1 High-Level Service architecture

Based on these principles and following the guidelines from the AARC project, we designed an architecture for the EGI AAI and a roadmap in order to incrementally introduce the new service elements in the EGI AAI platform.

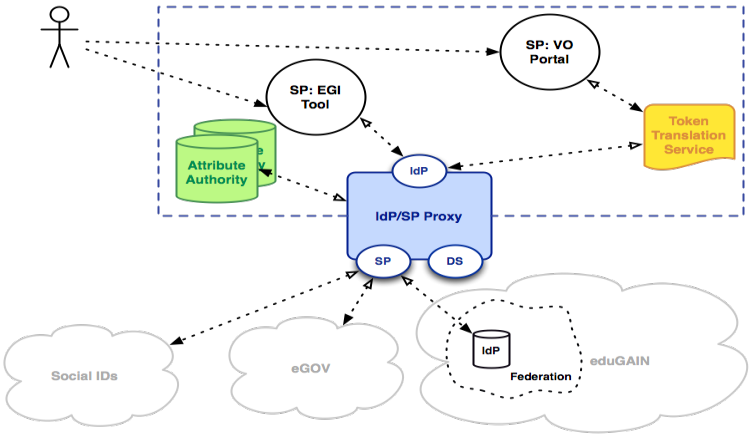


Figure 1: EGI AAI CheckIn Service high-level architecture (SP: Service Provider; DS: Delegation Service; VO: Virtual Organization)

The core of EGI AAI CheckIn Service is the IdP/Service Provider (SP) Proxy component, which acts as a bridge between the EGI services and external authentication sources and

identity providers. This decoupling of the internal services and the external authentication sources/identity providers, reduces the complexity of the service implementation as it removes dependencies on the heterogeneity of multiple IdPs, Federations, Attribute Authorities and different authentication and authorization technologies. This complexity is handled centrally by the proxy.

The introduction of a IdP/SP Proxy entity introduces additional benefits as detailed below. As illustrated in Figure 1, services only need to establish trust with one entity, the IdP/SP proxy. Typically, services will have one static configuration for the IdP/SP proxy.

Having one configured IdP also removes the requirement from the service providers to operate their own IdP Discovery Service (a common requirement for services supporting federated access). Furthermore, all internal services will get consistent and harmonized user identifiers and attributes, regardless of the home organization or the research community the user belongs to. Finally, this separation simplifies change management processes, as the internal services are independent from the IdPs run by the home organizations. Similarly, IdPs establish trust with one entity, the operator of the IdP/SP proxy, and they are not impacted by the operational changes introduced by each individual service.

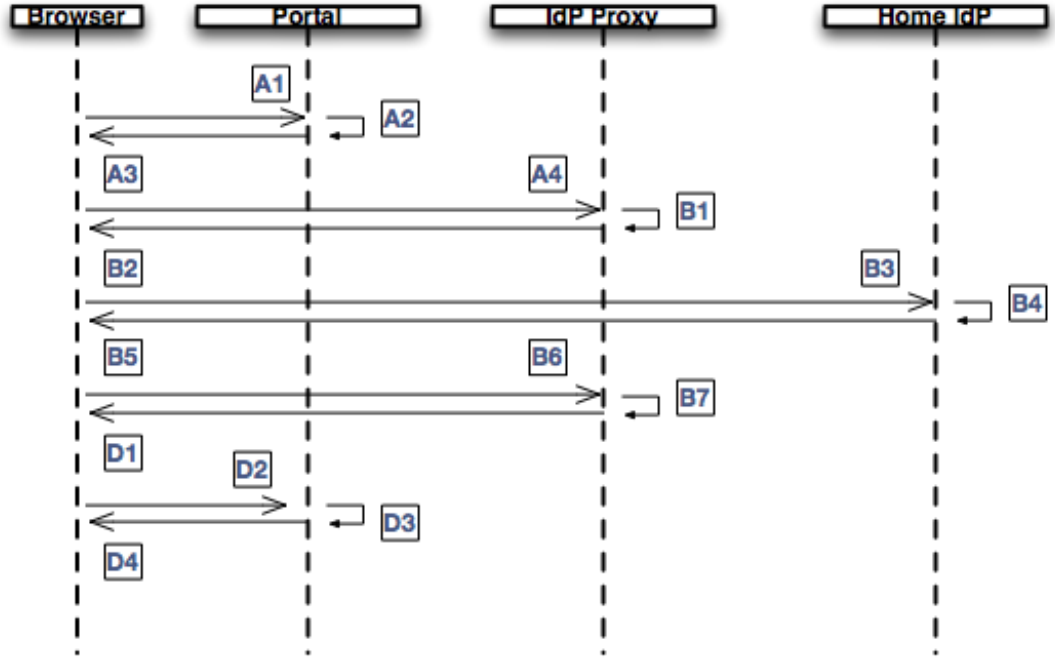
The following section details the workflow linking the entities illustrated in Figure 1.

2.2 Integration and dependencies

2.2.1 Integration with SAML Identity/Service Providers

This is the foundation of EGI AAI integration use case aiming at enabling users to access a web-based EGI services (Service Provider - SP) using their credentials and attributes from their home organisation (Identity Provider - IdP). The most common way for enabling federated access on web-based EGI services is implemented through the use of SAML. The typical Single Sign-On (SSO) flow begins with the user accessing an EGI application through their web browser (SP-initiated SSO), as depicted in the following diagram.

The workflows involving the user, the Service Provider and the IdP Proxy service are detailed below.



EGI SP flow:

A1. The user visits the web-based SP ("portal") using a personal browser
A2. The user selects to login using her/his federated account
A3. The portal generates a SAML request and redirects the user's browser to the SAML endpoint of the EGI IdP Proxy, embedding the SAML request in the URL
A4. The user's browser redirects to the EGI IdP Proxy and passes along the SAML request as a URL parameter

EGI AAI IdP Proxy flow:

B1. The EGI AAI IdP Proxy verifies the SAML request, presents a set of Home Organisations (IdP Discovery Service) to the user and the user selects a Home Organization of choice
B2. The EGI AAI IdP Proxy generates a new SAML request and redirects the user's browser to the SAML endpoint of the Home IdP of the user
B3. The user's browser redirects to the Home IdP and passes along the SAML request as a URL parameter
B4. The user authenticates herself/himself at the Home IdP and upon successful authentication, the Home IdP builds a SAML assertion representing the user's logon security context, it is digitally signed and placed within a SAML message. The message is then placed within an HTML FORM as a hidden form control named SAML response.

B5. The Home IdP sends the HTML form back to the browser in the HTTP response. For ease of use purposes, the HTML FORM is accompanied by script code that will automatically post the form to the destination site
B6. The browser, due either to a user action or execution of an "auto-submit" script, issues an HTTP POST request to send the form to the EGI AAI IdP Proxy.
B7. The EGI AAI IdP Proxy decrypts and verifies the SAML Assertion.
    a. The SAML Assertion must include at least one of the following: eduPersonUniqueId (ePUID), eduPersonPrincipalName (ePPN), or eduPersonTargetedID (ePTID).
    b. The EGI AAI IdP Proxy replaces the original user identifier with an ePUID which is generated by hashing the [ePUID|ePPN|ePTID] and is scoped at "egi.eu".
    c. The EGI AAI IdP Proxy builds a new SAML assertion representing the user's logon security context, it is digitally signed and placed within a SAML message. The message is then placed within an HTML FORM as a hidden form control named SAML response.

SP "Termination" flow:

D1. The EGI AAI IdP Proxy asks for user consent and when it is given, it sends the HTML form back to the browser in the HTTP response. For ease of use purposes, the HTML FORM is accompanied by script code that will automatically post the form to the destination site.
D2. The browser, due either to a user action or execution of an "auto-submit" script, issues an HTTP POST request to send the form to the SP.
D3. The SP decrypts and verifies the SAML assertion and makes an access check to establish whether the user has the correct authorisation to access the resource.
D4. If the access check passes, the resource is then returned to the browser.

In the context of the IdP Proxy pilot activities and based on the flows described above, the following SAML IdP/SP entities have been tested and are currently interconnected following the best practices and guidelines for interconnecting IdPs[1] and SPs[2] defined and tested during the pilot:

[1] https://wiki.egi.eu/wiki/AAI_guide_for_IdPs

[2] https://wiki.egi.eu/wiki/AAI_guide_for_SPs

[2] https://wiki.egi.eu/wiki/AAI_guide_for_SPs

Name of entity	Role in the pilot	AAI Technology involved in the pilot
EGI AAI IdP/SP Proxy	IdP/SP Proxy	SimpleSAMLphp[3]
GRNET VHO	IdP	Shibboleth[4]
EGI SSO	IdP	Shibboleth
GOCDB	SP	Shibboleth
AppDB	SP	Shibboleth
ELIXIR AAI	IdP Proxy	OpenConext[5]

2.2.2 Integration with Social Identity Providers

This is an extension to the above integration use case allowing users to authenticate against commonly used Social Identity Providers. As these providers do not provide SAML-compliant authentication mechanisms, the EGI AAI CheckIn Service needs to act as an OpenID Connect/OAuth2-to-SAML bridge. In this context, it is required to map social identity profiles into SAML attribute assertions. These mappings are described hereafter. It should be noted that only user information, which is relevant to the REFEDS Research and Scholarship (R&S[6]) attribute bundle, is covered in this section.

2.2.2.1 Google / OpenID Connect

Google's OAuth 2.0 APIs can be used for both authentication and authorisation. This OAuth 2.0 implementation conforms to the OpenID Connect (OIDC) specification and is OpenID Certified. As such, when including the OpenID scope, information about the user can be retrieved from the UserInfo endpoint in OpenID Connect format. The Claims[7] returned in the UserInfo Response can be mapped to SAML attributes as follows:

[3] https://simplesamlphp.org/

[4] https://shibboleth.net/

[5] https://openconext.org/

[6] https://refeds.org/category/research-and-scholarship

[7] https://openid.net/specs/openid-connect-basic-1_0.html#StandardClaims

Google (OIDC) user claim	EGI Profile SAML attribute
sub	ePUID (scoped @google.com) → hashed ePUID (scoped @egi.eu)
name	displayName
given_name	givenName
family_name	sn
email	mail

2.2.2.2 *Facebook*

Facebook allows retrieving user information through the /{user-id} Graph API endpoint[8], following an OAuth 2.0 flow for authentication and authorisation. The returned fields of the Facebook user profile can be mapped to SAML attributes as follows:

Facebook user field	EGI Profile SAML attribute
third_party_id (an anonymous, but unique identifier for the person that can be shared with third parties)	ePPN (scoped @facebook.com) → hashed ePUID (scoped @egi.eu)
id (this ID is unique to each app and cannot be used across different apps)	ePTID ("http://facebook.com!" + id) (omitted from the EGI identity attribute profile since the included ePUID is generated from the ePPN above)
name	displayName
first_name	givenName
last_name	sn
email	mail

2.2.2.3 *LinkedIn*

LinkedIn relies on the OAuth 2.0 protocol for enabling authenticated access to its REST APIs that provide access to member data. More specifically, following a three-legged OAuth2

[8] https://developers.facebook.com/docs/graph-api/reference/v2.6/user

flow, LinkedIn user profile[9] information can be accessed through the `/people/~` REST API endpoint. The returned user fields can be mapped to SAML attributes as follows:

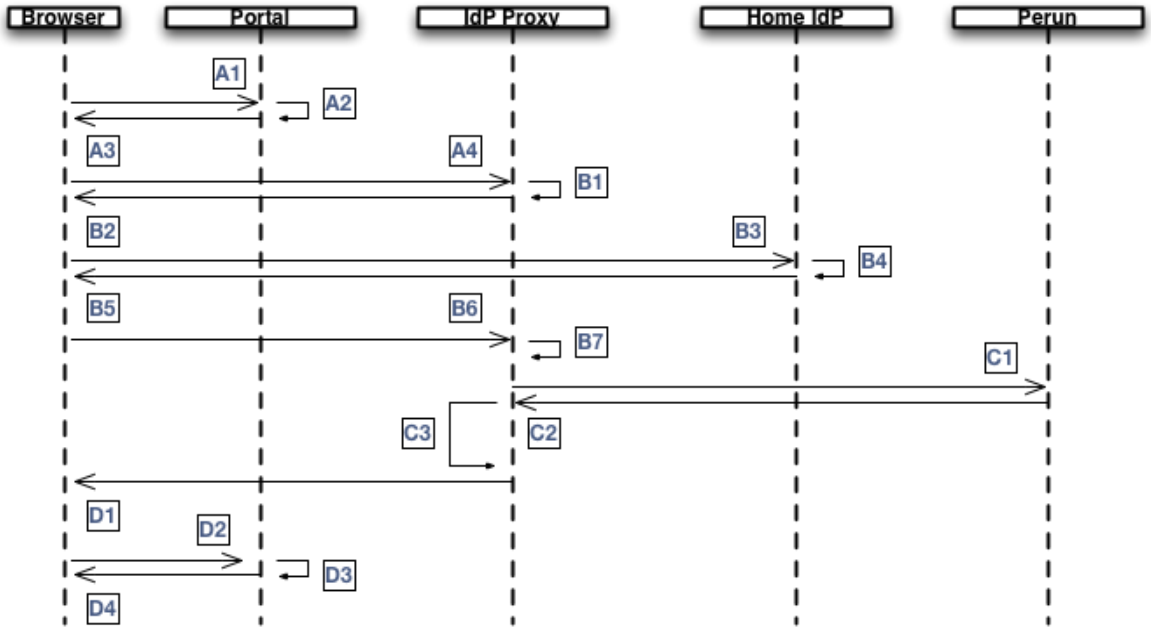
LinkedIn user field	EGI Profile SAML attribute
id (a unique identifying value for the user, which is linked to the specific application)	ePTID ("http://linkedin.com!" + id) → hashed ePUID (scoped @egi.eu)
formatted-name	displayName
first-name	givenName
last-name	sn
email-address	mail

2.2.3 Integration with Perun Attribute Authority

The EGI AAI proxy has been integrated with the Perun[10] technology in order to retrieve information describing the user's VO/group memberships. This information is encapsulated in URN-formatted eduPersonEntitlement values, which are incorporated into the original SAML attribute assertion sent by the user's IdP before being passed on to the relying party. The interactions among the involved components have been visualised in the following diagram.

[9] https://developer.linkedin.com/docs/fields/basic-profile

[10] https://perun.cesnet.cz/web/index.shtml

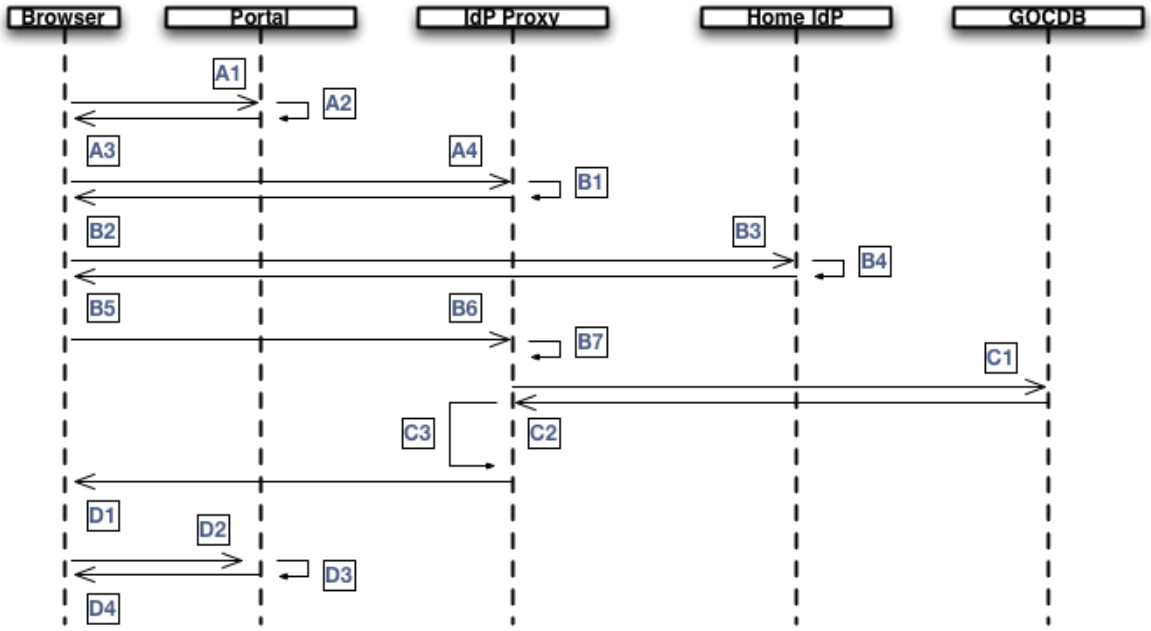


Perun flow:

C1. The EGI AAI IdP Proxy makes a back channel LDAP query to the Perun service passing along the user identifier
C2. Perun returns the group membership information in the LDAP query response
C3. The EGI AAI IdP Proxy encapsulates the returned information in an eduPersonEntitlement attribute which is added to the SAML Attribute Response

2.2.4 Integration with GOCDB Attribute Authority

The EGI AAI proxy has been integrated with the GOCDB in order to retrieve information describing the user's role(s) when operating EGI Resource Centres (also named "sites"). This information is encapsulated in URN-formatted eduPersonEntitlement values, which are incorporated into the original SAML attribute assertion sent by the user's IdP before being passed on to the relying party. The interactions among the involved components have been visualised in the following diagram.



C1. The EGI IdP Proxy makes a back channel REST API call to GOCDB[11] passing along the user's identifier contained in the SAML authentication response received by the IdP.
C2. GOCDB returns the user roles in the HTTP response
C3. The EGI IdP Proxy adds the received attributes to the SAML Attribute Response

2.2.5 Integration with CILogon-like Token Translation Service – End-Entity Certificates

The EGI AAI Proxy has been integrated with additional components to ensure the mapping of arbitrary user credentials to X.509 certificates[12], in order to ensure easy access to services that require identity and attribute certificates for federated access. A Token Translation Service (TTS) has been integrated, based on CILogon[13], an open source project. TTS supports OAuth[14] via the MyProxy product OA4MP[15]. More specifically, the CILogon-like TSS service comprises the following components:

[11] https://wiki.egi.eu/wiki/GOCDB/PI/get_user_method

[12] https://tools.ietf.org/html/rfc5280

[13] http://www.cilogon.org/

[14] http://oauth.net/

[15] http://grid.ncsa.illinois.edu/myproxy/oauth/

- Shibboleth: Service Provider 2.0.
- MyProxy Server[16], for credential management and OA4MP, the MypProxy client software for OAuth 2.0 support.
- SimpleCA[17], a simple implementation by Globus.org of a certification authority that can issue X.509 certificates.

The components of the CILogon-like TTS service and their interconnections with the EGI AAI are illustrated in the following diagram[18].

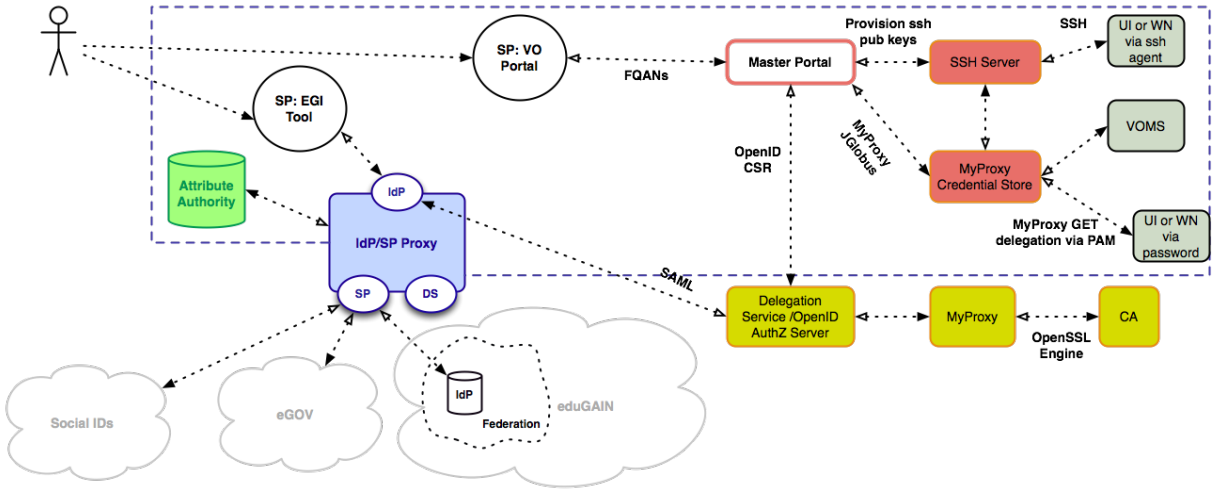


Figure 2. Extended technical architecture of the EGI AAI Checkin service in order to include TTS capabilities

User flow:

E1. User goes to VO portal
E2. browser redirect to `/authorize` endpoint on Master Portal of TTS (see picture above)
E3. browser redirect to `/authorize` endpoint on Delegation Service (DS)
E4. browser redirect (SAML) to EGI AAI IdP Proxy
E5. user authenticates at their Home IdP through the EGI IdP Proxy (via WAYF)
E6. redirect back (SAML) to `/authorize` on DS
E7. redirect back to 'redirect_uri' on Master Portal

[16] http://grid.ncsa.illinois.edu/myproxy/ca/

[17] http://toolkit.globus.org/toolkit/docs/latest-stable/simpleca/

[18] A more detailed view of the CILogon-like TTS service architecture is available at https://wiki.nikhef.nl/grid/CILogon_Pre-Pilot_Work#Detailed_Architecture

(Master Portal retrieves access_token-2 and uses it to obtain userinfo)

E8. redirect back to 'redirect_uri' on VO portal

The next steps are all hidden from the user:

E9. VO portal retrieves access_token-1 from Master Portal
E10. VO portal uses access_token-1 to obtain userinfo from Master Portal
E11. VO portal calls `/getproxy` endpoint (on Master Portal) using access_token-1, optionally with VOMS information (VO FQANs and/or VOMSES)
E12. Master Portal checks presence of long lived proxy in MyProxy credential store, using MyProxy INFO command. In case there is no proxy yet:
   a. Master Portal creates keypair + Certificate Signing Request (CSR)
   b. Master Portal calls `/getcert` on DS using access_token-2 and CSR
   c. Delegation Service (DS) issues a MyProxy GET request at online CA, using the CSR
   d. online CA signs the CSR and returns the end-entity certificate to DS
   e. DS returns certificate to Master Portal
   f. Master Portal uses end-entity certificate and key to delegate (MyProxy PUT) a long-lived proxy to the MyProxy credentials store
E13. Master Portal retrieves short lived (VOMS) proxy from the MyProxy credstore
E14. Master Portal returns proxy to VO portal

2.2.6 Integration with CILogon-like Token Translation Service – Per-User Sub-Proxies (PUSP)

The steps here are almost identical to those presented in Section 2.2.5, except a (Per-User Sub) proxy is returned from the online CA instead of an end-entity certificate, which can then be stored directly. Details on the MyProxy reconfiguration can be found at https://wiki.nikhef.nl/grid/PUSP_from_MyProxy

User flows: E1-E8 (refer to Section 2.2.5)

The next steps are all hidden from the user:

E9. VO portal retrieves access_token-1 from Master Portal
E10. VO portal uses access_token-1 to obtain userinfo from Master Portal
E11. VO portal calls `/getproxy` endpoint (on Master Portal) using access_token-1
E12. Master Portal checks presence of long lived proxy in MyProxy credential store, using MyProxy INFO command. In case there is no proxy yet:
   a. Master Portal creates keypair + CSR
   b. Master Portal calls /getcert on DS using access_token-2 and CSR
   c. DS does a MyProxy GET request at 'online CA', using the CSR
   d. online CA signs the CSR using the robot private key and returns a PUSP chain to DS

e. DS returns proxy chain to Master Portal
f. Master Portal uses proxy chain with the previously generated private key to store (MyProxy STORE) the long-lived proxy to the MyProxy credstore

E13. Master Portal retrieves short lived proxy from the MyProxy credstore
E14. Master Portal returns proxy to VO portal

2.2.7 Integration with COmanage Registry – User Enrollment

The EGI AAI Proxy has also been integrated with the COmanage Registry[19] to provide a seamless onboarding experience for new EGI users and to support advanced account management features, such as the ability to link user accounts from different identity providers into a single EGI profile. This section puts the focus on the user enrollment workflow, which has already been finalised, whereas the account linking process is still under development.

The COmanage Registry comprises a database for maintaining account information, as well as a Web UI and a rich set of APIs enabling federated access (SAML) to the user attributes managed by the COmanage Registry. To allow for the automatic registration of new users upon accessing an EGI service for the first time, the COmanage registry has been connected with the EGI AAI Proxy as both an Attribute Authority (AA) and a Service Provider (SP). Effectively, the flow is an extension of the baseline scenario presented in Section 2.2.1. More specifically, the final step (Step B7) becomes as follows:

B7. The EGI AAI IdP Proxy decrypts and verifies the SAML Assertion.
   a. The SAML Assertion must include at least one of the following: eduPersonUniqueId (ePUID), eduPersonPrincipalName (ePPN), or eduPersonTargetedID (ePTID).
   b. The EGI AAI IdP Proxy replaces the original user identifier with an ePUID which is generated by hashing the [ePUID|ePPN|ePTID] and is scoped at `"egi.eu"`.
   c. The EGI AAI IdP Proxy queries the COmanage Registry (AA) using the ePUID from step B7b.
   If the user is found in the COmanage Registry:
      i. The EGI AAI IdP Proxy builds a new SAML assertion representing the user's logon security context, it is digitally signed and placed within a SAML message. The message is then placed within an HTML FORM as a hidden form control named SAMLResponse.

   Alternatively, if the user is not found in the COmanage Registry:
      i. The user's browser is redirected to the new user registration page in COmanage (SP)

[19] https://spaces.internet2.edu/display/COmanage/

ii. The registration page gets populated with the user's generated EGI unique identifier and all attributes asserted by the Home IdP
iii. User fills in any mandatory attributes that may be missing, such as name or email address information.
iv. User explicitly consents to the Terms of Use of the EGI AAI SP Proxy and then chooses to submit the user registration form.
v. COmanage sends confirmation link to the email address associated with the user's EGI account
vi. User opens the link in the email sent by COmanage and is navigated to the registration page to confirm ownership of the email address
vii. User confirms email address through the COmanage registration page
viii. User re-authenticates at their Home IdP (through the EGI AAI IdP Proxy) to finalise the new EGI user enrolment process.

3 Release notes

3.1 Requirements covered in the release

The EGI AAI CheckIn Service is still under active development and the first "production release" is expected by the fall of 2016. As this is a high impact service, the existing EGI Pilot is already operated in terms of a production service, with the difference that new features/changes are pushed into production on a daily basis. With the first production release, the service will be subject to standard change management processes already applied to other EGI services.

Information about the requirements covered and the features that have been implemented or which are in the plan until the first production release can be found in the EGI AAI CheckIn Service development roadmap[20].

4 Customer satisfaction and future plans

Currently the EGI AAI CheckIn Service is in pilot phase. We have started the integration of EGI Services and Tools and the on-boarding of the Research Communities from the EGI Competence Centres. In addition to this, EGI federation services like GOCDB, AppDB and GGUS have already been integrated and we have started integration activities for the Cloud Federation.

The choice of the services added to the integration plan, is driven by priorities and needs of the Research Communities that are being on-boarded. We have started with the research

[20] https://wiki.egi.eu/wiki/EGI-Engage:TASK_JRA1.1_Authentication_and_Authorisation_Infrastructure#Development_Roadmap

communities from the ELIXIR Competence Centre and we will continue with the research communities from the other Competence Centres.
During QR4 2016, a service assessment will be performed in collaboration with user communities and service providers.

The outcome of this will be considered to finally select the products needed for the technical implementation of the service and its future evolution.