MS31: Integration and Federation guidelines

Lead Partner:	INFN
Version:	**1.0**
Status:	**draft**
Dissemination Level:	**PU**
Document Link:	https://repository.eosc-pillar.eu/index.php/f/29611

Document Abstract
This is the accompanying document of MS31, which contains the initial guidelines for the technical integration/federation of resources and services with the EOSC.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 857650.

COPYRIGHT NOTICE

DELIVERY SLIP

Date	Name	Partner/Activity	Date
From:			
Moderated by:			
Reviewed by:			
Approved by:			

DOCUMENT LOG

Issue	Date	Comment	Author
v0.1	26.3.2020	Table of Content and introduction	L. Gaido
v0.2	20.4.2020	Added sections 2.2. and 2.4	L. Gaido
v0.3	15.5.2020	Added sections 2.1, 3.1, Executive summary, conclusions and Future work	B. Chiavarini, A. Ceccanti, D. Salomoni, L.Gaido
v0.4	20.5.2020	First draft released	L. Gaido
v1.0		Final version after feedback from the Technical Board	L. Gaido

TERMINOLOGY

https://eosc-portal.eu/glossary

Terminology/Acronym	Definition
INDIGO-DataCloud	H2020 development project (2015-2017): www.indigo-datacloud.eu
EUDAT CDI	EUDAT Collaborative Data Infrastructure: https://eudat.eu/eudat-cdi

Contents

1 Executive summary

This document describes the guidelines for the integration of EOSC-Pillar services with the so-called EOSC federating core services.

Although various discussions are currently ongoing also at the EOSC Governance Working Groups1 level about what the core services are, we will rely on the definition of the EOSC-hub project for the time being.

These guidelines for the integration have been defined according to the needs of the EOSC-Pillar project, specifically of the use cases and services identified in Work Package 6 (EOSC in action: Use-cases and Community driven pilots) and task T7.4 (Services ready to use).

Also for the concept of federation of services, there are different perceptions of what this means. For the scope of this document, federation means connecting the services with the EOSC, i.e. making them available through the EOSC catalogue and Marketplace.

Of course the guidelines will evolve in order to address additional need which may emerge during the project lifetime and specifically considering the needs of the national infrastructures and services within the countries participating to the project.

The final version will be described in deliverable D7.1 (Guidelines and Recommendations for the Technical Integration of Resources and Services with the EOSC), which is due at month 24 (June 2021).

[1] https://www.eoscsecretariat.eu/eosc-working-groups

2 Introduction

The EOSC-Pillar project involves partners from 5 different European countries (Austria, Belgium, France, Germany and Italy) with quite different situations regarding national e-infrastructures and services as well as participation to international initiatives and projects.

Within the EOSC-Pillar project, several relevant use cases have been identified during the project preparation and their activities are part of the project within Work Package 6, with a specific task (T6.1 to T6.9) for each of them.

Additional use cases may be identified during the course of the project, through an open call to communities (task T6.10).

The starting point for the T7.1 activity related to guidelines for the services integration is the inventory of the services needed by the WP6 use cases. This is responsibility of task T7.4.

According to the results of this assessment, specific guidelines will be defined to help the partners involved in task T7.2 to support the integration of the EOSC-Pillar services with the EOSC federating core services whenever appropriate.

In addition it is also important to understand which services are of interest of the countries involved in the project in order to provide support for their integration, too. This in principle may require a huge man power effort, depending on the number of services identified, maybe greater than the one available in task T7.2. Therefore a prioritization of the support activities may be needed.

The results of this activity are reported in this document.

2.1 Analysis of the WP6 requirements

The D6.1 deliverable[2] document, where requirements for WP6 use cases are collected, was taken as the basis to identify which services to be integrated are needed by the single use cases.

From the use case analysis templates were taken into account the following fields:

- Need of Use Cases in terms of compute power
- Auxiliary applications
- Gap Analysis/Missing pieces

The following fields from the single use case templates were considered to collect further information:

- Infrastructure needs
- Capacity needs

[2] https://repository.eosc-pillar.eu/index.php/apps/files/?dir=/EOSC-Pillar%20Shared%20Files/Workpackages/WP6/Deliverables&fileid=25364

- What are the services currently missing?

Some answers do not refer to services but only to generic needs that are not already carried out by the current services involved in the use cases, while others point to thematic services or already identified particular services (e. g. INDIGO-DataCloud components for UC6.6).

Other required services are those related to CPU or GPU resources, virtual machines or storage space.

None of the so-called EOSC core services is mentioned.

Explicit references about AAI services are only in UC6.4, UC6.6 and UC6.8. UC6.4 and UC6.8 already have an AAI system while UC6.6 will be integrated with INDIGO-IAM.

For a further analysis of WP6 use cases requirements on AAI services, the results of a specific survey about this theme managed by task T7.2 were examined. According to the responses, all the communities or research infrastructures involved in the WP6 use cases already use AAI solutions for their use cases. None of them think to lack information about federated identity management and neither have expressed any need for material to inform their organization about federated access nor for a support relating to AAI services. No responses were gathered from UC6.1, UC6.5 and UC6.9.

2.2 Results of the T7.4 services inventory

In order to understand among the services of interest of task T7.4, which ones need support for their integration, a survey has been conducted among the different user communities behind the T7.4 use cases.

They are:

- Pico2
- DataTerra
- VIP
- BDSS, Materials Modelling Marketplace and VIMMP services
- VRE
- Laniakea

With respect to AAI, the result of the survey shows that two services (Laniakea and VRE) are already connected with an AAI service which supports federated identity providers (EDUGAIN) and as such they don’t need support for integration with INDIGO-IAM, the reference AAI service selected for the EOSC-Pillar project.

Two other services (Pico2 and DataTerra) currently are in a prototype phase and will need support for the integration with INDIGO-IAM at a later stage, as soon as they will be consolidated.

Also VIP is in a prototype phase and they will have to discuss internally which AAI service can be considered for integration, either the EGI service (Check-in) or INDIGO-IAM.

Concerning “BDSS, Materials Modelling Marketplace and VIMMP services”, this is a quite complex situation because these services essentially consist in Marketplace/inventory services developed in the frame of an EU-funded project for a specific community, namely the Materials Modelling community. The aim of this community is to make these services available to a broader range of users, also through the connection with the corresponding EOSC services. A common AAI service is paramount for this, but no decision has been taken yet with respect to which AAI service they may be willing to adopt/integrate with. This will be clarified after a thorough evaluation of the technical ways to make their services interoperable with the EOSC ones.

None of the respondents expressed the need to integrate other services.

3 Guidelines

In order to avoid reinventing the wheel, information has been collected about the guidelines already available at European level within the projects which foresee activities to support the service integration as well as at national level.

According to the results of the assessment done within WP6 and T7.4, the only service identified which requires support for integration is Authentication and Authorization system (AAI).

The INDIGO-IAM has been selected as the baseline AAI service for the EOSC-Pillar project and the guidelines for its integration are described in the section 3.1.

Information and guidelines for the integration with other services which may be needed in the near future are reported in section 3.2

3.1 Guidelines for the integration of the INDIGO-IAM service

INDIGO-IAM (https://github.com/indigo-iam/iam) is an Identity and Access Management Service providing a layer where identities, enrolment, group membership and other attributes and authorization policies on distributed resources and applications can be managed in a homogeneous way. INDIGO-IAM supports identity federations and other authentication mechanisms, such as X.509 certificates and social logins.

The IAM service has been successfully integrated with many off-the-shelf components like OpenStack, Kubernetes, Atlassian JIRA and Confluence, Grafana and with several key Grid computing middleware services such as FTS, dCache, StoRM. IAM can also be easily integrated with applications thanks to its adoption of industry standards such as the OpenID-Connect protocol, the OAuth protocol and JSON Web Tokens, or JWT.

The adoption of these technologies, which are standard and widely used in the industry, allows the reuse of existing knowledge and tools, reduces integration complexity through off-the-shelf libraries and components, provides an authentication-agnostic mechanism, and scales thanks to the distributed verification of access and identity tokens.

This paragraph shows a sample integration example and provides pointers to the general integration of INDIGO-IAM into applications.

In summary, some the key features that make it worthwhile and easy to integrate an application into INDIGO-IAM are the following:

- Simplicity, thanks to the use of open standards.
- Support of an enrolment and registration service, allowing users to submit membership requests to a given organization.
- Linking of various credentials or authentication methods to a single identity.
- Flexibility, possibly granting administrator privileges to users, adding users to groups, managing membership requests, editing user registration information.

- Expose or manage user and group information, so that it could be consumed by applications through programmable interfaces.
- Support of full auditing to keep track of all interesting security events.

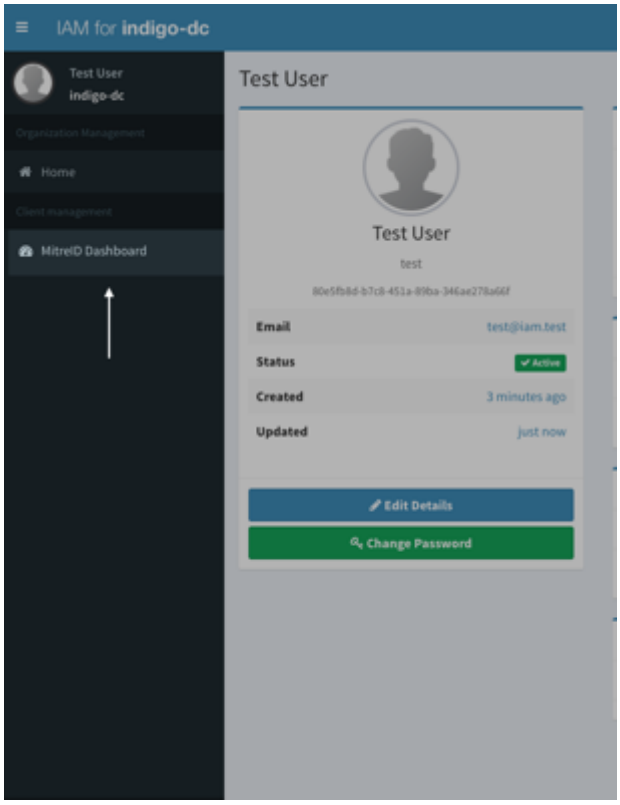
In order to show a sample integration of an application, we will show how a standard web server could be modified, so that access to that server goes through INDIGO-IAM first. This is a general picture of this scenario:

INDIGO-IAM natively uses OpenID-Connect to talk to applications. The application in our use case is a web server; we'll use the popular Apache server, so we need to enable OpenID-Connect authentication in Apache. We will use containers to make the entire process easily reproducible.

The steps that need to be followed are:

- Register a "new application" in INDIGO-IAM. We will need to tell INDIGO-IAM the public IP address and port of the web server.
- Create a sample index.html file for a sample web home page.
- Create the configuration file for the web server.
- Create a Docker file which will build the container, installing all necessary packages and copying the html file and the configuration.
- Run the container.

In order to register an application with INDIGO-IAM, one needs to log in to an INDIGO-IAM instance with administrator privileges and click on MitreID dashboard on the navigation bar:



From the MitreID dashboard, select Self service client registration and Register a new client.

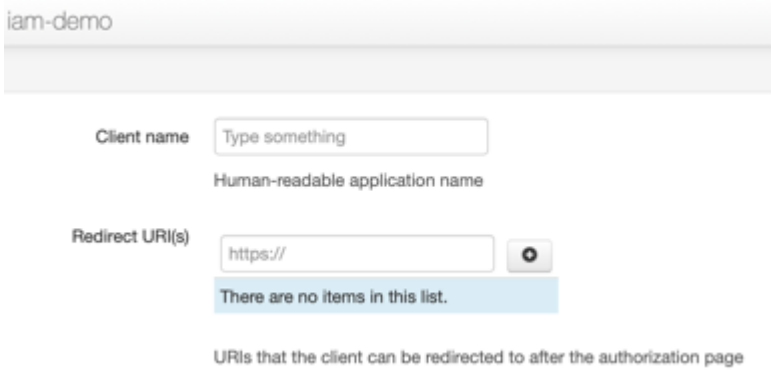
We now need to give INDIGO-IAM information about the application to be integrated. In the Client name field, write a descriptive text, e.g. “Test Web Server”.

In the Redirect URI(s), one should put this:

http://<public_ip_address>:<port>/redirect_uri

Where <public_ip_address> is the IP public address of the web server, and <port> is the port under which the web server runs. If https is used, replace http with https.

The other fields can be left to their default values.



If one now clicks on **Save** to register the application, INDIGO-IAM will generate the client credentials. These are **needed** for the web server configuration; in particular, the **Client ID** and the **Client Secret** must be noted down.

For the sake of this example, we will create a very simple home page in html format for the web server. Create the following index.html file:

```
<html>
<h1>Welcome to an OpenID-Connect protected page!</h1>
This is a sample web server home page.
</html>
```

We now need to configure Apache, and this is where the integration with the INDIGO-IAM instance happens. Create a file called default.conf with the following lines:

```
<VirtualHost *:80>
 ServerAdmin webmaster@localhost
 DocumentRoot /var/www/html
 OIDCProviderMetadataURL https://<indigo-iam_ip_address>/.well-
known/openid-configuration
 OIDCClientID <put the Client ID here>
 OIDCClientSecret <put the Client Secret here>
 OIDCRedirectURI http://<public_ip_address>:<port>/redirect_uri
 OIDCCryptoPassphrase <put a random password here>
 <Location />
   AuthType openid-connect
   Require valid-user
 </Location>
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Note that in the configuration file above **<indigo-iam_ip_address>** must be the IP address of the INDIGO-IAM instance where we are integrating our web server.

We will now create a Dockerfile to generate a Docker image containing our Apache web server, properly configured to connect to INDIGO-IAM and with our very simple home page:

```
FROM ubuntu
RUN apt update
RUN apt install -y apache2 libapache2-mod-auth-openidc
COPY index.html /var/www/html/
COPY default.conf /etc/apache2/sites-enabled/000-default.conf
EXPOSE 80
CMD ["apachectl", "-D", "FOREGROUND"]
```

At this point, we can generate the image, calling it e.g. web_server_oidc:

```
docker build -t web_server_oidc .
```

We can now run the image, instantiating a new web server connected to INDIGO-IAM:

```
docker run -d -p 80:80 web_server_oidc
```

Opening a web browser to the page http://<public_ip_address>:<port>/ should now redirect to the INDIGO-IAM service.

One could now log in to INDIGO-IAM with Google credentials, with a IAM local user name or with other federated identities (if configured in the INDIGO-IAM service) and, if the authentication is successful, be automatically redirected to the home page on the web server, thus implementing the full flow described in the use case architectural picture shown above.

For more information:

- Full tutorial on integrating Apache into INDIGO-IAM: https://github.com/andreaceccanti/iam-tutorial/tree/master/apache-integration-demo.
- General information about integration possibilities with INDIGO-IAM: https://indigo-iam.github.io/docs/v/current/.
- A complete webinar on INDIGO-IAM: https://indico.in2p3.fr/event/21072/.

3.2 Guidelines for the integration of other services

One of the most relevant EOSC implementation projects is EOSC-hub, which relies on the outcome of the two main e-infrastructures at European level (EGI and EUDAT) and of the main software development project, INDIGO-DataCloud. It started its activities in January 2018 and will last until December 2020.

Among other things, this projects is doing several activities to support integration of services, specifically with the so-called core services and this activity is supported by several specific documents made available through a dedicated wiki page[3]

These services are:

- ARGO Messaging Service (AMS)

The ARGO Messaging Service is a Publish/Subscribe Service, which implements the Google PubSub protocol. Instead of focusing on a single Messaging API specification for handling the logic of publishing/subscribing to the broker network the API focuses on creating nodes of Publishers and Subscribers as a Service. It provides an HTTP API that enables Users/Systems to implement message oriented service using the Publish/Subscribe Model over plain HTTP.

Various other services rely on AMS:

- Operations Portal

[3] https://wiki.eosc-hub.eu/display/EOSC/EOSC+Services+User+documentation

- Accounting
- FedCloud
- ARGO Availability and Reliability Monitoring Service

- EGI Accounting

EGI Accounting stores user accounting records from various services offered by EGI, such as Cloud, HTC and storage usage. It works thanks to a network of message brokers that transfer usage data from the host to a central repository of information. The data is handled securely and can be consulted online through the EGI Accounting Portal[4].

- Application Database

The EGI Applications Database (AppDB) is a central service that stores and provides to the public, information about:

- **software solutions** in the form of native software products and/or virtual appliances,
- the **programmers** and the **scientists** who are involved, and
- **publications** derived from the registered solutions
- enabling users to **deploy and manage Virtual Machines** to the EGI Cloud infrastructure through the VMOps Dashboard

- GOCDB

GOCDB is the official repository for storing and presenting EGI and WLCG topology and resource information. It is a definitive information source, intentionally designed to have no dependencies on other operational tools for information. Because GOCDB is a primary data-input source, the portal applies a range of business rules and data-validations to control input. It applies a comprehensive Role-based authorization model that enables different actions over different target resources. The Role model allows communities to manage their own resources where users with existing roles can approve or reject new role-requests

- Helpdesk

The EOSC-hub Helpdesk allows users to submit their requests through a single-entry point. The unified system acts as first level of support, which automatically forwards tickets to the appropriate underlying support system.

- Marketplace

EOSC Portal Catalogue and Marketplace provides a platform which enables different kinds of users, with different skills and interests to discover, access, use and reuse a broad spectrum of EOSC Resources. It not only does offer advanced compute and data services from publicly-

[4] https://accounting.egi.eu

funded and commercial organisations, it also allows researchers and institutions to focus on value creation and increase the excellence of research and European competitiveness.

- Service Version Monitoring (SVMON)

The software version monitoring framework SVMON collects the information on software versions from EUDAT services and their components installed in EUDAT CDI.

These documents are the baseline documents for the integration of user-community or national infrastructure services in the scope of EOSC-Pillar project.

Addition specific documents to support integration with further services will be produced and made available according to the needs which may emerge in the course of the EOSC-Pillar project.

4 Future work and Conclusions

As already stated, these guidelines are the preliminary version which will be updated according to the input collected within the project, specifically the needs which will emerge from the relevant infrastructures in the countries participating to the project.

The final version of the guidelines will be described in deliverable D7.1 (Guidelines and Recommendations for the Technical Integration of Resources and Services with the EOSC), which is due at month 24 (June 2021).

If a different, and agreed, definition of “federation” will emerge, it will be considered and the final version of the guidelines will take it into consideration.