



EOSC-hub

D10.3 EOSC Hub Technical Architecture and standards roadmap v1

Lead Partner:	INFN
Version:	V1
Status:	Under EC review
Dissemination Level:	Public
Document Link:	https://documents.egi.eu/document/3417

Deliverable Abstract

This deliverable describes the Service Architecture of the EOSC-Hub, detailing the different service types, and defining their functions and their relationships with other components of the architecture and end users. Services in the Hub have been categorised in Access Enabling services and Research Enabling services. Access Enabling services includes Federation services (authentication and authorisation, monitoring, accounting, etc.) and Open Collaborative services (open science platforms for discovering and sharing research digital objects). Research Enabling services includes Common services (e.g. EGI Cloud Compute or EUDAT B2SAFE), and Thematic services (e.g. scientific applications offered by the Research Infrastructures). This ecosystem of services offered by the Hub enables a simple end-to-end composition of the services, leveraging on the adopted open interfaces and standards, allowing the easy and fast development of scientific services.



COPYRIGHT NOTICE



This work by Parties of the EOSC-hub Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). The EOSC-hub project is co-funded by the European Union Horizon 2020 programme under grant number 777536.

DELIVERY SLIP

<i>Date</i>	<i>Name</i>	<i>Partner/Activity</i>	<i>Date</i>
From:	Giacinto Donvito	INFN/WP10	19/01/2019
Moderated by:	Małgorzata Krakowian	EGI Foundation/WP1	18/01/2019
Reviewed by:	Małgorzata Krakowian	EGI Foundation/WP1	
Approved by:	AMB		

DOCUMENT LOG

<i>Issue</i>	<i>Date</i>	<i>Comment</i>	<i>Author</i>
V.0.1	28/06/2018	Initial structure	Giacinto Donvito (INFN), Stefano Nicotri (INFN), Diego Scardaci (EGI Foundation), TCOM members
V.0.2	28/11/2018	Completed content	Giacinto Donvito (INFN), Stefano Nicotri (INFN), Diego Scardaci (EGI Foundation), TCOM members
V.0.3	3/12/2018	Revision and corrections	Giacinto Donvito

			(INFN), Stefano Nicotri (INFN), Diego Scardaci (EGI Foundation)
V.0.4	18/12/2018	Revision and corrections	Giacinto Donvito (INFN), Stefano Nicotri (INFN), Diego Scardaci (EGI Foundation)
V.1	14/01/2018	Final revision	Giacinto Donvito (INFN), Diego Scardaci (EGI Foundation)

TERMINOLOGY

The EOSC-hub glossary of terms is available at: <https://wiki.eosc-hub.eu/display/EOSC/EOSC-hub+Glossary>

Contents

1	Introduction.....	8
2	The EOSC-hub architecture.....	9
2.1	2.1 Description of the baseline architecture.....	9
2.2	Relationships between service types.....	10
2.3	Initial EOSC-hub service catalogue.....	11
2.4	End-to-end composition of services.....	12
3	Extending the Hub - new services boarding.....	14
3.1	Levels of integration.....	14
3.2	Minimum level of integration according to the service category.....	14
3.3	Service description template.....	15
3.4	Criteria and procedures to include services into the architecture.....	16
3.5	Service Maturity (TRL 7, 8 and 9).....	17
4	Service offer in the Hub.....	19
4.1	Service categories.....	20
5	Technical areas.....	22
5.1	TCOM area 1: Data Platforms for Processing.....	22
5.2	TCOM area 2: Data Publishing and Open Data.....	23
5.3	TCOM area 3: Data Preservation/Curation/Provenance.....	25
5.4	TCOM area 4: Metadata Management and Data Discovery.....	26
5.5	TCOM area 5: HTC/HPC Compute.....	28
5.6	TCOM area 6: Cloud Compute (including containerisation and orchestration).....	31
5.7	TCOM area 7: Software Release and SQA.....	33
5.8	TCOM area 8: Federation tools.....	34
5.9	TCOM area 9: PaaS Solutions.....	35
5.10	TCOM area 10: Workflow management and user interfaces and Data analytics.....	37
5.11	TCOM area 11: Security.....	38
5.12	TCOM area 12: AAI.....	38
6	Conclusions.....	40
7	References.....	41
	Appendix I. EOSC-hub Service Description Template.....	42
	Appendix II. Service description.....	59

Executive summary

This deliverable describes the Service Architecture of the EOSC-Hub, detailing the different service types, their functions and their relationships with other components of the architecture and end users.

Services in the Hub have been categorised in **Access Enabling services** and **Research Enabling services**. Access Enabling services includes **Federation services**, guaranteeing the operation of the Hub itself and support the EOSC Service Management System, and **Open Collaborative services**, including open science platforms for discovering and sharing research digital objects like datasets, scientific applications, code, pipelines and virtual appliances. Research Enabling services includes **Common services**, providing added value to exploit storage, compute and data resources and can be re-used by other services (e.g. EGI Cloud Compute or EUDAT B2SAFE), and **Thematic services**, scientific applications like those offered by the Research Infrastructures, research data, advanced data brokering and analysis capabilities for specific research communities and multidisciplinary research.

Services offered by the Hub can work together to offer integrated solutions to the end users. For example, the **Research Enabling services** (the user-facing services offered by the Hub) can benefit of the features offered by **Federation services**, **Common services** and **Collaborative services** for implementing generic features, in this way the developers can focus on designing and implementing the scientific features reducing the time-to-market. The same **Research Enabling services** can be also combined to create a new solution, for example a data repository used as source of data for an analysis framework or two or more analysis algorithms to create workflow.

Integration between services in the Hub is made easy by the large adoption of open and standard interfaces. Indeed, EOSC-hub services were selected also according to their openness, with a preferences for services that offer open interfaces and adopt well-known standards. This key criteria allowed the creation of an ecosystem where the the **end-to-end composition of the services** can be simply achieved, enabling the easy and fast development of scientific services. This ecosystem can be considered one of the most important added value provided by the Hub, allowing users to select various services offered by the Hub and compose them according to their needs to create added-value solutions for research. Although we are still far from offering the possibility to create custom composition of any type of service, we have already identified key service workflows to be enabled and related integration activities have been undertaken in technical work-packages of the project (WP5, WP6 and WP7) leveraging on the openness of the services in the catalogue. First achievements on integration are already available and presented in this document.

The EOSC-hub service catalogue, initially providing EOSC with an initial set of existing mature services (at least TRL 8 *System complete and qualified*) selected from the catalogues of **EGI** and **EUDAT**, two largest Pan-European e-infrastructures, mature Research Infrastructures (like **CLARIN**, **WLGC**, **GEOSS**, **DARIAH**, **WeNMR**, etc.) and other relevant initiatives like the **INDIGO-DataCloud** project, will be continuously extended during the project lifetime to include data and thematic services of pan-European relevance. In order to do so, the project runs a network of Competence

Centres involving early adopters, and a stakeholder engagement programme aiming at reaching out to new user groups and service providers. A procedure to on-board new services in the catalogue has been defined and is depicted in this deliverable.

Services of the Hub are presented to the end users through the service catalogue in EOOSC-hub Marketplace. The Marketplace is a tool that allows potential customers to look for services, retrieve information (link to the services, documentation, success stories, etc.) and submit expression of interest via orders. Furthermore, the Marketplace enables the creation of project as group of composable services.

1 Introduction

EOSC-hub aims at developing a service ecosystem for researchers and innovators to discover, access, use and reuse a broad spectrum of resources for advanced data-driven research. In particular, the project delivers a catalogue of services, software and data from the EGI Federation, EUDAT CDI, INDIGO-DataCloud and major research e-Infrastructures, such as ELIXIR, CLARIN, DARIAH, etc. and builds on mature processes, policies and tools from the leading European federated e-Infrastructures to cover the whole life-cycle of services, from planning to delivery.

This document presents the EOSC-hub Technical Architecture and the first release of standards roadmap. It is organized as follows:

- Section 2 presents the EOSC-hub technical architecture, describes the service categories and how these services can work together.
- Section 3 describes the procedures and process we have defined to extend the service offer of the Hub onboarding new services in the catalogue.
- Section 4 depicts how the services are offered to customers and end-users
- Section 5 describes the organisation of the technical coordination in the project presenting the technical areas. For each area, technical details about adopted open interfaces and standards are presented.

2 The EOSC-hub architecture

In this section we describe the baseline architecture of the Hub, introducing the services currently available in the catalogue and the process to manage the evolution of the architecture in such a way that EOSC-hub services can be suitable for a growing number of user communities and scientific disciplines.

We also focus on the definition of the technical requirements needed for a service in order to be part of the official catalogue, explaining the Rules of Participation including the possible level of integration and the criteria and procedure for being included.

The last part of the section addresses how the end users could interact with and consume the services. The concepts of service portfolio, service catalogue and Marketplace are introduced.

2.1 2.1 Description of the baseline architecture

The baseline architecture of the hub includes different service types and defines their functions and their relationships with other components of the architecture and end users.

Services have been categorised as follows:

- **Access Enabling services:**
 - **Federation services:** they guarantee the operation of the Hub itself and support the processes and procedures of the EOSC Service Management System (SMS). This category includes services dealing with AAI, monitoring, accounting, operations management, order management, security and incident response procedures, etc.
 - **Open Collaborative services:** this category includes open science platforms for discovering and sharing research digital objects like datasets, scientific applications, code, pipelines and virtual appliances. The Application Database, the Marketplace and the Configuration Management DataBase (CMDB) are examples of services belonging to this category.
- **Research Enabling services:** user facing services offered to users and research communities by means of the EOSC Marketplace. They can be further split in:
 - **Common services:** provide added value to exploit storage, compute and data resources and can be re-used by other services (e.g. EGI Cloud Compute or EUDAT B2SAFE). High-throughput computing, cloud computing, storage, data management and other specialised services from local, regional and national digital infrastructures in Europe. Those services provide different levels of abstraction that allow exploiting functionalities like: storing and fetching data, deal with metadata of files and datasets, deploying application and services on-top of cloud resources, manage docker container, etc.
 - **Thematic services:** scientific applications like those offered by the Research Infrastructures, research data, advanced data brokering and analysis capabilities for specific research communities and multidisciplinary research. An initial set of

services for Humanities, Physical Sciences, Earth Sciences, Biological Sciences, Medical and Health Sciences is included.

2.2 Relationships between service types

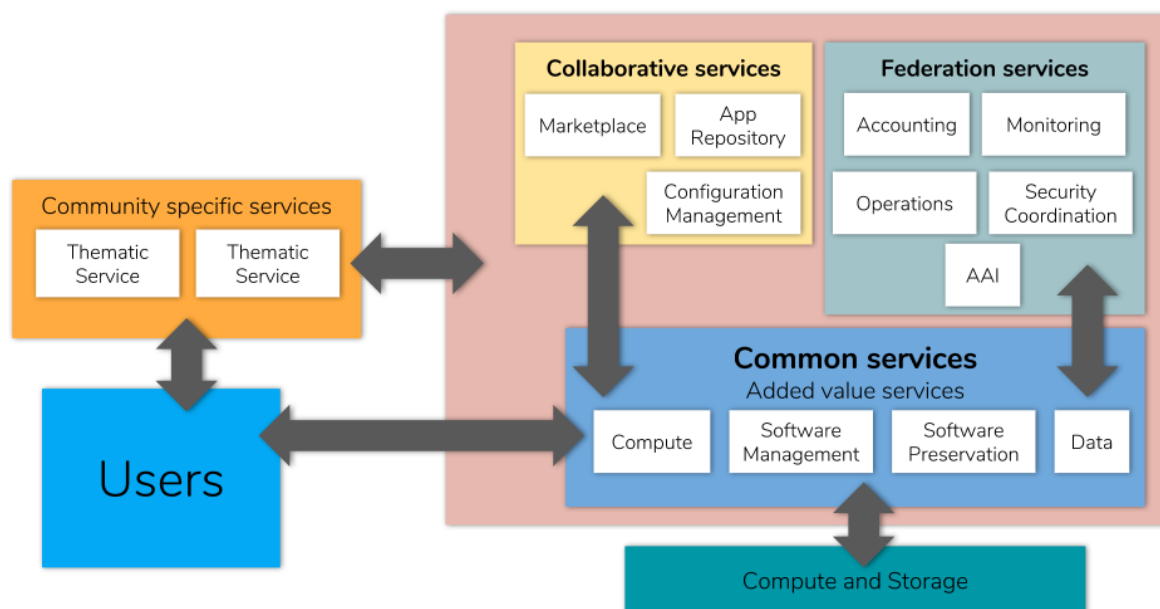


Figure. The EOSC-Hub Service Architecture

Services offered by the Hub can work together to offer integrated solutions to the end users.

For example, the **Research Enabling services** (the user-facing services offered by the Hub), can benefit of the features offered by other services in the catalogue in order to implement their final function, e.g. allowing to execute a certain scientific application. Indeed, they do not need to re-implement basic features, like authentication and authorisation, accounting, monitoring, etc., but can adopt the EOSC-hub **Federation services** for this aim. Furthermore, Research Enabling services can use the **Common services** to easily exploit advanced functionalities to handle compute and storage resources, data repositories, software, etc. and can rely on the **Collaborative services** for discovering and sharing research digital objects like datasets, scientific applications, code, pipelines and virtual appliances.

Integration between services in the Hub is made simple by the large adoption of open and standard interfaces. This allows the creation of an ecosystem enabling an easy and fast development of scientific services. The developers can focus on the scientific added value and rely on well-established services from main European e-infrastructures and other relevant initiatives (EGI, EUDAT and INDIGO-DataCloud) both for implementing basic features (AAI, accounting, monitoring, etc.) and exploiting in the best way compute, storage and data resources.

2.3 Initial EOSC-hub service catalogue

The EOSC-hub service catalogue provides EOSC with an initial set of existing mature services (at least TRL 8 *System complete and qualified*¹) which were selected during the preparation of the proposal and proven against the following criteria:

- The value proposition of the service is defined with the related functional capabilities. A web page with the description of the service (including its features) is available.
- The service was deployed in an operational environment and successfully used in real-world scenarios by end-users, all its components achieved the expected performances level within the scope. Links to either an available running instance of the service or to the release notes are available.
- User and admin manual are available and enable effective use and operation of the service within the defined scope.
- Helpdesk channels are available for support, bug reporting and requirements gathering.

Furthermore, openness was another qualifying criteria adopted to select services, preferring the ones implementing open interfaces and adopting well-established standards.

As a result of this process, around fifty services were selected and included in the initial catalogue² spanning all the categories defined above, Access Enabling (Federation and Collaboration services) and Research Enabling (Common and Thematic services) and coming from the service catalogues of **EGI**³ and **EUDAT**⁴, two largest Pan-European e-infrastructures, and mature Research Infrastructures like **CLARIN**, **WLGC**, **GEOSS**, **DARIAH**, **WeNMR**, etc. The initial service catalogue was further enriched with services delivered as outcome of the **INDIGO-DataCloud** project⁵.

During the first month of the project, it was decided to split the service catalogue in two:

1. the **internal catalogue**, containing access enabling services developed as part of the project and necessary for the operation of the EOSC-hub (e.g. helpdesk and AAI).
2. the **external catalogue**, containing common services which many services depend on (data, compute, orchestrators) and research-enabling services offering services to the end user, typically building on the common services.

This separation allows defining ad-hoc procedures to manage in a more appropriate way services belonging to these two categories that have very different targets.

During the project lifetime, the service catalogue will be continuously extended to include data and thematic services of pan-European relevance according to user requirements and to the latest technological developments. In order to do so, the project runs a network of Competence Centres involving early adopters, and a stakeholder engagement programme aiming at reaching out to

¹ Technology Readiness Level:

https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf

² Initial EOSC-hub service catalogue: <https://wiki.eosc-hub.eu/pages/viewpage.action?pageId=13336588>

³ <https://www.egi.eu/>

⁴ <https://eudat.eu/>

⁵ <https://www.indigo-datacloud.eu/>

new user groups and service providers. The procedure to onboard new services in the catalogue is described later in this deliverable.

2.4 End-to-end composition of services

As mentioned in the previous sections, EOSC-hub services were selected also according to their openness, with preferences for services that offer open interfaces and adopt well-known standards. Detailed information about relevant standards, protocols and interfaces adopted in EOSC-hub are provided later in this deliverable. This was key criteria to enable the end-to-end composition of the services that can be considered one of the most important added values provided by the Hub.

Indeed, the objective is allowing users to select various services offered by the Hub and compose them according to their needs to create added-value solutions for research. In this context, the user can be a service provider selecting a set of Access Enabling and Common services to implement a new scientific service, so re-using well-established components for the basic features like AAI, but also an end-user that composes together different scientific algorithms, offered as a service by the Hub, creating new workflows/solutions that use as input datasets made available by the Hub as well. A pre-condition to achieve this objective is the interoperability between services that requires the adoption of open interfaces.

Although we are still far from offering the possibility to create custom composition of any type of service, we have already identified key service workflows to be enabled and related integration activities have been undertaken. First achievements are already available and shortly summarised later in this section with some integration samples. The integration activities undertaken in the technical work packages of the project (WP5, WP6, WP7 and also WP8) have been planned with the aim to increase composability of the services of the Hub. Some examples are:

- interoperability between the AAI services (EGI Check-in, EUDAT B2ACCES and INDIGO-DataCloud IAM) behind the EOSC-hub AAI, implementing the guidelines defined in the AARC project⁶, that enables the single sign-on on top of many EOSC-hub services and their joint usage;
- the discoverability of EGI DataHub datasets via B2FIND: data output of analysis performed in the EGI Cloud Compute infrastructure and stored in the EGI DataHub are made automatically discoverable via EUDAT B2FIND;
- staging data stored in EGI DataHub by B2STAGE for processing: B2STAGE stages data to the EGI Data Hub to make them available as input for analysis;
- sharing processed data by B2SHARE: data processed in the EGI FedCloud are moved via B2STAGE to B2SHARE to be shared.

More details about integration activities are available or will be made available in the WP5, WP6 and WP7 deliverables.

⁶ <https://aarc-project.eu/>

Integration activities will continue for all the rest of the project to increase the composability of the services of the Hub, leveraging on open interfaces and standards, with the aim of automating as much as possible the service integration.

3 Extending the Hub - new services boarding

Service onboarding within EOSC-hub is the process whereby a new service joins the EOSC-hub service catalogue and EOSC Marketplace. This provides services with all the benefits offered by the catalogue and marketplace - promotion of the service to users outside their local community domain a single gateway for users to discover and use services, regardless of their nature and the scientific discipline of the user, and potential integration with other services in the catalogue.

3.1 Levels of integration

In the context of categorising services and differentiating between internal and external catalogues, different levels of integration of the service providers into the EOSC-hub SMS have been identified, see D4.1⁷. The current approach within EOSC-hub to define different levels of integration complies to one of the main principles of the EOSC to be open and to be as inclusive as possible to any service provider that complies with the Rules of Participation. Depending on the level of integration within the EOSC(-hub) infrastructure, a service provider shall comply with the different levels of the Rules of Participation.

Currently three levels of integration have been defined with the related operational requirements:

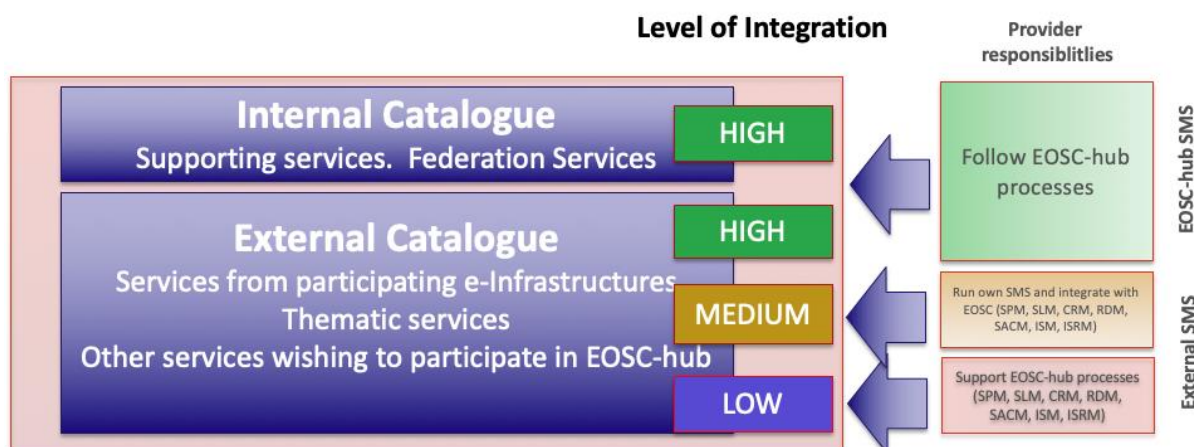
- **LOW level** has the minimum set of SMS requirements. Services which enter the catalogue at this level may either have a less mature SMS which they plan to develop or a mature SMS but would like to join the EOSC-hub initially without committing additional resources for integration until a later stage.
- **MEDIUM level** is aimed at services in the external catalogue that are being delivered as part of an existing and mature SMS complying with the majority of requirements of FitSM or other recognised Service Management Framework.
- **HIGH level** is for services delivered as part of the EOSC-hub SMS. Services with a High level of integration are expected to participate closely with the EOSC-hub and expected to be represented at the Service Management Board meetings.

3.2 Minimum level of integration according to the service category

For each of the identified service classes a minimum level to be compliant to have been identified. This can be translated to a set of operational requirements according to the level definition:

- Access enabling services: must satisfy the requirements of the *High* level of integration.
- Researchers enabling services:
 - Common services: must satisfy the requirements of the *Medium* level of integration.
 - Thematic services: must satisfy the requirements of the *Low* level of integration.

⁷ D4.1 Operational requirements for the services in the catalogue:
<https://documents.egi.eu/document/3342>



3.3 Service description template

All the services (e.g. common, thematic and federation) proposed for integration into the Hub have to be described in a homogeneous way. For this reason, EOSC-hub has been working on a service description template (SDT). In the context of the development of the EOSC service catalogue, EOSC-hub is collaborating with the eInfraCentral (eIC) project⁸ to define a standardized SDT. The EOSC service catalogue is the union of the eInfraCentral service catalogue⁹ and EOSC-hub catalogue¹⁰, both branded as EOSC services. The EOSC-hub SDT¹¹ is based on the SDT of eIC¹² and has been used with modifications where necessary.

The main difference between the EOSC-hub and eIC SDTs is introduction of new fields and marking fields as either mandatory or optional. In particular, the EOSC-hub SDT includes additional parameters for Marketplace service description options, the order management and service delivery process and additional operational requirements, as for example, a security contact point. For the EOSC Portal, it was decided to use shared SDT as outcome of an harmonisation between the EOSC-hub and eIC SDTs that should be completed early 2019. In the meantime, for the launch of the Portal, it has been agreed to adopt the eIC SDT as basis and to provide an appendix describing the additional requirements and description fields for the inclusion of the services into the Hub.

⁸ <http://einfracentral.eu/>

⁹ <http://catalogue.eosc-portal.eu/home>

¹⁰ <https://marketplace.eosc-portal.eu/>

¹¹ See Appendix I

¹² <https://jnp.gitbooks.io/service-description-template-v1-12/content/>

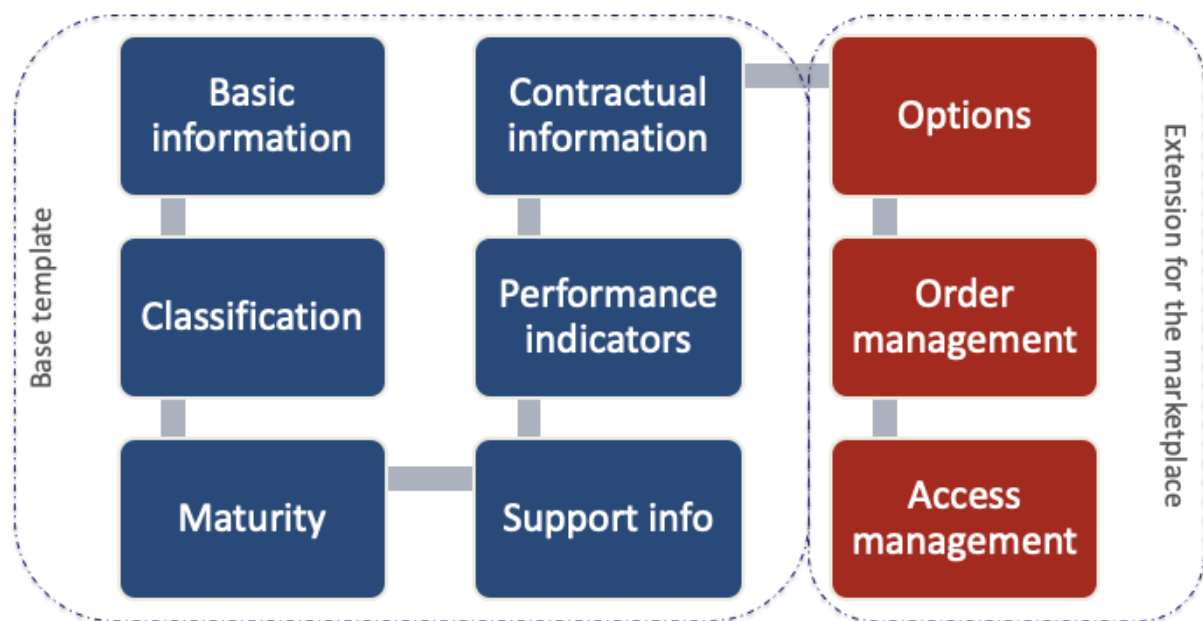


Figure. The EOSC-Hub Service Design Template: the base template is in common with eIC.

3.4 Criteria and procedures to include services into the architecture

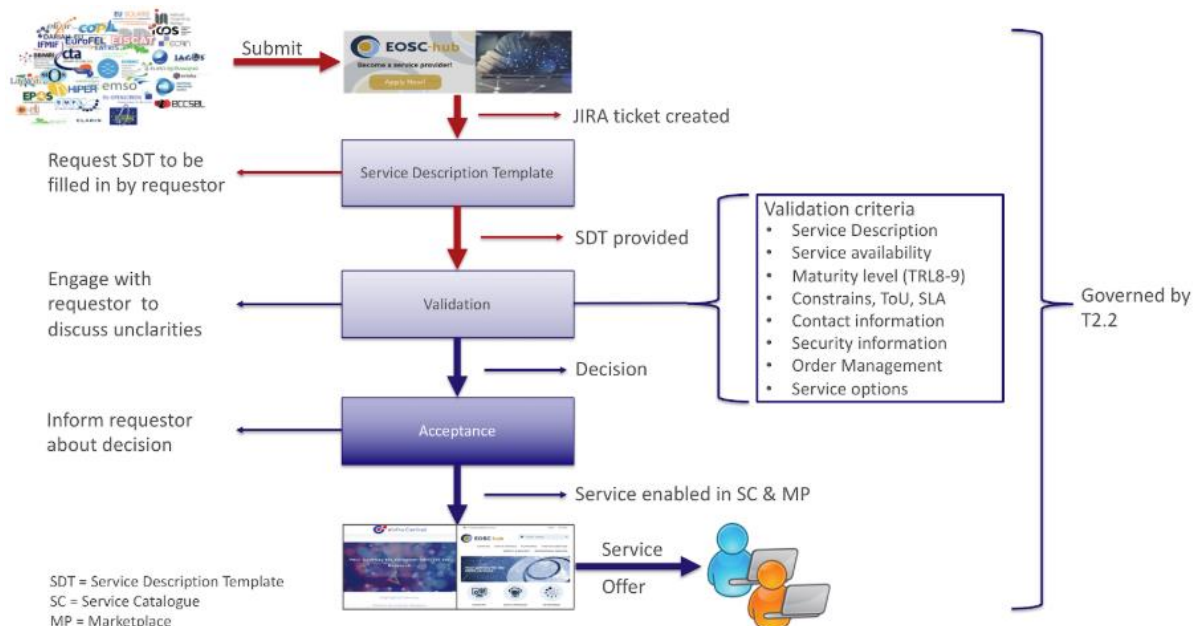
During the first year of the project, to lower the barrier for service providers to enter the EOSC the focus was to define the **Rules of Participation (RoP)** to comply with the LOW level of integration targeting service providers willing to:

- promote their service through the Hub;
- manage access and order requests from the researchers and communities willing to use EOSC services.

Despite the RoP are still under construction, for the purpose of the EOSC launch, the RoP for LOW level service providers to promote their services via the EOSC Service Catalogue and Marketplace has been initially defined.

The only requirements currently requested to add a service to the Hub with the LOW level of integration are:

- The service provider has to fill in the EOSC Service Description Template with all the requested parameters. See section 3.2.
- The service maturity should be at least TRL 8. See next section for details on how to assess the service maturity.
- A security contact for the service provider has been identified.



After the inclusion of a LOW level service in the catalogue, a nominated contact from the Service Provider is invited to engage with the Service Provider Forum by joining its mailing list and attending quarterly meetings. The Service Provider Forum is designed to facilitate communication between Service Providers and the EOSC-hub project, in addition to requirements gathering.

Effort on defining a more complete procedure¹³ to include services into the Hub is ongoing, in particular to cover also the higher levels of integration.

3.5 Service Maturity (TRL 7, 8 and 9)

This guide proposes characteristics to help assess the service maturity of a service via the operational definition of the Technology Readiness Level (TRL) indicators: TRL, 7, 8 and 9.

These definitions are results of the EOSC-hub RoP Task Force:

- TRL 7 (EC description - "System prototype demonstration in operational environment")
 - Service has passed through development and is an advanced stage of pre-production: the software is stable, reliable and has been deployed in an operational environment.
 - Functionality as required by the target users are documented, understood, validated with target sample users and accepted by them. Internal documentation exists regarding preliminary validation tests.
 - An assessment has been made of the required load of the system once the transition into production is complete and a plan has been made to service this load. This assessment has been documented.
 - An SLA is optional.
- TRL 8 (EC description: "System complete and qualified"):

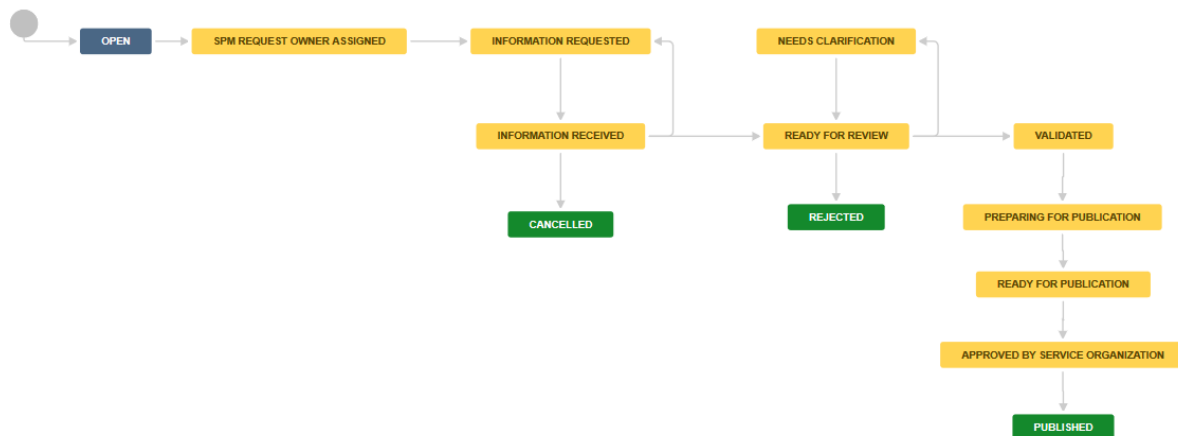
¹³ Service Provider Onboarding procedure: <https://wiki.eosc-hub.eu/display/EOSC/Service+Provider+Onboarding>

- There are users who are making real use of the service and rely on it for their work.
- Service documentation for end-users exists and is made available.
- Acceptable use policy/terms of use/SLA is in place.
- Evidence that the service is being delivered in a way consistent with user expectations.
- Provision is made for user support, with response to incident and problem management.
- TRL 9 (EC description: "Actual system proven in operational environment"):
 - All requirements of TRL 8 are met.
 - Customer feedback is gathered and documented. The service has been in a production state and relied upon by users for at least 1 year and evidence is provided to show this.
 - There are quantitative outputs as a direct result of the service usage.

3.5.1 Assessment of the candidate services

The assessment of the candidate services happens in the context of Service Portfolio Management (SPM) process of the EOSC-hub SMS. A dedicated procedure has been defined: *SPM1 Add a service in the EOSC-hub Service Portfolio*.

The workflow for the evaluation of the proposed services is shown in the following picture.



This procedure will change during the project lifetime according to the evolutions on the EOSC-hub SDT, rules of participation, etc.

3.5.2 Future evolution of the Rules of Participation

In the perspective of increasing the composability of services, we are evaluating the possibility to create a set of tags that allow identifying services that adopt certain open interfaces and standards. This is essential to enable the automatic creation of service workflows in the future. Anyway, this activity is still in early stage and requires further analysis and discussions that are planned for the first part of the 2019.

4 Service offer in the Hub

Services of the Hub are offered to the end users through the service catalogue in the Marketplace. The Marketplace is a tool that allows potential customers to look for services of their interest, retrieve information (link to the services, documentation, success stories, etc.) and to submit expression of interest via orders. Furthermore, the Marketplace enable the creation of project as group of composable services the user decides to access.

The screenshot shows the European Open Science Cloud Marketplace interface. At the top left is the logo for 'EUROPEAN OPEN SCIENCE CLOUD MARKETPLACE'. To its right is a search bar with the placeholder text 'Find service...', a dropdown menu for 'All services', and a search icon. Further right is a 'Login' button. Below the search bar is a blue sidebar menu titled 'ALL SERVICES' with the following categories: Compute, Data management, Networking, Processing & Analysis, Security & Operations, Sharing & Discovery, Storage, and Training & Support. Below the sidebar is a light blue box for 'For providers' with the text: 'Contribute to develop EOSC into a rich environment with a wide range of services and resources for researchers. Become a provider'. The main content area features a large yellow and orange banner with the text 'In the EOSC Marketplace you can find the services you need for your research'. Below the banner are three statistics: '46 providers', '57 services', and '32 countries', connected by arrows. At the bottom is a 'Popular services' section with four service cards: 'Training infrastructure' (By: EGI Federation, For: Research organisat...), 'SpotOn' (By: Rijswet Center, UT..., For: Researchers), 'Sentinel Hub' (By: Sinergise, For: Researchers, Resea...), and 'PROMINENCE' (By: EGI Federation, For: Research organisat...).

Figure The EOSC Marketplace.

The EOSC Marketplace is accessible through the EOSC Portal and, together with the eInfraCentral catalogue, composes the database of services behind the Portal.



Figure The EOSC Marketplace and the eInfraCentral catalogue compose the EOSC database of services behind the EOSC Portal.

4.1 Service categories

Services published in the Hub (catalogue and marketplace) have been categorised to allow potential end-users to easily identify services that are interesting for them.

The service categories used in the Hub have been defined in collaboration with the eInfraCentral project to have a common set of categories to be published in the EOSC Portal under the menu item "services & resources":

- Sharing & Discovery, Processing & Analysis, Data Management, Compute, Storage, Networking, Training & Support, Security & Operations.

These categories that are currently used in the EOSC portal should be considered as temporary. Indeed, EOSC-hub, in collaboration with eInfraCentral, will revise them with a broader consultation. Furthermore, both eIC and EOSC-hub will continue to work on the definitions for each of these categories, on the mapping of the sub-categories to each of main categories and on the re-mapping of services that are part of the existing services indexed by eIC portal and EOSC-hub marketplace.

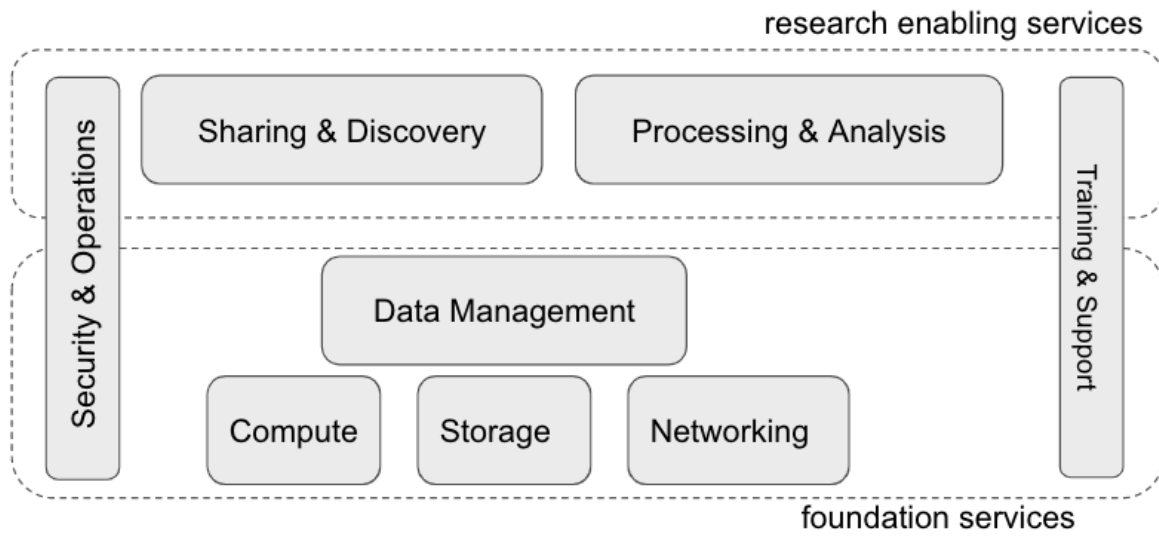


Figure Service categories in the EOSC Portal, EOSC Marketplace and eInfraCentral catalogue.

5 Technical areas

This Section briefly describes the main thematic areas in which the EOSC-Hub services could be split.

In this section we will use a different service categorization, much more oriented at explaining features and capabilities that each service is able to provide in the context of the EOSC Architecture.

The services have been grouped by the Technical Area they belong to, depending on their features. This has mainly two positive effects: collecting together all the services providing features strictly connected to each other, and making easier for product teams to interact with other services that share similar areas of interest. We also describe standards and protocols used in the implementation of each area. This information is important in order to understand the issues regarding interoperability among the services of the EOSC-Hub, but also to let the external services interact with the common services, to exploit the computing and storage resources.

Moreover we provide detailed information about the interfaces provided or exploited by the services in order to interoperate with each other.

Wherever it is pertinent, for each technical area the interoperability and interaction among the EOSC-Hub services is addressed, describing how common services exploit Federation and Collaboration services.

The Technical Areas are:

- Data Platforms for Processing
- Data Publishing and Open Data
- Data Preservation/Curation/Provenance
- Metadata Management and Data Discovery
- HTC/HPC Compute
- Cloud Compute (inc Containerisation and orchestration)
- Software Release and SQA
- Federation tools
- PaaS Solutions
- Workflow management and user interfaces and Data analytics
- Security
- AAI

5.1 TCOM area 1: Data Platforms for Processing

Challenges coming from existing and emerging scientific applications are more and more focused on the issues related to efficiently and transparently accessing and processing huge datasets. Data volumes generated by experiments or *in silico* are becoming too big to be stored and processed in

a single site, and the users are forced to move their computations to federated infrastructures, either research-oriented or commercial. Unfortunately, this poses several problems, such as the need to adapt application code to handle data access interfaces of a given cloud provider, the various authentication and authorization methods, and to manually prestage data in advance for the computation tasks.

In order to address these issues, EOSC-hub will support use cases and services with intensive data processing needs by providing and integrating the following services:

- **EGI DataHub** - DataHub is a service based on Onedata, which enables federated access to data repositories provisioned using the Oneprovider (Onedata component) service directly on the computing VM without any prestaging
- **EGI Online Storage** - Online Storage allows storing data in a reliable and high-quality environment and sharing it across distributed teams. Data can be accessed through different standard protocols (e.g. GridFTP and SRM) and can be replicated across different providers to increase fault-tolerance

Data access on computational worker nodes without any need of prestaging and manual management will be handled by Onedata, by enabling automatic, on-demand, block based data prefetching based on the POSIX requests from user applications and automatically caching the files (up to the defined capacity) based on an analysis of file popularity. In most cases, prestaging is not necessary at all, as the data blocks are fetched on the fly when requested for reading; however, it provides a REST API for manually controlling data replication or integrating it with third party services.

For data ingestion and storage of the produced results, the EGI Online Storage will be used, which provides highly-scalable capacity with the features of assigning global identifiers to files and controlling data sharing.

Relevant standards and protocols

In order to provide a seamless user experience, the integration within this platform area will focus on enabling access to EGI Online Storage from Onedata by means of GridFTP protocol driver.

In this TCOM area, the following standard and protocols are involved: GridFTP, SRM, CDMI, HTTP/Webdav, OpenID Connect, X.509.

On EGI Online Storage, data can be accessed through different standard protocols (e.g. GridFTP and SRM) and can be replicated across different providers to increase fault-tolerance.

5.2 TCOM area 2: Data Publishing and Open Data

In order to foster the idea of Open Science reproducibility and to stimulate the optimal use and reuse of research data, data need to be consistently managed and maintained according to the

FAIR principles (*findable, accessible, interoperable and re-usable*) within a secure and trustworthy environment. An essential step within the research data lifecycle and within the FAIR principles is to provide access to scientific artefacts, to share data with fellow researchers to foster scientific collaboration and to store the results in community based, institutional or public repositories to support open data and open science.

Important aspects to ensure when making research data accessible are:

1. data and metadata objects must be retrievable via persistent identifiers
2. data must be logically organised in data collections
3. data collections need to be properly described according to well established formal metadata standards

To extend findability and discoverability, the data repositories will expose the metadata to discovery services.

For the services in this technology area it is important to collaborate with services from other areas and within the joint collaboration with OpenAIRE. The main focus will be on establishing and adopting common standards and guidelines for data providers, in order to produce metadata descriptions that make data and metadata harvestable.

The EOSC-hub services included in this area are:

- **B2DROP** - B2DROP is a secure and trusted data exchange service for researchers and scientists; it keeps their research data synchronized and up-to-date, ready to be shared and exchanged during research with other fellow researchers
- **B2SHARE** - B2SHARE is a user-friendly, reliable and trustworthy way for researchers, scientific communities and citizen scientists to store and publish research data from diverse contexts. It facilitates research data storage, guarantees long-term persistence of data and allows data, results or ideas to be shared worldwide.

Relevant standards and protocols

B2DROP: is based on Nextcloud as base technology and Nextcloud provide 2 basic APIs:

- Webdav for basic upload and download of data
- OCS for the management of a Nextcloud instance

Detailed information on the APIs can be found in the Nextcloud developer manual. To support federated login Nextcloud supports next to basic username/password, also LDAP, SAML, OAuth and OpenID. To register and authenticate users in B2DROP, it has been integrated via SAML with B2ACCESS.

Nextcloud also has an application framework allowing its extension with specific applications, that can be used by the communities to develop community specific functionalities. For example, B2DROP has been integrated with B2SHARE to directly publish data objects and the CLARIN community has developed a Language Research Switchboard app to allow automatic uploads from B2DROP into the Dropzone of the LRS.

B2SHARE: is based on Invenio as base technology and is integrated with the following services:

- B2HANDLE for the registration of Persistent Identifiers
- DataCite for the registration of DOIs (optional)
- B2ACCESS for user management and authentication
- B2FIND for the harvesting of metadata from a B2SHARE instance (optional)
- B2DROP for easy access and upload of data objects (optional)
- B2NOTE to support annotations on data objects (optional)

Data in B2SHARE are organised in datasets, called records, each with a DOI referring to the landing page of the dataset in B2SHARE, and B2HANDLE PIDs referring to the datasets and individual data objects. The datasets are described via community descriptive metadata. B2SHARE supports community domains and communities are allowed to define a metadata template per community domain. The metadata templates are based on a common base template which can be extended with community specific fields. The base metadata template complies with the B2FIND, OpenAIRE and DataCite guidelines for data providers. To make datasets stored in B2SHARE discoverable and findable by communities and researchers, B2SHARE is automatically harvested by B2FIND.

To support easy upload of and the creation and management of datasets in B2SHARE, the B2SHARE service has a HTTP REST API. Via the REST API communities can integrate the B2SHARE service within their working and portal environment.

Federating services

The current services (e.g. B2SHARE and B2DROP) included in the Data Publishing and Open Data TCOM area are EUDAT services and are therefore integrated with the federating services of EUDAT. For the management and authentication of users of B2SHARE and B2DROP, both are integrated with B2ACCESS, which is the central AAI within the EUDAT CDI infrastructure. To assess B2SHARE and B2DROP to the availability and reliability requirements of the SLA and OLA the services are being monitored via the ARGO monitoring system. The DPMT is being used as configuration management database to manage configurations of individual instances and to manage resource pledges and usage metrics for the different data projects in which B2SHARE and B2DROP are being used. To provide support to users of B2SHARE and B2DROP, the EUDAT website provides a contact and support request page. Support requests result in a helpdesk ticket in the EUDAT helpdesk service (e.g. TTS). In EUDAT Service Portfolio Management is conducted via Service Portfolio Management Tool (e.g. SPMT), which manages the whole lifecycle, from initial thoughts, proof-of-concept, pilot, pre-production, production until the end-of-life of a service.

To promote B2SHARE and B2DROP towards communities and users from EOSC-hub, they are included in the EOSC-hub Service Catalogue and Marketplace

5.3 TCOM area 3: Data Preservation/Curation/Provenance

The preservation of data and the tracking of provenance metadata are preliminary steps for data curation. The domain of such technical area includes all the tools that support such processes

within EOSC-hub. Data preservation implies storing data in repositories with a suitable level of safety, in order to grant integrity even after many years. This result is achieved enforcing policies on a geographically distributed set of repository instances, which guarantees the bit stream preservation of the data. Data curation starts where the bit stream preservation ends and its objective is the full implementation of the FAIR principles. In particular, reproducibility is possible only whenever the provenance of the data is correctly tracked.

Moreover, another pillar of data preservation is the fact that research communities must trust the tools offered by EOSC-hub. In order to gain the trust of the most demanding communities, some repositories offer additional guarantees about data preservation and curation.

The EOSC-hub services supporting the aforementioned features are:

- B2SAFE, a highly-available multi-purpose service that allows community repositories to implement data management policies on their research data, distributed across multiple administrative domains.
- the European Trusted Digital Repository (ETDR), a service provided to ensure that digital information remains findable, accessible, interoperable and reusable over time. A group of Trusted Digital Repositories (TDRs) that have been granted the Core-Trust seal (<https://www.coretrustseal.org>) certification will be progressively integrated into the catalogue to offer a sustainable long-term data preservation service.
- B2HANDLE, which provides a service suite to manage persistent identifiers for data to make them referenceable independently of location, ownership or storage type. B2HANDLE supports metadata associated with identifiers to enable middleware services to decide and execute data management procedures autonomously.

Relevant standards and protocols

To enable interoperability with other services, B2HANDLE offers a REST interface for CRUD operations on identifiers and associated metadata, and supports Python libraries that simplify interaction with these interfaces. Moreover, it offers identifier management in a supporting role to other services (e.g., B2SAFE, B2SHARE, B2FIND).

5.4 TCOM area 4: Metadata Management and Data Discovery

Metadata management refers to structured provisioning of metadata to allow an easy search for and quick identification of the correct source of required data. The goal within this area is to create a common discovery layer of EOSC-hub for searching, identifying, interlinking and annotating data and metadata.

In addition, metadata management supports the enrichment, curation and quality assurance of metadata. The implementation of good metadata management is guided by the FAIR principles, including the establishment of common standards and guidelines for data providers and end

users. Close cooperation and coordination with other EOSC services and initiatives such as OpenAire Advance and the EOSCpilot 'Data Interoperability' is hereby essential.

Good metadata management is usually realised in the form of integrated metadata catalogues and annotating services that support the process of indexing, tagging, structuring and annotating data collections.

The EOSC-hub services included in this area are:

- **B2FIND** - B2FIND is a cross-disciplinary research data discovery service based on a comprehensive joint metadata catalogue that spans a wide spread scope of diverse metadata collected from heterogeneous sources which are searchable and findable via certain search functionalities and referenceable due to unique identifiers. It will become the central indexer of (meta) data distributed within and beyond EOSC-hub.
- **B2NOTE** - B2NOTE is an annotation service, which allows easily creating, searching and managing annotations. An annotation is a keyword or commentary attached to a data object (data collection, file) that explains or classifies it. Annotations are based on the W3C Web Annotation data model, serialized in JSON-LD and can be created and accessed using a RESTful API.

While findability is already addressed (by name) by B2FIND, it is of course a central feature and is implemented via the elaborated discovery portal with powerful faceted search. Data accessibility and reusability is supported by offering persistent identifiers and information about provenance, licences and citation in the presented metadata. Interoperability is the central aim, realized by the cross-discipline and wide-spread scope of the underlying metadata catalogue and following international and accepted standards:

The modular metadata ingestion workflow of B2FIND is implemented with standardized interfaces and protocols:

- **Harvesting:** B2FIND preferably uses the standardized protocol OAI-PMH to harvest metadata from the data providers. Additionally other API's like CSW, SparQL or JSON-API are supported by B2FIND.
- **Mapping:** the community specific metadata records, formatted in domain specific formats and schemas, are mapped by B2FIND to a unique and generic target metadata schema based on the DataCite4.1 metadata schema. The mapping of the metadata elements and facets is hereby based on standardized vocabularies and ontologies (e.g. the field 'Language' is mapped on the ISO 639 library).
- **Uploading and Indexing:** the search portal and the graphical user interface is based on the open source software CKAN, which comes with the Apache Lucene SOLR Servlet allowing indexing of the mapped JSON records and performant faceted search.

Relevant standards and protocols

Annotations created via B2NOTE are based on the W3C Web Annotation data model and serialized in JSON-LD. Using these standards and through a reduced API which is aligned with the OpenAPI description B2NOTE boosts interoperability with other services and web clients.

Federating services

Data Discovery indirectly relies on federated AAI and computing tools. For example, in order to access datasets found in the discovery portal, persistent identifiers and access licenses are provided in metadata. Whether the referenced data resource can be directly retrieved or the user gets linked to a landing page depends on the access policies of the data archives. Similar re-use and further processing of data objects found in the discovery portal is supported by offering provenance metadata and references to federated computing resources.

Federated services as accounting is important to collect information about the usage of and statistics of data search and access requests submitted to the discovery portal. Monitoring is used to determine and provide system availability and reliability metrics.

Similarly the B2NOTE service is using with federated services like accounting, monitoring and others.

5.5 TCOM area 5: HTC/HPC Compute

High Throughput Computing (HTC) and High Performance Computing (HPC) are two terms related to the way computing intensive applications are implemented. Typically, HPC refers to increased capability, addressing highly-coupled problems with multiple computing resources, meanwhile HTC refers to increased capacity, addressing many loosely-coupled problems, again with multiple computing resources. HTC is related to the management, reliability and dynamic scheduling of many individual jobs. HPC requests fast interconnection and support of parallel computing environments.

In HTC models, resource availability may (should) vary during execution time, while HPC executions normally keep the resource allocation constant. HTC models can run on distributed, hybrid resource infrastructures and HPC models benefit from coherent, centralised resources. HTC model will not speed-up single problem executions, as HPC could do, and HPC applications must be specifically implemented for a parallel computing model.

Despite the fact that this requirement relates mainly to the job scheduling, it has some strong relations with the way the (virtual) computing infrastructure is provisioned, directly linked to TCOM Area 6.

A preliminary analysis of the user community requirements identifies that 8 communities need HTC (Marine, ELIXIR, GEOSS, DODAS, WeNMR, OpenCoastS), only 2 explicitly mention the need for HPC (EOS, ECAS/ENES), and 2 communities have not yet expressed need for HTC/HPC (CLARIN, DARIAH). Another important point is that several communities (WeNMR, OpenCoastS) already identify an HTC job scheduler in the EOSC Service catalogue (DIRAC4EGI), and other communities (ELIXIR, GEOSS), state the need for a container-based or even function-based HTC model which can be supported with an Elastic Kubernetes as a Service plus additional services. At least 2 communities (DODAS, ECAS/ENES) have their own scheduling systems, so they are more relevant to the TCOM Area 6.

The identified services are:

- EGI Workload Manager (ex DIRAC4EGI), a workload management service to distribute and centrally manage thousands of computational tasks on cloud and HTC
- EGI High-Throughput Compute, to execute thousands of computational tasks to analyse large datasets.
- Advanced IaaS, a set of common solutions including Docker support for OpenStack and OpenNebula and on HPC clusters, such as uDocker, which can ease the execution of Dockerized applications in HPC resources.

Indirectly (for the case of Kubernetes as a Service), the following services will be relevant:

- PaaS Orchestrator, TOSCA-based deployment orchestration on multiple IaaS, as ECAS/ENES already have their scheduling system based on this technology.
- CVMFS, Application software distribution service, used in DODAS.
- EGI Cloud Compute and EGI Container Compute, to provide resources.

Relevant standards and protocols

We can group the above services into:

1. Services related to the scheduling of batch jobs: in this group we consider the EGI High-Throughput Compute and the EGI Workload Manager. The EGI High-Throughput Compute and EGI-High Throughput MPI are the supporting infrastructures, which the EGI Workload Manager is a scheduling system to dispatch batch jobs into.
2. Services related to application delivery and execution in cloud/container-based infrastructures: application services and components (uDocker, PaaS Orchestrator and CVMFS) and infrastructure services (EGI Cloud Compute and EGI Container Compute).

As the EGI Cloud Compute and EGI Container Compute will be deeply analysed in the next section, we will briefly consider them here.

Services related to the scheduling of batch jobs

EGI High-Throughput Compute integrates a set of services of the UMD distribution¹⁴ which enable users to submit jobs in an efficient manner. Jobs are described using the Job Description Language. A set of Command Line Interface tools and software libraries is compiled in a single package¹⁵. EGI High-Throughput Compute supports MPI resources, also requested by some of the use cases.

EGI Workload Manager¹⁶ enables the interaction with the EGI High Throughput Compute infrastructure by means of a web interface, a REST API, a Python module and a Command Line Interface. It also uses the JDL¹⁷ from the EGI High Throughput Compute. Users can manage the whole job lifecycle (e.g. create, submit, monitor, retrieve job output, and clear them) from the API. EGI Workload Manager also provides a Data Management System (DMS).

For the management of the jobs, EGI Workload Manager interacts with standard LRMS such as PBS/Torque, LSF, Sun Grid Engine, Condor, BQS and Microsoft Compute Cluster, as well as GRAM-

¹⁴ https://wiki.egi.eu/wiki/UMD_products_ID_cards

¹⁵ <https://twiki.cern.ch/twiki/bin/view/LCG/EL7UIMiddleware#Description>

¹⁶ <http://dirac.readthedocs.io/en/latest/>

¹⁷ <https://dirac.readthedocs.io/en/latest/UserGuide/GettingStarted/UserJobs/JDLReference/index.html>

based Grid services. It supports POSIX and SRM for data access and X509 certificates for authentication, using VOMS extensions for authorisation. EGI Workload Manager integrates with the services of EGI Federated Cloud.

Services related to application delivery and execution in cloud/container-based infrastructures

The services in this group are related to the way applications are delivered in resource management systems with a higher level of abstraction. They apply partially or jointly to the configuration and deployment of complex application topologies and applications with complex software dependencies. In this sense we identify three approaches (which could be combined in some cases):

1. *Use of Docker containers for application delivery.* This is a growing tendency for packaging the applications and their dependencies, and matches the HTC model when combined with container management systems (which could be the EGI Container Compute or the Kubernetes as a Service provided in the EGI applications on demand). It is also related to systems that make use of Docker containers in batch systems without administrative privileges, where uDocker can be used. A combination of uDocker and the EGI High-Throughput Compute has been successfully explored¹⁸.
2. *Deploy and configure applications on the fly.* This will need the user to code their application dependencies and configurations using TOSCA¹⁹, defining the topology of a multi node application and the configuration rules to be applied. By using the PaaS orchestrator service it will be possible to deploy such applications in EGI Compute Cloud backend.
3. *Use of a shared distributed file-system remotely mounted as a POSIX.* CVMFS does this by using fuse and only requires HTTP connections and SQLITE. This approach uses an overlay-based filesystem (union) to optimise data layer distribution and can be used on EGI High-Throughput Compute, EGI Cloud Compute or even Supercomputers²⁰.

Federating services

All the services described in this section are federated by definition. They are designed to integrate and work with the compute resources of EGI and are extendable to a wide range of LRMS systems. Authentication and authorisation are federated.

How they match the use cases

The analysis has been done taking into account the requirements of the application use cases. We identify the following scenarios:

- Communities that already use some of the identified components, such as WeNMR.

¹⁸ Mónica Chillarón, Vicente Vidal, Damián Segrelles, Ignacio Blanquer, Gumersindo Verdú, "Combining Grid Computing and Docker Containers for the Study and Parametrization of CT Image Reconstruction Methods", Proc. Comp. Sci., Vol. 108, 2017, pp. 1195-1204, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2017.05.065>.

¹⁹ <https://www.oasis-open.org/committees/tosca/>

²⁰ <https://cvmfs.readthedocs.io/en/stable/cpt-hpc.html#>

-
- Communities that have their own HTC/HPC scheduling mechanism or data analysis system (ECAS/ENES, DODAS), whose needs could be addressed by providing TOSCA blueprints with the configuration required ready to be deployed on the EGI Cloud Compute infrastructure.
 - Communities that require a container resources as a service (ELIXIR CC, GEOSS), which could be addressed through the deployment of Kubernetes on top of the EGI Container Compute. In the case of GEOSS they propose function as a service models, which could be supported extending the “EGI applications on demand” catalogue.
 - Communities that have not identified the HTC/HPC scheduler, such as Marine CC, which could fit on the EGI Workload Management System, or the EOS Pillar, which could benefit from the CVMFS application delivery model. OPENCoastS also fits this scenario.
 - Communities not using HTC/HPC services yet (DARIAH, CLARIN).

5.6 TCOM area 6: Cloud Compute (including containerisation and orchestration)

This technical area covers Infrastructure as a Service (IaaS) computing services that allow users to provision computing resources as virtual machines (managed by a hypervisor that provides virtualized hardware resources) or containers (isolated user-space with shared kernel) and the tools and services that provide orchestration of IaaS resources. Cloud Computing provides users with complete control over operating systems, storage, and the entire software stack that runs their applications without the overhead of managing physical servers. Computing resources are provided on-demand, i.e. with the user unilaterally provisioning them as needed via APIs, without requiring any human interaction with the service provider. Elasticity is supported, i.e. resources can be provisioned and released to scale up or down following the user demand.

The EOSC-hub service included in this area is:

- EGI Cloud Compute to run virtual machines on demand with complete control over computing resources.
- EGI Cloud Container and Advanced IaaS to run containers on computing resources.
- TOSCA for HEAT to orchestrate and deploy complex TOSCA templates on OpenStack providers

Relevant standards and protocols

There are several non-interoperable protocols and APIs in the IaaS area, with two widely-known standard IaaS interfaces available: OGF OCCI API²¹ and DMTF CIMI²²; these never gained enough support, neither in commercial nor in open source implementations. EGI currently supports

²¹ <http://occi-wg.org/>

²² https://www.dmtf.org/sites/default/files/standards/documents/DSP0263_2.0.0.pdf

providers which expose either OpenStack API²³ or the OGF OCCl API. Other providers exposing different APIs (e.g. OpenNebula XML-RPC²⁴, AWS EC2²⁵, or GCP Compute Engine²⁶) can be supported if the integration with the federating services described in the section below is implemented.

In the EGI Cloud, interoperability between IaaS providers is not achieved at the IaaS API level, but using higher level tools like Infrastructure Manager or Terraform that allow interaction with different vendors using a common high-level configuration syntax. TOSCA is the main standard for these tools.

Other standards and protocols used in the service:

- **OAuth2.0**²⁷ and **OpenID Connect** for AAI²⁸
- **GlueSchema 2.1**²⁹ for representation of resources
- **HEPIX** image list³⁰
- **OGF UR** adapted to account Virtual Machine usage³¹

Federating services

The cloud services rely on the following federating services:

- **EGI Check-in** for Authentication and Authorisation of users into the providers. Legacy VOs leveraging X.509 certificates and VOMS proxy extensions are also supported.
- **Accounting** to collect and process the usage information of every provider in a central location
- **Monitoring** to calculate Availability and Reliability metrics for each provider
- **Configuration Database/Information discovery** (using GlueSchema 2.1) to provide discovery of providers and their capabilities
- **AppDB** cloud marketplace to provide a common VM Image catalogue and distribution of those images to the providers.

²³ <https://developer.openstack.org/>

²⁴ https://docs.opennebula.org/5.6/integration/system_interfaces/api.html

²⁵ <https://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html>

²⁶ <https://cloud.google.com/compute/docs/reference/rest/v1/>

²⁷ <https://oauth.net/2/>

²⁸ https://openid.net/specs/openid-connect-core-1_0.html

²⁹ <https://tinyurl.com/y74arzcd>

³⁰ https://github.com/hepix-virtualisation/image_list_format_docbook

³¹ https://wiki.egi.eu/wiki/Federated_Cloud_Technology#Cloud_Usage_Record

5.7 TCOM area 7: Software Release and SQA

Supporting e-Science in the EOSC-hub Infrastructure requires extensive and reliable software, for advanced computing use, deployed across over 300 European and worldwide data centres. This technical area covers ways to deliver software for the EOSC consumption. The software is compiled, validated and distributed following the Software Provisioning Process (SWPP), where the Quality Criteria (QC) definition sets the minimum quality requirements for acceptance. The growing number of software components currently existing to support EOSC infrastructure favours the adoption of automated solutions towards the manual-based validation mechanisms.

The EOSC-hub services included in this area are:

- AppDB which contains virtual appliances and application software repository and management³²
- Repositories of verified software to be deployed by the service providers.

The work of this TCOM area is harmonized with the common Quality Assurance (QA) processes proposed in the following standards:

- IEEE 730-2014 - IEEE Standard for “Software Quality Assurance.”
- ISO/IEC/IEEE 12207:2008 “Systems and software engineering -- Software life cycle processes.”
- ISO/IEC/IEEE 15289:2011 “Systems and software engineering -- Content of life-cycle information products.”

and the following guidelines and “best practices”:

- A set of Common Software Quality Assurance Baseline Criteria for Research Projects
 - <http://digital.csic.es/bitstream/10261/160086/4/CommonSQA-v2.pdf>
- ESA Guide to Software Quality Assurance
- [OWASP Foundation](#):
 - Secure Coding Practices
 - Source Code Analysis Tools
 - Dynamic Application Security Testing
 - Code Review Guide

The development of a software component within the EOSC ecosystem should follow some quality and best practices conventions followed by a continuous integration approach. INDIGO DataCloud already set Common Software Quality Assurance Baseline Criteria for Research Projects on which EOSC fits. The next items provide a short description of the guidelines considered more relevant to the EOSC services:

- Code Accessibility: following the open-source model, the source code should be open, publicly available and with a version control system.
- Licensing should adhere to an open-source license to be freely used, modified and distributed by others.
- Code Workflow and versioning should follow a change-based approach with a branching model. Semantic versioning is recommended for tagging software releases.³³

³² <http://digital.csic.es/bitstream/10261/160086/4/CommonSQA-v2.pdf>

³³ Tom Preston-Werner, Semantic Versioning 2.0.0, URL: <https://semver.org/>

- Testing:
 - Functional, focusing on covering the functionalities that the software provides.
 - Integration, guaranteeing the operation among coupled software components.
 - Acceptance, tackling user requirement compliance.
 - Scalability, evaluating the performance of the application by testing the capabilities of scaling up/down through different loads of common user requests.
- Documentation must be in an online available documentation repository and updated for each new software versions. It should also provide support for different targets (user, admin, developer)
- Security, covering:
 - Static application security testing (SAST)
 - Dynamic application security testing (DAST)

Federating services

The Applications Database (AppDB) is a service designed to interact with a number of EGI services as the EGI authentication and authorisation service, GOCDB, EGI Operations portal, PERUN, BDII, OpenAIRE (external), and many more. Compared to the repository, it is a more independent service, interacting only with the EGI RT ticketing system and the Resource Providers/sites. The repository implements all the actions related to software product release (e.g. submit, move, delete), as commanded by the RT system. Moreover, the frontend component makes the produced repositories available for the EOSC Grid & Cloud infrastructure.

5.8 TCOM area 8: Federation tools

This technical area includes the operational and collaboration tools to operate and manage a large distributed infrastructure, guaranteeing standard operation of heterogeneous resources from multiple independent providers, and to facilitate collaboration between research communities.

Federation tools are a key enabler for distributed management and processing of big data and a fundamental baseline to implement the service integration and management system of the EOSC.

Federation tools included in this area are:

- **Configuration Management Databases** - service registries for configuration management of federated services:
 - **GOCDB**: it is the Configuration Information Service for EOSC-Hub and also provides a central information service for third parties as well.
 - **DPMT**: it is a content management system for management and the implementation of data management plans during a managed enabling process. The tool registers service providers, services, service components, resources and data project requests.
- **ARGO** monitoring tool, performing service availability monitoring and reporting of the distributed service endpoints.
- The **EOSC-hub helpdesk**, which includes:

- **XGUS**: the technology adopted to create the integrated EOSC-Hub helpdesk that provides the first level of support
- **GGUS**: the helpdesk of the EGI infrastructure
- **TTS**: the helpdesk of the EUDAT CDI
- **Accounting portal** for collecting, and displaying usage information.
- **Operations Portal**, an integrated tool to collect and display information from monitoring, service configuration and user communities
- The **SVMON**: a framework to collect information about the installed software versions.
- Information system, allowing information discovery, in particular about capabilities and services available in the federation
- The **security monitoring tools**: to identify vulnerability on cloud virtual machine images
- The **Messaging service** enables reliable asynchronous messaging for the EGI infrastructure. Its most recent version offers an HTTP interface, which will make the implementation of new clients easier and more robust. It is a real-time messaging service that allows you to send and receive messages between independent applications.
- **Virtual Machine image catalogue and distribution**: allows researchers to share their virtual appliances for deployment in a cloud federation.
- The **Marketplace**: a platform where an ecosystem of services and research data, delivered by providers and partners, can be promoted, discovered, shared and accessed, including offered services as well as discipline and community-specific tools and services. It will expose all the services belonging to the EOSC-Hub service catalogue. Its backoffice allows to manage the orders, generates the related SLA/OLA documents and oversee the process to enable the access to the services. Marketplace backoffice can be connected to order management system of third parties.
- The **Service Portfolio Management Tool (SPMT)**: it provides a full list of services, with detailed descriptions for each service and its components, and allows to manage service descriptions according to the service management guidelines of FitSM
- The **Application Database (AppDB)**: a tool that stores and provides information about software solutions in the form of native software products and virtual appliances, the developers and the scientists who are involved, and publications derived from the registered solutions.

5.9 TCOM area 9: PaaS Solutions

This technical area focuses on the automated provisioning and configuration of compute (virtual machines/containers) and storage resources on top of heterogeneous cloud environments.

The services included in this area allow the users to deploy virtual infrastructures with complex topologies (such as clusters of virtual machines or dockerized applications) using a standardized interface based on the TOSCA template language (see the following).

The PaaS layer features advanced federation and scheduling capabilities ensuring transparent access to the different cloud environments, both the “traditional” cloud management frameworks,

like OpenStack, OpenNebula, AWS and Azure, and the more innovative container orchestration platforms like Apache Mesos.

The selection of the best cloud provider to fulfill the user request is performed by the PaaS orchestrator taking into account criteria like the user's SLAs, the services availability and the data location.

The main EOSC-Hub services included in this area are:

- The PaaS orchestrator, that relies on a set of PaaS microservices to schedule and coordinate the deployment workflow;
- The Infrastructure Manager that is steered by the Orchestrator to perform the provisioning of resources on the selected cloud provider.

Relevant standards and protocols

Interoperability is one of the key features of the PaaS orchestration area.

The OASIS³⁴ (Organization for the Advancement of Structured Information Standards) has defined and developed a standard for enhancing portability and operational management of cloud and other types of applications and services across their entire lifecycle.

Both the Orchestrator and the Infrastructure Manager, the core services of the PaaS solutions area, implement this standard: the TOSCA³⁵ (Topology and Orchestration Specification for Cloud Applications) language. It aims at enabling the interoperable description of applications and infrastructure cloud services, independently of the supplier creating the service, and of any particular cloud provider or hosting technology.

The deployment requests submitted by the users to the orchestration tools must adhere to the TOSCA template syntax defined by the TOSCA's YAML Simple Profile³⁶ that specifies a rendering of TOSCA providing a more accessible syntax as well as a more concise and incremental expressiveness of the TOSCA DSL (Domain Specific Language).

The adoption of the TOSCA standard ensures the portability of the deployment topology description across different cloud providers and the support of the cloud bursting use-case.

The interoperability of the PaaS orchestration tools with the other services is promoted through the provision of REST³⁷ (Representational State Transfer) API endpoints; request/response data are transferred in the compact and easy-to-use JSON data-interchange format.

Federating services

Currently the PaaS Orchestration system relies on the following services for federating cloud sites and users:

- INDIGO IAM to manage the user identities (authentication/authorization);
- INDIGO Configuration Management DB (CMDDB) to gather information about the resource providers (images, compute and storage endpoints, etc.). The CM DataBase can be populated by different sources, one of this is the CloudInfoProvider component;

³⁴ <https://www.oasis-open.org/>

³⁵ https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

³⁶ <http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/csprd02/TOSCA-Simple-Profile-YAML-v1.0-csprd02.html>

³⁷ <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest>

- INDIGO SLAM to gather user preferences according to the signed SLAs;
- INDIGO Monitoring to gather information about the metrics and the health of the cloud services.

As explained in the section “Evolution of the services”, the orchestration tools are expected to evolve in the next months in order to reach a better integration with the other EOSC-HUB services.

5.10 TCOM area 10: Workflow management and user interfaces and Data analytics

This technical area covers the graphical user interfaces that support both software developers and researchers running applications. It covers the application database as well as high-level user interface components like portals, science gateways, scientific workflows frameworks, big data analytics workflows and other user interfacing tools. Such services are built on top of underlying infrastructure services aiming at provisioning a simple and effective final user experience.

The identified EOSC-hub services currently included in this area are:

- the **Application Database** which allows sharing, discovering and using community-specific scientific software. In this context, the Application Database is one of the user interfaces which allow running on demand applications on the cloud. Among those application and tools there are scientific workflow systems as WS-PGRADE, Kepler, Chipster
- **Thematic services:**
 - Portals and science gateways:
 - WeNMR suite - Online portals for structural biology analytics (providing access to a variety of tools for structural biology and computational modelling covering NMR, cryo-electron microscopy, and integrative modelling)
 - DISVIS, POWERFIT, HADDOCK, GROMACS, AMPS-NMR, CS-ROSETTA, UNIO, FANTEN, AMBER
 - Extended descriptions of the suite portals available at <https://confluence.egi.eu/display/EOSC/WeNMR+VA>
 - DARIAH Science Gateway - A portal tailored for the digital arts and humanities communities - a web-oriented portal, developed during the EGI-Engage project (DARIAH Competence Centre), tailored for researchers coming from digital arts and humanities. It currently offers several cloud-based services and applications: Semantic and Parallel Semantic Search Engines (SSE, PSSE), DBO@Cloud, Workflow Development and supports several file transfers protocols.
 - Analytics services:
 - ECAS - climate Analytics Service (including Ophidia) - a service which allows scientific end-users to perform data analysis experiments on large volumes of research data coming from multiple disciplines. Users can define parallel processing workflows, executed remotely without the need to download

data or provide their own computing resources, as these are provided by ECAS. Moreover, users can explore workflows created and shared by others, and apply these to their own data. ECAS allows users to write a workflow once and apply it to different data without having to customize it again.

Federating services

The services described in this section rely on underlying federated infrastructure and services. They are designed to integrate and work on top of the AAI and IaaS, PaaS layers providing user interfaces and application level workflow orchestration.

5.11 TCOM area 11: Security

In order to build federations and operate a proxy-infrastructure using federated identity management, the infrastructure should follow the following policies:

- SNCTFI - <https://www.igtf.net/snctfi/> (which includes the requirement to comply with SIRTFI)
- The GEANT Code of Conduct - <https://wiki.geant.org/display/eduGAIN/Data+Protection+Code+of+Conduct+Cookbook>
- Any (other) AARC policy guidelines <https://aarc-project.eu/guidelines/>

Specifically, these policies succinctly define *inter alia* requirements for collaboration of participants in managing security incidents, processing personal data, keeping systems patched, making users aware (and requiring them to accept) acceptable use policies, etc.

In particular, it follows that EOSC-Hub must have:

- An AUP, and a way of signalling acceptance of the AUP by the users at suitable intervals (once a year)
- Instruments to handle security incidents, e.g. encrypted mailing list, respecting the traffic light protocol, with security contacts for every site subscribed to the list.
- Processes to ensure and monitor software patches at participating sites.

Relevant standards and protocols

Relevant standards and protocols for TCOM area 11 include SNCTFI, SIRTFI, REFEDS Assurance Framework, IGTF, GEANT CoCo.

5.12 TCOM area 12: AAI

The Authentication and Authorisation technical area focuses on enabling federated access to EOSC-Hub resources and support of identity management for research infrastructures. Best practices for managing identities and authentication for research infrastructures is to use federated identity management; in other words, users use identities they already have to

authenticate to the infrastructure. Specifically, it is highly recommended that e-infrastructures run an AAI, in order to manage federated identities and accommodate identity providers that use different protocols, provide different levels of assurance and connect different service providers in a more scalable way. The AARC project has defined a blueprint for such an architecture, which is implemented by EGI, EUDAT and GEANT infrastructures as well as by several research infrastructures.

Research infrastructures can operate their own identity management system which can be connected to the EOSC-Hub AAI via standardized protocols. If the research infrastructure does not have any identity management system it can leverage EOSC-Hub AAI services for identity and access management. EOSC-Hub AAI provides services for user/group/virtual organization management and all subsequent features like user enrolment flow management, identity consolidation and access control provisioning/deprovisioning to EOSC-Hub thematic services.

The EOSC-Hub AAI includes:

- **B2ACCESS, Check-in, eduTEAMS³⁸, IAM, Perun.**
- additional services such as **GÉANT Step-up Authentication Service** and other research communities AAI components as needed.

The Authentication and Authorisation technical area involves and supports below listed standards and protocols. EOSC-hub services have to support at least one of the mentioned protocols in order to be seamlessly accessible by the users. Regarding the authorization, EOSC-hub AAI supports AARC guidelines^{39 40} for authorization, therefore EOSC-hub services requiring authorization have to follow those standards as well.

Relevant standards and protocols

Supported protocols/standards by EOSC-hub AAI:

1. SAML2
2. OpenID Connect
3. OAuth2 (RFC 6749)
4. X.509
5. SCIM (RFC7642, RFC7643 and RFC7644)

Federating services

By definition, all the services pointed out in this section are federated. They are designed to integrate and work with the compute resources of EGI and are extendable to a wide range of LRMS systems. The authentication and authorisation is federated.

³⁸ In the context of the collaboration agreement between EOSC-hub and GN4-2 project.

³⁹ <https://aarc-project.eu/wp-content/uploads/2017/11/AARC-JRA1.4A-201710.pdf>

⁴⁰ <https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4E.pdf>

6 Conclusions

The main goal of Work Package 10 is to provide technical coordination within the project, identifying the scientific and technical roadmap from requirements and existing technologies, promoting standardization, defining criteria for inclusion of services into the catalogue and assessment of conformance, and helping service providers to understand technical solutions.

In this document, a first version of the technical architecture has been presented, with focus on each technical area and a detailed description of the involved services and the interactions and relations among them.

In particular, importance has been given to relevant standards and protocols used by each service, stressing their interdependence and compatibility.

This will be the starting point for the implementation of the proposed integrated solutions, and will serve as a technical reference for providers of the services of the EOSC-hub catalogue, in order to help them clarifying the complexity of the architecture.

Updates of the presented architecture, including the results developed in collaboration with and by other WPs, following the method of work of WP10, will be provided in D10.4 or other updates, if necessary.

7 References

No	Description/Link
R1	EOSC-hub Grant Agreement
R2	EOSC-hub Collaboration Agreement
R3	D1.1 Quality and Risk Management Plan
R4	D1.2 Data Management Plan
R5	D3.1 Communications and Stakeholder Engagement Plan
R6	The Plan for the Exploitation and Dissemination of Results in Horizon 2020 https://www.iprhelphdesk.eu/sites/default/files/newsdocuments/FS-Plan-for-the-exploitation-and-dissemination-of-results_1.pdf
R7	Confluence Innovation Management related pages https://confluence.egi.eu/display/EOSC/Project+Results
R8	https://confluence.egi.eu/display/EOSC/Dissemination+Activities
R9	The European Innovation Management Standard CEN/TS 16555 https://standards.cen.eu/dyn/www/f?p=204:110:0:::::FSP_PROJECT,FSP_ORG_ID:35932,671850&cs=13A816A57184977C465944D2F2E2C5645
R10	Catalogue of Project Results https://wiki.eosc-hub.eu/display/EOSC/Catalogue+of+Project+Results
R11	Catalogue of Aggregate Project Results https://wiki.eosc-hub.eu/display/EOSC/Catalogue+of+Aggregate+Project+Results
R12	EOSC-hub Deliverables Page https://confluence.egi.eu/display/EOSC/Deliverables
R13	EOSC-hub Milestones Page https://confluence.egi.eu/display/EOSC/Milestones

Appendix I. EOSC-hub Service Description Template

Service								
Attribute name	Description	Format		To be provided by	Permissions	Mapping to eInfraCentral	Mandatory/M - Optional/O	
		Type	Multiplicity				EOSC-hub	eInfraCentral
Basic information								
Service ID	Global unique and persistent identifier of the service	URN	1			Service ID	M	M
Service Name	Name of this specific service as assigned by the service provider	String	1	SO		Service Name	M	M
Service URL	Webpage with information about the service that is maintained and	URL	1	SO		Service URL	M	M

	hosted by the service provider							
Service Endpoint	Main URL to use the service (in case of networked service)	URL	1	SO		n/a	M	
Service Page (a)	Webpage with information about the service in the EOOSC-hub website	URL	1	SCM		n/a	O	
Service Marketplace Page (a)	Webpage with information about the service in the EOOSC-hub marketplace	URL	1	SCM		n/a		
Service Description	Description of the service in terms of functionalities it provides and resources it enables access to	String	1	SO		n/a		

Service Description for Web (a)	High-level description in fairly non-technical terms of what the service does, functionality it provides and resources it enables access to; written for the Web	String	1	SCM		Service Description		M
Service Description for Print (a)	High-level description in fairly non-technical terms of what the service does, functionality it provides and resources it enables access to; written for printed material	String	1	SCM		n/a		
Service Tagline	Short catch-phrase for marketing and advertising purposes (1 line). It will be usually displayed close the service name and should refer to the main value or purpose	String	1	SO		Service Tagline		O

	of the service							
Service Organisation Name	Name of the main organisation providing the service and acting as main contact point (in case the service is operated by different organisation(s), the individual service provider(s) delivering the service can be described using the "service provider" entity)	String EOSCSPR-1	1	SO		Service Provider Name		M
Service Organisation Description	Short description about the service organisation	String	1	SO		Service Provider Description		O

Service Organisation Logo	Link to the logo/visual identity of the service organisation	URL	1	SO		n/a		
User Value	The benefit to a customer and their users delivered by a service; benefits are usually related to alleviating pains (e.g., eliminate undesired outcomes, obstacles or risks) or producing gains (e.g. increased performance, social gains, positive emotions or cost saving).	String	1	SO		Customer/User Value		0

Target Customers	Type of customers who are allowed to commission this service. Restrictions may apply according to various criteria like the location (e.g. country) or type of activity (e.g. research, commercial). By customer, we mean an organisation that commissions a service provider to deliver one or more services, doing so on behalf of a number of users; customers commission a service and usually discuss the terms of the contract and of the SLA but do not necessarily use it; users use the service but do not necessarily commission it	String	1	SO		Target Customers/Users		0
------------------	---	--------	---	----	--	------------------------	--	---

Target Users	Type of Individuals that primarily benefits from and uses a service	String	1	SO				
Service Logo	Link to the logo/visual identity of the service	URL	1	SO		Service Symbol/Visual Element		O
Service Screenshots and Videos			1			Service screenshot and videos		
Service Language	Language of the user interface	Country code	1..*	SO		Service Language		O
Use cases/case studies	List of use cases supported by this service	URL	0..*	SO				

Standards	List of standards supported by the service	String	1	SO				O
Certifications	List of certifications obtained for the service from independent third parties (including the certification body)	String	1	SO		?		O
Service Maturity								
Service TRL	Used to tag the service to the Technology Readiness Level	(0,1,2,3,4,5,6,7,8,9)	1	SO		Service TRL		M

Service Phase	Phase of the service lifecycle selected among:	(discovery, planned, alpha, beta, production, retired)	1	SO		Service Life Cycle Status		M
	<ul style="list-style-type: none"> ● discovery: researching users needs, exploring technological or policy constraints (TRL: 1,2) ● planned: a plan to develop the service is defined (TRL: 3, 4) ● alpha: prototype available for closed set of users; (TRL: 5, 6) ● beta: service being developed while available for testing publicly (TRL: 7) ● production: service available in the live environment meeting security/performance 							

Service Version	Version identification that refers to a specific set of service components	String	1	SO		Service Version		O
Service Last Update	Date when this service version was released	Date	1	SO		Service Last Update		O
Service Change Log	Summary of the main changes from the previous version	String	1	SO		Service Change Log		O
Service Classification								
Service Category 1	A named group of services that offer access to the same type of resource or capabilities	(compute, storage, data, operations, training, coordination, ...)	1	SO		Service Category		M

Service Subcategory 1	A named group of services that offer access to the same type of resource or capabilities, within the defined service category	Open enumeration	1	SO		Service Subcategory		M
Service Category 2	(Optional, choose a second category if strictly needed) A named group of services that offer access to the same type of resource or capabilities	(compute, storage, data, operations, training, coordination, ...)	1	SO				O
Service Subcategory 2	(Optional, to be defined if a second category is chosen) A named group of services that offer access to the same type of resource or capabilities, within the defined service category	Open enumeration	1	SO				O

Service Tags	Comma-separated list of keywords associated to the service to be used to simplify search by relevant keywords	String	1	SO		Service Tags		O
Service Management								
Service Owner Name	Name of the person who has accountability for the whole service from a management point of view	String	1	SO		Service owner		M
Service Owner Contact	E-mail contact of the service owner	E-mail	1	SO		missing		M
Service Support Contact	Contact to request support for this service that is used by the federator (EOSC-hub) to contact the service provider for this service	E-mail	1	SO		Service Helpdesk		M

EOSC-hub Service Support	Contact to request support for this service through the federator (EOSC-hub)	E-mail	1	SCM				M
Security Contact	Contact of the person responsible for the security aspects of the service	E-mail	1	SO		?		M
Service Monitoring	Webpage with monitoring information about this service	URL	1	SO		Service Monitoring		O
Service Accounting	Webpage with usage accounting information about this service	URL	1	SO		Service Accounting		O
Service Maintenance	Webpage with information about planned maintenance windows for this service	URL	1	SO		Service Maintenance Window		O

Service Helpdesk (a)	URL to the EOSC-hub helpdesk	URL	1	SCM		Service Helpdesk		M
Service Helpdesk Support Unit	Name of the support unit in the Helpdesk	?	1	SCM				M
EOSC-hub Service Order	Webpage to order the service through EOSC-hub	URL	1	SCM		Service request		M
Service Order	Webpage to order the service directly via the main service	URL	1	SO				M

	organisation							
Service Order Procedure (a)	Procedure to pass the order from EOSC-hub to the service provider	String	1	SO				?
Service User Manual	URL to user manual and documentation	URL	1	SO		Service User Manual		0
Service Training Information	URL to training information on the service	URL	1	SO		Service Training Information		0
Service Feedback (a)	URL to page where customers can provide feedback on the service	URL	1	SO		Service Feedback		0
Service Contract								

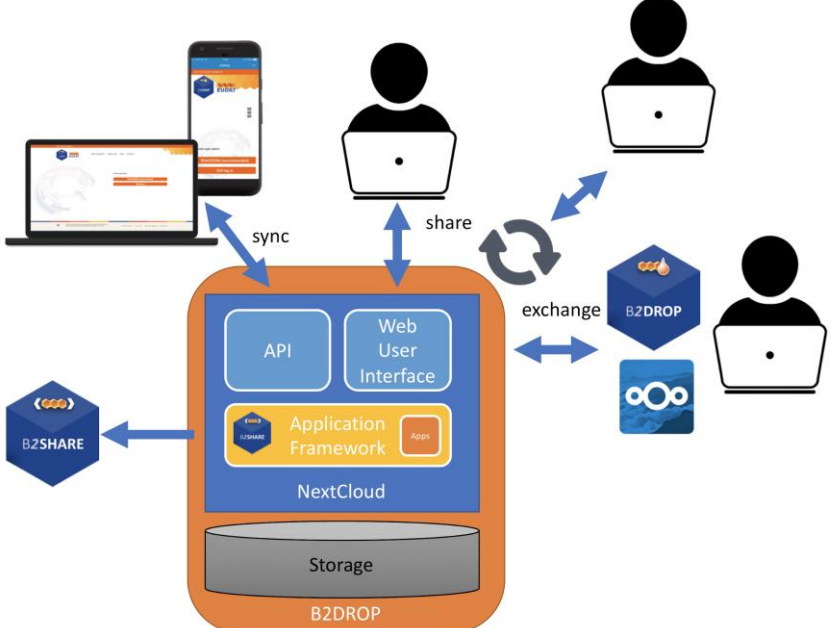
Service Terms Of Use	Link describing the rules, service conditions and usage policy which one must agree to abide by in order to use the service	URL	1	SO		Service Terms Of Use	M	O
Service Level Agreement	<p>Link to the Service Level Agreement (SLA) applicable to the service.</p> <p><i>A SLA is a documented agreement between a customer and a service provider that specifies the service to be provided and the service targets that define how it will be provided (FitSM definition)</i></p>	URL	1	SO		Service Level Agreement	M	M
Service Funding	Sources of funding for the development and/or operation of the service	(National funding, EC funding, ...)	0..*	SO		Service Funding	O	O

Service Access Policies	List of access policies for this service	Service Access Policy IDs	0..*	SO		'Service Price' can contain a URL to the EO SC-hub page with info on the access policies	M	M
Service Dependencies								
Service Components	List of service components	Service Component IDs	0..*	SO		n/a		M
Required Services	List of required services for this service to function	Service IDs	0..*	SO		Required Services		O
Related Services	List of services that are commonly used with this service	Service IDs	0..*	SO		Related Services		O

Appendix II. Service description

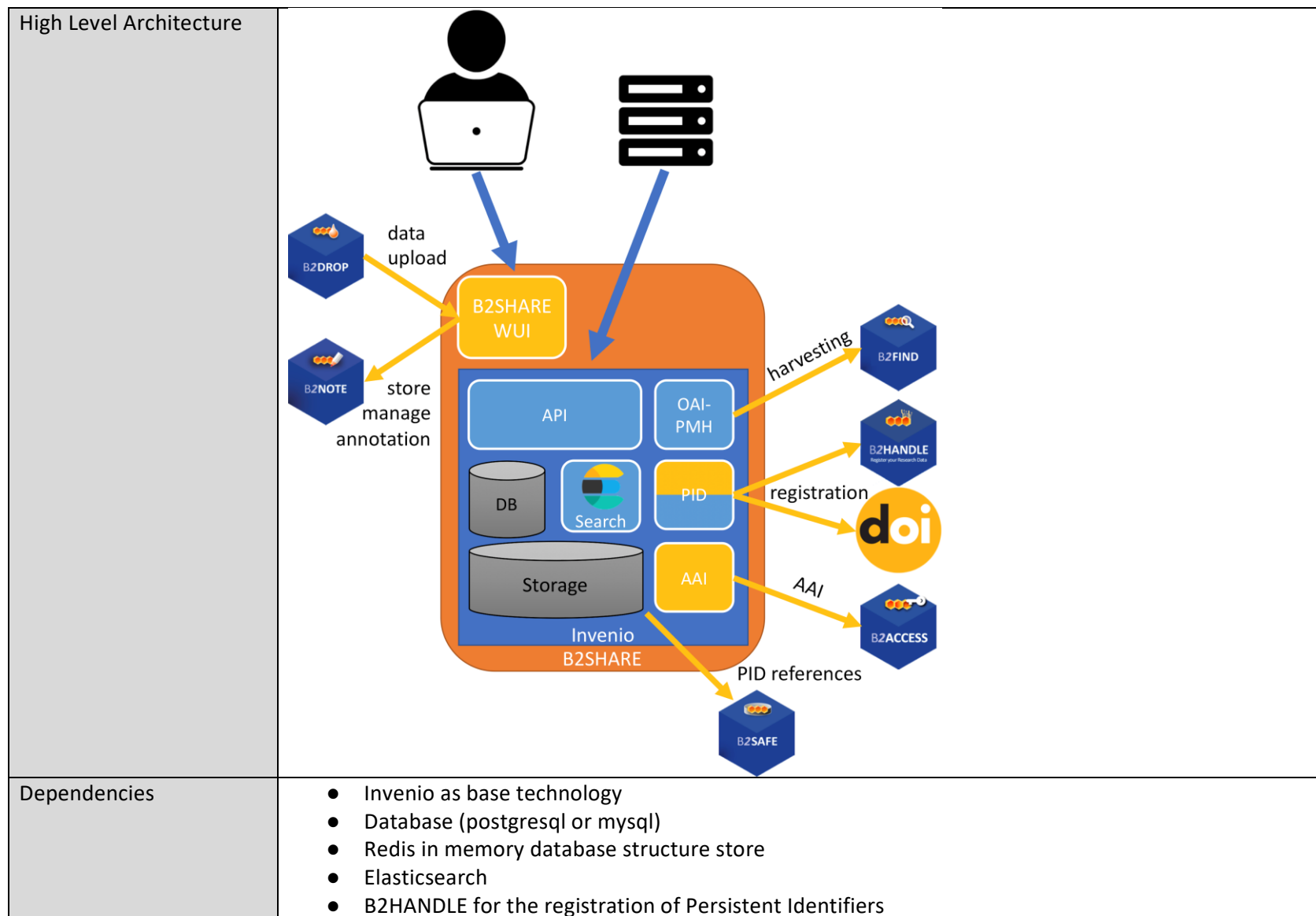
B2DROP

Identification	B2DROP: https://eosc-hub.eu/catalogue/B2DROP
Type	B2DROP is a EUDAT branded version of Nextcloud in which users can get access via self-registration. The service can also be offered as dedicated instances to communities and research groups or institutes.
Purpose	The purpose of B2DROP is to provide personal cloud storage in a trusted environment in which researchers can easily maintain share and exchange research data with fellow researchers.
Function	In the central hosted B2DROP instance of EUDAT, each registered user receives a default storage quota of 20GB. Users can easily upload and maintain their data via the web users' interface of B2DROP or via desktop and mobile sync apps. Users can exchange data with other researchers via defining sharing rules or via a HTTP based share link. Data can be shared, via federated sharing, across B2DROP and other similar services which are based on Nextcloud/ownCloud. B2DROP (and its base technology Nextcloud) is a flexible platform for applications to be developed to support specific functionality. To support open access and to enlarge the discovery of research datasets, B2DROP enables easy data uploads to B2SHARE. Communities can request to enable of specific apps to support their community.
High Level Architecture	

	 <p>The diagram illustrates the B2DROP architecture. At the center is a large orange rounded rectangle labeled 'B2DROP' at the bottom. Inside this rectangle, there is a blue box labeled 'NextCloud' which contains an 'API' and a 'Web User Interface'. Below the NextCloud box is a grey cylinder labeled 'Storage'. To the left of the B2DROP box is a blue cube labeled 'B2SHARE'. A blue arrow labeled 'sync' points from the B2DROP box to the B2SHARE cube. A blue arrow labeled 'share' points from the B2DROP box to a laptop icon. To the right of the B2DROP box is another blue cube labeled 'B2DROP' with a circular refresh icon above it. A blue arrow labeled 'exchange' points from the B2DROP box to this second B2DROP cube. Below the second B2DROP cube is a blue square icon with two white circles. To the right of this square icon is a laptop icon. At the top of the diagram, there are three user icons: a laptop, a smartphone, and a person. Arrows indicate synchronization and sharing between these users and the central B2DROP system.</p>
Dependencies	Nextcloud
Interfaces	Nextcloud provide 2 APIs: Webdav and OCS, detailed information on the APIs can be found at https://docs.nextcloud.com/server/12/developer_manual/client_apis/index.html
Data	The metadata of file objects stored in B2DROP (Nextcloud) is stored in the database, the data itself is stored on a primary storage device.
Needed improvement	<ul style="list-style-type: none"> • Optimise and extend integration of B2DROP with B2SHARE • Integration with other EOSC-hub services (e.g. Onedata, OpenStack Swift and/or EGI Federated Cloud for on-demand scaling of services) • Adopt federated sharing on basis of the GEANT OpenCloudMesh developed standard for federated sharing

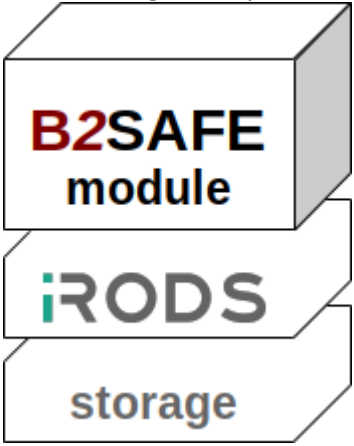
B2SHARE

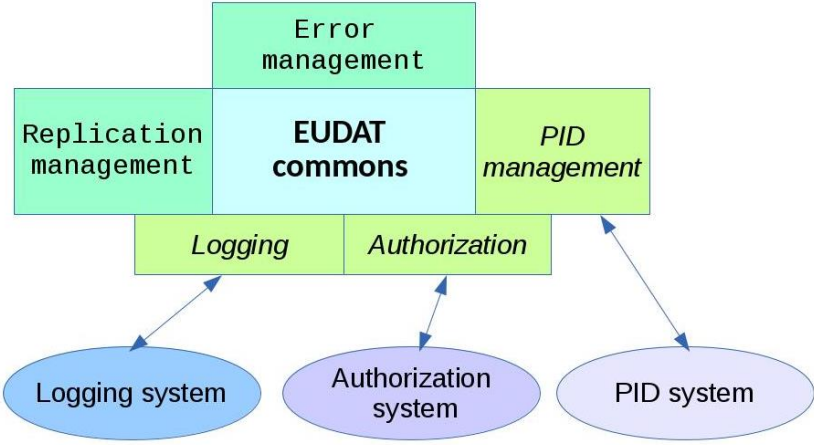
Identification	B2SHARE; https://eosc-hub.eu/catalogue/B2SHARE
Type	B2SHARE is a user-friendly, reliable and trustworthy tool for researchers, scientific communities and citizen scientists to store and publish small-scale research data from diverse contexts.
Purpose	B2SHARE is a solution that facilitates research data storage, guarantees long-term persistence of data and allows datasets, results or ideas to be shared worldwide.
Function	B2SHARE is provided as central EUDAT service in which users and communities can share datasets worldwide and to make them discoverable. B2SHARE is also provided as a technology which communities can rely on to easily setup a data repository service to persistently manage, describe and make their datasets discoverable. B2SHARE has a flexible metadata model and supports community defined metadata templates. Data objects are organised in datasets, which get a landing page and a DOI. The data objects have a persistent identifier, and for each of them checksums are calculated and download statistics are maintained. B2SHARE supports data lifecycle management and versioning. Community managers can authorise users to publish datasets in a community domain and can define a review process. B2SHARE has a REST API for easy upload and download of datasets.



	<ul style="list-style-type: none"> ● DataCite for the registration of DOIs (optional) ● B2ACCESS for user management and authentication ● B2FIND for the harvesting of metadata from a B2SHARE instance (optional) ● B2DROP for easy access and upload of data objects (optional) ● B2NOTE to support annotations on data objects (optional)
Interfaces	<ul style="list-style-type: none"> ● HTTP REST API for the upload and download of dataset ● OAI-PMH to enable harvesting of metadata
Data	<p>Metadata on datasets and data objects are stored in a database, Data are stored on a local filesystem. Metadata are exposed via OAI-PMH to enable harvesting. If data are stored in B2SAFE, then B2SHARE only stores PID references to the data objects in the database. The PIDs are automatically registered in B2HANDLE, while DOIs are registered at DataCite. B2SHARE makes use of B2ACCESS for registration of user identities and for authentication users. To optimise full text search in Invenio, metadata is automatically indexed in Elasticsearch. Annotations on data objects are stored in B2NOTE.</p>
Needed improvement	<ul style="list-style-type: none"> ● B2SHARE HTTP API integration with the B2STAGE HTTP API ● Adapting the OpenAIRE Guidelines for data providers in B2SHARE ● Integration of B2SAFE with B2SHARE to allow data access and to support data discovery ● Extend B2SHARE to support diverse data organisations ● Improve two-way integration between B2SHARE and B2NOTE ● Further integration of B2SHARE with other EOSC-hub service to allow data publishing from EGI and INDIGO services

B2SAFE

Identification	B2SAFE: https://eosc-hub.eu/catalogue/B2SAFE
Type	The B2SAFE Service is implemented as a package on top of iRODS , providing a set of iRODS rules and scripts.
Purpose	The Service offers functionality for the long term data preservation, allowing community and departmental repositories to implement data management policies on their research data across multiple administrative domains in a trustworthy manner
Function	The service provides an abstraction layer of large scale, heterogeneous data storages and guards against data loss in long-term archiving. It allows to optimize access for users (e.g. from different regions) and brings data closer to facilities for compute-intensive analysis. The main feature is the function to replicate data sets across different data centres in a safe and efficient way while maintaining all information required easily finding and querying information about the replica locations. The information about the replica locations and other important information are stored in a PID (Persistent Identifier) registry.
High Level Architecture	<p>The B2SAFE service, at the core, exploits the iRODS rule engine in order to implement a set of actions to implement specific behaviour defined in data management policies.</p> <div style="text-align: center;">  <p>The diagram illustrates the high-level architecture of B2SAFE. It consists of three stacked, 3D-style rectangular blocks. The top block is labeled 'B2SAFE module' in red and black text. The middle block is labeled 'iRODS' in green and black text. The bottom block is labeled 'storage' in black text. The blocks are connected by a vertical line, suggesting a layered or dependent architecture where B2SAFE module sits on top of iRODS, which in turn sits on top of storage.</p> </div> <p>The actions are defined by a set of iRODS rules which can be executed on regular basis or be triggered by</p>

	<p>actions like data ingest. The rules interact with external software components which deliver functionalities such as PID registration. Several python scripts facilitate the interaction. The whole B2SAFE module, including the python scripts, can be conceptualized as modules and represented in the following architectural overview:</p> 
Dependencies	<ul style="list-style-type: none"> ● iRODS server, either version 4.0.X, 4.1.X or 4.2.X (User Documentation - iRODS Deployment) ● a Handle prefix (User Documentation - PIDs in EUDAT) with installed private public keys and certificates (see B2HANDLE documentation)
Interfaces	<ul style="list-style-type: none"> ● Persistent Identifiers (PIDs) are associated to the data and registered in the B2HANDLE service ● Persistent Identifiers (PIDs) are globally resolvable, they can be used in B2SHARE and B2STAGE services ● HTTP API and GridFTP allow downloading and uploading data using standard protocols. ● A PAM based module allow the B2SAFE service to support the oauth2 protocol, therefore it is compatible with B2ACCESS ● A webDAV interface is supported
Data	<p>The service is data agnostic in relation to both the data content and the data format. It supports natively metadata as key-value pair.</p>
Needed improvement	<p>Mainly integration enhancements, as described in the M6.2, the integration rolling plan.</p>

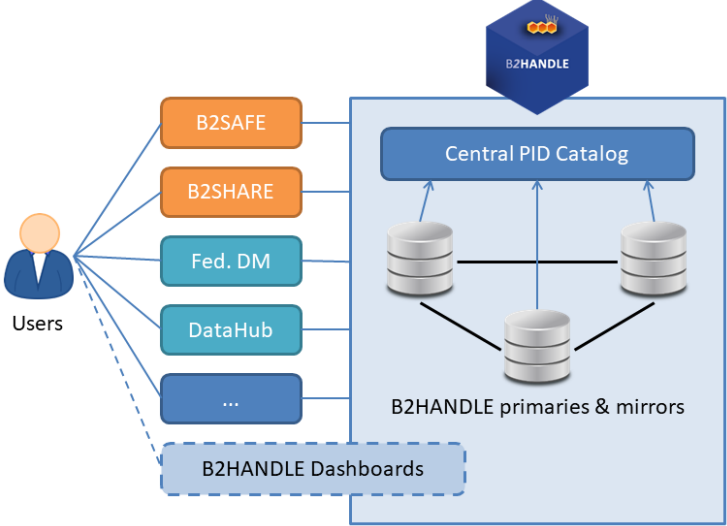
B2FIND

Identification	B2FIND: http://b2find.eudat.eu
Type	The B2FIND service is implemented as a metadata catalogue and a search portal, based on CKAN , and a modular metadata ingestion workflow.
Purpose	The service offers functionality for publishing metadata, allowing communities and data repositories to make research data collections visible, searchable and accessible in a wide cross-discipline range
Function	The service provides a discovery portal with powerful search features at interdisciplinary level. Heterogeneous metadata are directly harvested from data providers, mapped to a common metadata schema and indexed in the metadata portal. This allows searching and accessing research data distributed over a wide range. Access and reuse is supported by providing persistent identifier referencing to the underlying data collections. While the main feature is the findability and identifiability of research output, (meta) data curation and enrichment are supported during negotiations with data providers through the ingestion process.
High Level Architecture	<p>The B2FIND architecture is based on the CKAN software that comes with an SOLR indexer, a PostgreSQL database and an extensible user interface and web server.</p> <p>The B2FIND ingestion workflow includes harvesting of metadata records from the metadata repositories, mapping of the heterogeneous and specific formatted datasets onto the B2FIND metadata schema and uploading and indexing the metadata sets in the discovery portal.</p>

	<p>The diagram illustrates the B2HANDLE workflow. It starts with a Data provider (Community) providing Harvest specification (OAI-URL, OAI subsets, MD formats) and Mapping specification (XPath rules, Community specific schemas and ...). This feeds into 1. MD Generation and Specification, which leads to 2.a MD Provider. 2.a MD Provider then feeds into 2.b MD Harvesting. 2.b MD Harvesting feeds into 3. MD Mapping and Validation, which is supported by EUDAT-B2FIND (Serviceprovider). 3. MD Mapping and Validation feeds into 4. MD Uploading and Indexer. 4. MD Uploading and Indexer feeds into 5. MD Search and Data Access, which is supported by a User (Scientist / Re-Researcher). The workflow is supported by five modules: Module 01 (01.a MD Generation, 01.b MD Specification), Module 02 (02.a MD Provider, 02.b MD Harvester), Module 03 (03.a MD 'Mapping', 03.b MD Validation), Module 04 (04 MD Uploader and Indexer), and Module 05 (05.a MD Search, 05.b Data access).</p>
Dependencies	<ul style="list-style-type: none"> ● CKAN2.4 (CKAN API)
Interfaces	<ul style="list-style-type: none"> ● Implementation of OAI-PMH, JSON-API and other APIs to harvest metadata from data provider endpoints standard. ● Handles are associated to the B2FIND datasets using the B2HANDLE service ● Synchronized harvesting from B2SHARE services
Data	The service is data agnostic with respect to both the metadata content and schema. .
Needed improvement	Automation of the ingestion workflow and enhancement of the search functionality and usability.

B2HANDLE

Identification	B2HANDLE: https://eosc-hub.eu/catalogue/B2HANDLE
Type	B2HANDLE is implemented as a federation of PID servers, supported by common policies and processes.

Purpose	B2HANDLE offers a trustworthy and reliable service to manage persistent identifiers for data and other digital resources, with a focus on supporting data management tasks in middleware / e-infrastructure processes.
Function	B2HANDLE offers multiple technical interfaces to request actions on persistent identifiers (create, read, update, delete, and search). Reliability and scalability support is transparent for users. The Central PID Catalogue allows basic reverse-lookups and simple filtering queries.
High Level Architecture	<p>B2HANDLE is offered by service providers in Europe as a federated service. Each service provider maintains a number of Handle servers and additional components, which serve a number of prefixes (Handle namespaces). The servers are mirrored to provide scalability and reliability. The Central PID Catalogue is a redundant system of meta-mirrors across multiple primary sites.</p>  <p>The diagram illustrates the B2HANDLE architecture. At the top is the B2HANDLE logo. Below it is a large light blue box containing the 'Central PID Catalog' (a blue rounded rectangle) and 'B2HANDLE primaries & mirrors' (represented by three database server icons). A 'Users' icon (a person) is on the left, connected by solid lines to five service boxes: B2SAFE, B2SHARE, Fed. DM, DataHub, and an ellipsis (...). A dashed line connects the 'Users' icon to a dashed blue box labeled 'B2HANDLE Dashboards'. Arrows point from the service boxes to the Central PID Catalog, and from the Central PID Catalog to the primaries & mirrors. The primaries & mirrors are interconnected with bidirectional arrows.</p> <p>B2HANDLE currently focuses on delivering its service to support other e-Infrastructure services. With the dashboard as a future option, users may directly work more intensively with the B2HANDLE suite.</p>
Dependencies	<p>B2HANDLE can be used by communities directly for their data management or identification purposes, or by other EUDAT and EOSC services such as B2SAFE, B2SHARE and B2FIND. Customer services will use the standard interfaces of B2HANDLE to create and modify identifiers in real time.</p> <p>B2HANDLE relies on the DONA Handle System to provide PID functionality and on ePIC as intermediary.</p>

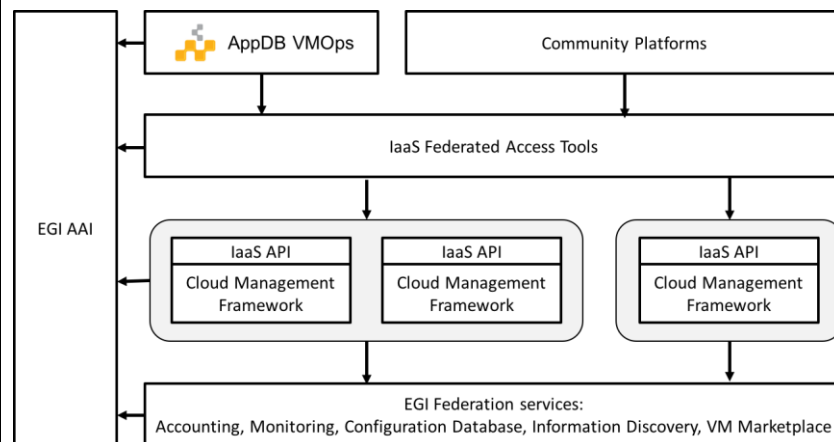
Interfaces	<ul style="list-style-type: none"> ● PID management (CRUD) interfaces: Native Handle JAVA API, REST interface supported by pyhandle/b2handle Python libraries. ● PID viewer (web GUI): Native Handle System viewer. ● Central PID catalogue HTTP interface, accessible via pyhandle/b2handle Python libraries.
Data	PID information is stored in SQL databases (MariaDB).
Needed improvement	<ul style="list-style-type: none"> ● Performance improvements, possible implementation via database optimization and additional message queueing system ● Extension of Central PID Catalogue functionality ● Provisioning of a management dashboard, providing high-level overview on namespace status and low-level detail views into small groups or individual identifiers ● Development of PID mass-management tools to better support more complex curation scenarios such as server relocations, community schema and policy changes, schema profile checking and identifier target monitoring

Cloud Compute

Identification	EGI Cloud Compute: https://www.egi.eu/services/cloud-compute/
Type	Compute, IaaS
Purpose	Cloud Compute provides a platform for running VM-based applications.
Function	Cloud Compute gives you the ability to deploy and scale virtual machines on-demand. It offers guaranteed computational resources in a secure and isolated environment with API access, without the overhead of managing physical servers.
High Level Architecture	<p>Cloud Compute is a federation of Infrastructure as a Service (IaaS) resource centres that deploy a Cloud Management Framework (CMF) providing management of Virtual Machines and of persistent Block Storage devices that can be associated to them. These end-user capabilities must be provided via community agreed APIs that must be integrated with the following EGI services:</p> <ul style="list-style-type: none"> - Check-in to provide Single Sign-On for authentication and authorization across the whole cloud federation. - GOCDB, to record information about the topology of the e-infrastructure. - Accounting to collect, aggregate and display usage information.

- Monitoring to perform federated service availability monitoring and reporting of the distributed cloud service endpoints, and to retrieve this information programmatically. Integration with monitoring is a passive activity of the resource centre, the monitoring is performed using the end-user APIs with regular user credentials from EGI Check-in
- Information Discovery, allowing users and tools to obtain information about capabilities and services available in the federation.
- AppDB Community-curated catalogue of Virtual Appliances (Virtual Machine Images) and distribution of appliances to the providers of the infrastructure.

EGI does not mandate deploying any particular or specific Cloud Management Framework; it is the responsibility of the Resource Centre to investigate, identify and deploy the solution that fits best their individual needs whilst ensuring that the offered services implement the required interfaces and domain languages of the federation realms they are member of.



Dependencies

EGI Cloud Compute depends on the underlying Cloud Management Frameworks deployed at the resource centres (OpenStack, OpenNebula and Synnefo currently supported) and several services from EGI internal catalogue: Accounting, Service Monitoring, Configuration Database, Check-in (and/or legacy X.509-based AAI)

It also depends on the Application Database (AppDB) for the VM image management and GUI.

Interfaces	<p>Users and Community platforms built on top of EGI Cloud Compute have several ways of interacting with the cloud providers:</p> <ul style="list-style-type: none"> - Directly using the IaaS APIs to manage individual resources. This option is recommended for pre-existing use cases with requirements on specific APIs. - Leveraging IaaS Federated Access Tools that allow managing the complexity of dealing with different providers in a uniform way (e.g. IaaS provisioning systems that allow to define infrastructure as code and manage and combine resources from different providers, thus enabling the portability of application deployments between them like IM or Terraform). - Using the AppDB VMOps dashboard, a web-based GUI that simplifies the management of VMs on any provider of the EGI infrastructure. AppDB VMOps in turn relies on the Infrastructure Manager (IM), a IaaS Federated Access Tool as described above
Data	The service is data agnostic in relation to both the data content and the data format.
Needed improvement	<p>Improvement areas:</p> <ul style="list-style-type: none"> - Better integration with EGI Check-in - Better support for CMF native APIs with complete information discovery about the capabilities of these APIs - Expansion of the service into commercial cloud providers - New mechanisms for information discovery leveraging messaging systems.

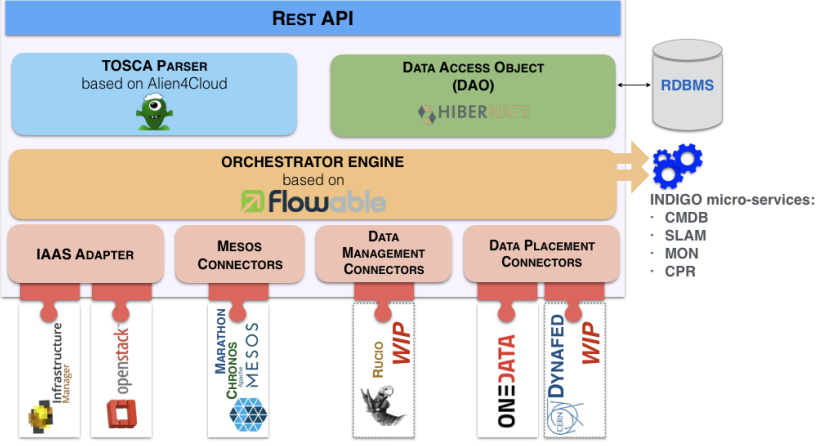
Cloud Container Compute

Identification	EGI Cloud Container Compute: https://www.egi.eu/services/cloud-container/
Type	Compute, IaaS
Purpose	Cloud Container Compute provides a platform for executing Docker container based applications using Kubernetes technology.
Function	Cloud Container Compute gives you the ability to deploy and scale Docker containers on-demand using Kubernetes technology. The service provides with easy provision of Kubernetes clusters on EGI Cloud Compute resources that can be scaled and upgraded without the overhead of installing, managing and operating the nodes.
High Level Architecture	The service deploys Kubernetes clusters for the users

	<p>EGI Cloud Container Compute</p> <p>2. Deploy Kubernetes cluster on provisioned VMs</p> <p>1. Provision cluster VMs</p> <p>IaaS Federated Access Tools</p> <p>IaaS API</p> <p>Cloud Management Framework</p> <p>EGI Cloud Compute Service</p> <p>3. Use native Kubernetes tooling with EGI Check-in authentication</p>
Dependencies	<p>EGI Cloud Container Compute depends on EGI Cloud Compute and external open source tools:</p> <ul style="list-style-type: none"> - Kubernetes as a container orchestration platform - Ansible for the deployment and configuration of the Kubernetes clusters - Infrastructure Manager (IM) for the management of the VMs where the cluster is deployed.
Interfaces	<p>The service uses native Kubernetes API with OpenID Connect authentication as described in Kubernetes Authentication Strategies documentation.</p>
Data	<p>The service is data agnostic in relation to both the data content and the data format.</p>
Needed improvement	<p>Improvement areas:</p> <ul style="list-style-type: none"> - Better documentation of the deployment of the clusters - Management of upgrades of Kubernetes - Auto-scaling of the cluster as needed.

PaaS Orchestrator

Identification	PaaS Orchestrator
Type	Java Application
Purpose	The Orchestrator receives high-level deployment requests and coordinates the deployment process over the underlying IaaS sites. This component is built around a workflow engine: the implemented workflows include the interactions with other microservices of the PaaS layer as detailed below.
Function	<p>The Orchestrator manages the deployment implementing a complex workflow. The main steps are summarized hereafter:</p> <ul style="list-style-type: none"> • User authentication/authorization • User request validation and TOSCA template parsing • Retrieval of information about the IaaS sites, including SLAs and monitoring data • Service/Virtual infrastructure deployment exploiting IaaS automation tools like INDIGO Infrastructure Manager, Openstack Heat, or docker orchestration tool like Apache Mesos (Marathon and Chronos framework are supported) or other tools that can be plugged in as shown in the figure below. • Check deployment readiness <p>This high-level schema is implemented using BPM technology (workflow). The Orchestrator implements a set of workflows to manage the deployment lifecycle requests, such as create, update and delete.</p> <p>The user only has to submit a deployment request providing the description of the topology of the services/infrastructure using the TOSCA language. After the workflow starts, the user can poll the Orchestrator APIs to know the service deployment status.</p> <p>A user-friendly CLI tool is available for interacting with the Orchestrator service.</p>
High Level Architecture	The Orchestrator component is implemented using Java technologies and open-source frameworks such as: Flowable (the workflow engine), Alien4Cloud TOSCA library (used for the Orchestrator TOSCA parser), Hibernate ORM with MySQL database, etc.

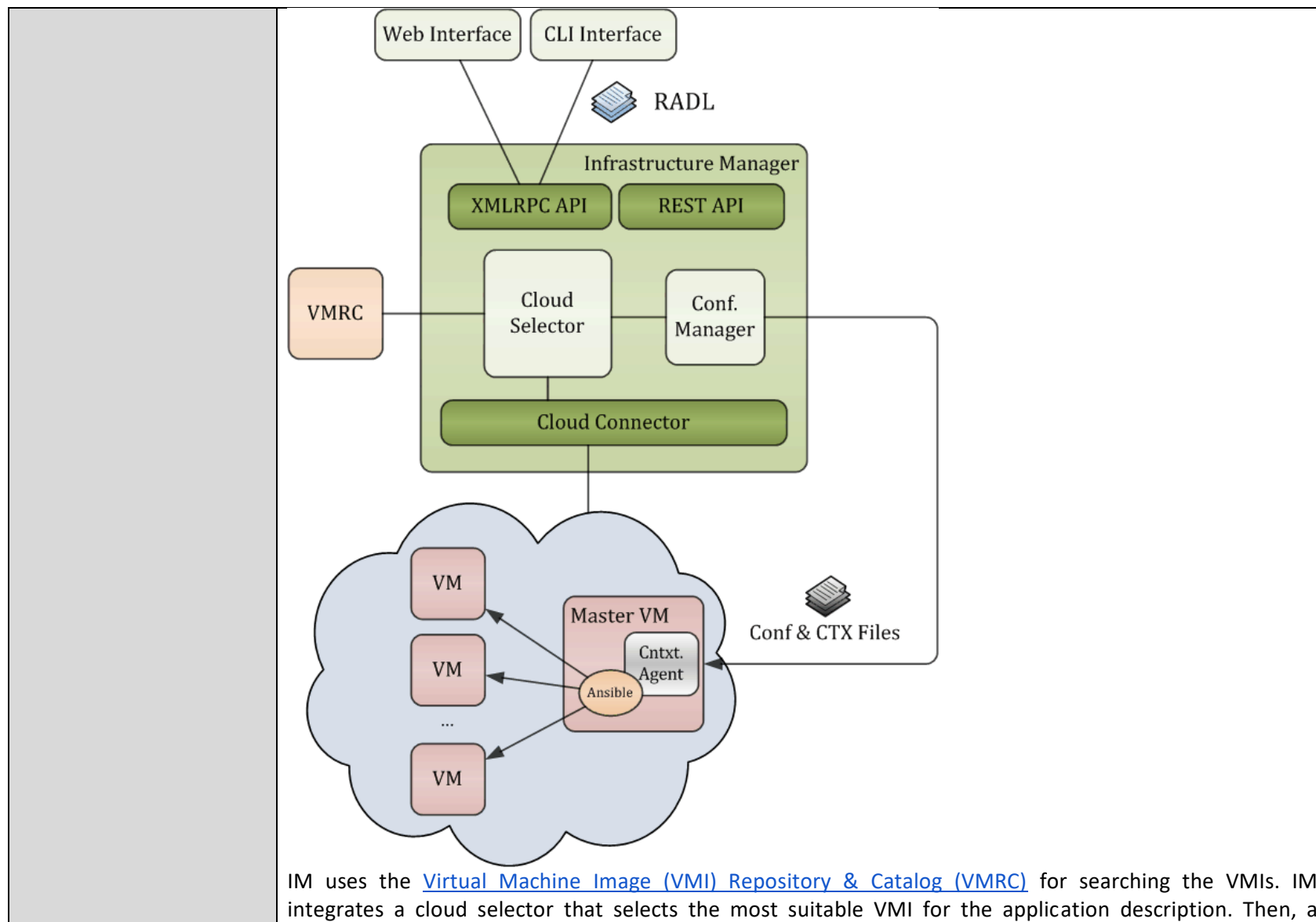
	 <p>After receiving a request of deployment, the Orchestrator interacts with different PaaS services in order to select the best IaaS site. It collects information about the user's SLAs, about the computing and storage services available at the sites, and monitoring data. If the deployment site and resources are available, the deploying process begins with IM/HEAT or a dedicated adapter.</p> <p>Information about the deployments is stored in a MySQL database.</p>
Dependencies	<p>The Orchestrator depends on the following services:</p> <ul style="list-style-type: none"> ● INDIGO IAM for user authentication; ● INDIGO SLAM for collecting information about the user's SLAs; ● INDIGO CMDB for collecting information about the IaaS sites (available virtual images, compute and storage services, etc.); ● INDIGO Monitoring service for collecting services health status and metrics; ● INDIGO Cloud Provider Ranker (CPR) for ranking the cloud providers; ● IaaS orchestrator like INDIGO IM, Openstack Heat for cloud resources provisioning and configuration; ● Apache Mesos (Marathon/Chronos) for deploying docker containers
Interfaces	The Orchestrator exposes RESTful Web Services.

	It complies with RFC 2616 standard and uses JSON data format as request and response encoding. All messages and error codes comply with standard status code definition.
Data	<p>Web Services message data are in JSON encoding.</p> <p>Deployments data, endpoints and configurations are stored in MySQL databases and are accessible through RESTful APIs. An additional MySQL schema stores the information of BPM processes.</p>
Needed improvement	<p>Integration with the EOSC-HUB AAI.</p> <p>Support for further EOSC-HUB data services. At the moment Onedata is supported.</p> <p>Potential changes/adaptation in the interfaces for getting SLAs info, monitoring data and configuration items from the cloud providers. This will depend on the changes that will affect the INDIGO components the Orchestrator relies on (SLAM, CMDB, Monitoring)</p>

Infrastructure Manager

Identification	Infrastructure Manager (IM)
Type	A service and a client.
Purpose	The IM is a service for the whole orchestration of virtual infrastructures and applications deployed on it, including resource provisioning, deployment, configuration, re-configuration and termination.
Function	<p>The service manages the complete deployment of virtual infrastructures or individual components within them.</p> <p>The status of a virtual infrastructure can be:</p> <p>pending: launched, but still in initialization stage; running: created successfully and running, but still in the configuration stage; configured: running and contextualized;</p>

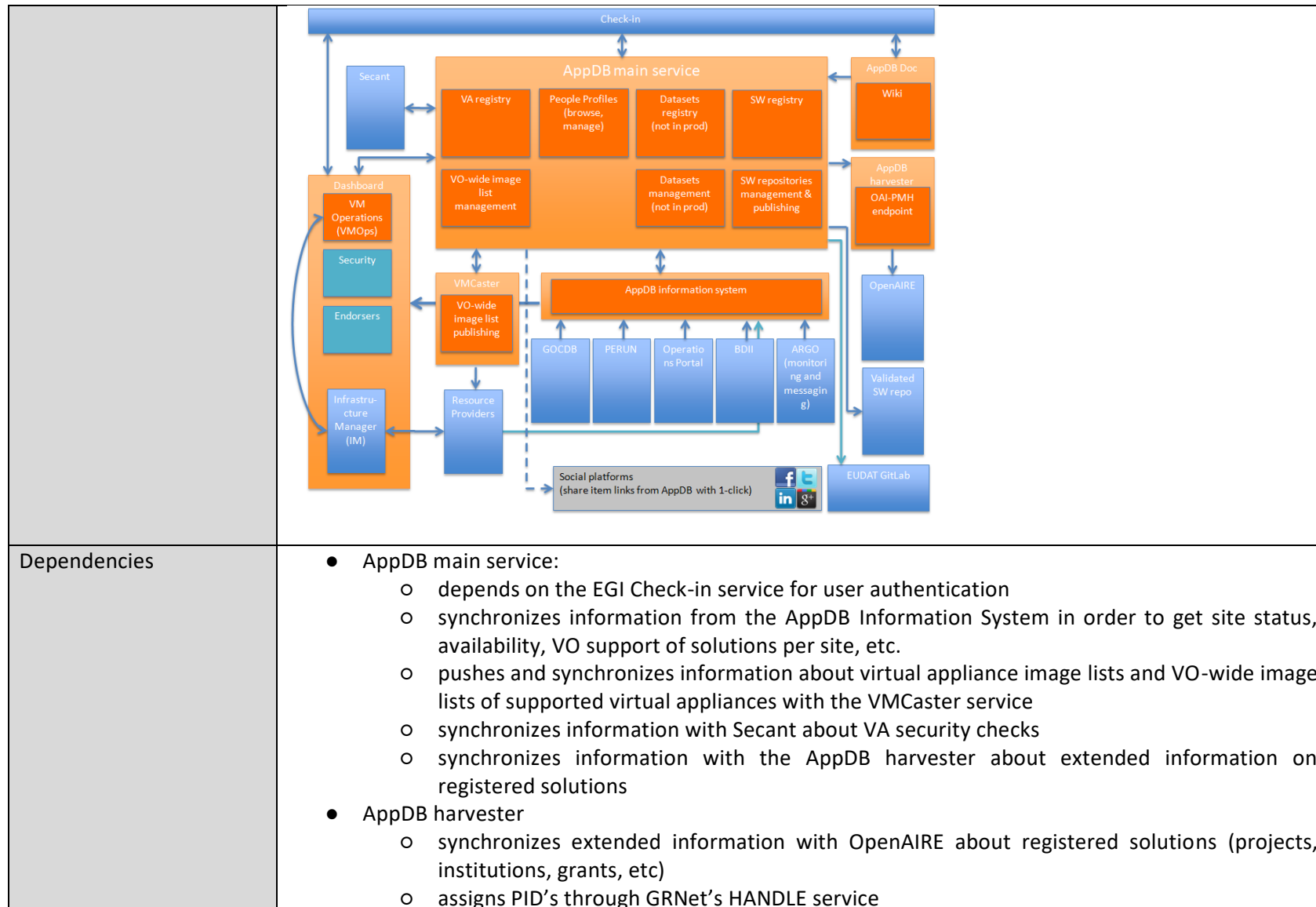
	unconfigured: running but not correctly contextualized; stopped: stopped or suspended; off: shutdown or removed from the infrastructure; failed: an error happened during submission; unknown: unable to obtain the status.
High Level Architecture	The following figure describes the high-level architecture of the IM, including external dependencies.



	configuration manager based on Ansible configures the VMs deployed by the cloud connector and installs the necessary software. The cloud connector provides the independence to the cloud IaaS platform.
Dependencies	<p>The IM service requires two additional components to work: IaaS cloud resources and a VMI repository.</p> <p>It supports multiple back ends and standards (Amazon EC2, Microsoft Azure, Google Cloud, OpenNebula, OpenStack, OCCl, Fogbow, Kubernetes, etc).</p> <p>IM uses VMRC as the VMI repository, although this can be skipped if the INDIGO PaaS Orchestrator is used to provide VMI selection.</p>
Interfaces	<p>The IM service supports two APIs:</p> <ul style="list-style-type: none"> ● The native one in XML-RPC ● A REST interface. <p>It also includes a command-line Python client which interacts with the XML-RPC API.</p>
Data	<p>The IM uses three types of information</p> <ul style="list-style-type: none"> ● Application descriptions following the OASIS TOSCA Simple Profile in YAML Version. ● Information about the cloud providers end-points and associated metadata to be used by the IM. ● Information about the deployed infrastructures (specifications, IDs, status, end-points, etc.) in a MySQL database or in a file. <p>Client and server exchange data through the parameters of the API calls.</p>
Needed improvement	Implement potential new EOSC-hub requirements/services (e.g. AppDB IS, EOSC-hub AAI, etc)

AppDB

Identification	Application Database https://appdb.egi.eu .
Type	Web service and portal
Purpose	The EGI Applications Database is a service that stores and provides information about software solutions in the form of native software products and virtual appliances, about the programmers and scientists involved, and about publications derived from the aforementioned solutions. Through its portal, AppDB enables users to search for and discover scientific software solutions that may meet their needs, across the breadth of the EGI/EOSC federated computing infrastructure, sparing them from having to span their search across multiple directories in order to find out where and how to use them. It also provides mechanisms for automatic distribution of Software items and Virtual appliances and through its VMOps Dashboard, it enables users to deploy and manage Virtual Machines on the EGI/EOSC Cloud infrastructure.
Function	<p>Metadata about the solutions registered in AppDB are correlated with information retrieved from the infrastructure about VOs, resource providers/sites, etc. by interfacing with multiple services, thus providing users with a complete overview of how and where they may use any solution that meets their need.</p> <p>In the case of native software solutions, AppDB offers release management functionalities of source and binary artifacts, by interfacing with the EGI Community Repository and allows the creation of RPM/DEB repositories for mass deployment activities on remote hosts and resource providers. As for the case of virtual appliances, AppDB allows users to deploy and manage solutions on the federated cloud infrastructure, through its VMops dashboard, which abstracts away any implementation-specific kinks of each provider.</p> <p>All of the data offered though the AppDB portal are also available through a RESTful API which may be used by registered users and external services in order to automate tasks and integrate.</p>
High Level Architecture	The following figure describes the high-level architecture of the EGI Applications Database, including external dependencies. Components in orange are internal to AppDB, while those in blue are external.



	<ul style="list-style-type: none"> ● AppDB Information System <ul style="list-style-type: none"> ○ aggregates site information from the GOCDB, BDII's, and Argo, and VO information from Perun and the EGI Operations portal ○ provides above aggregated information to the AppDB main service ● VMOps dashboard <ul style="list-style-type: none"> ○ receives VA information from the AppDB main service ○ synchronizes topology information with the IM external service
Interfaces	<ul style="list-style-type: none"> ● AppDB RESTful API <ul style="list-style-type: none"> ○ offers multiple resources which represent logical entities stored by AppDB ○ produces XML documents ○ may return any of the following error codes <ul style="list-style-type: none"> ■ RE_OK = 0 ■ RE_ACCESS_DENIED = 1 ■ RE_ITEM_NOT_FOUND = 2 ■ RE_INVALID_REPRESENTATION = 3 ■ RE_INVALID_METHOD = 4 ■ RE_INVALID_RESOURCE = 5 ■ RE_BACKEND_ERROR = 6 ■ RE_INVALID_OPERATION = 7 ○ used by the AppDB main service portal (r/w), the VMcaster service (r/w), and the VMOps dashboard (ro) ○ open to external services and users that may want to use it ● SAML interface <ul style="list-style-type: none"> ○ AuthN/AuthZ interface between EGI Check-in service and the AppDB main service ● OAI-PMH endpoint <ul style="list-style-type: none"> ○ provides XML documents in multiple schemata (Datacite, Dublin Core, etc) ○ used by OpenAIRE to pull data about registered solutions ● AppDB Information System web-API

	<ul style="list-style-type: none"> ○ a web-API which produces JSON documents, used by the AppDB main service ○ open to external services that may want to use it ● GOCDB RESTful API <ul style="list-style-type: none"> ○ external service RESTful API which the AppDB Information Service uses ● EGI Operations Portal / Perun web services <ul style="list-style-type: none"> ○ external web endpoints that serve XML data which the AppDB Information Service uses ● ARGO web-API <ul style="list-style-type: none"> ○ external service that produces JSON documents which the AppDB Information Service uses ● BDII LDAP <ul style="list-style-type: none"> ○ external LDAP service that provides information under the GLUE schema which the AppDB Information Service uses ○
Data	<ul style="list-style-type: none"> ● Data and metadata about the registered solutions in the AppDB main service are stored in a PostgreSQL ORDBMS ● The VMOps dashboard stores information in a combination of MongoDB and MySQL databases ● The AppDB Information System uses a CouchDB database to store aggregated data
Needed improvement	<ul style="list-style-type: none"> ● General <ul style="list-style-type: none"> ○ Development of the VM Image Security dashboard ○ Development of the Endorsers dashboard ● AppDB harvester <ul style="list-style-type: none"> ○ Enrich metadata with funding related information ● AppDB Information System <ul style="list-style-type: none"> ○ Support of Glue 2.1 schema ○ Fetch data directly from cloud resource providers (avoid BDII) ● VMOps dashboard <ul style="list-style-type: none"> ○ Support of complex topologies ○ Support of OIDC ○ Use of native CMF APIs instead of OCCI

Repositories

Identification	EGI Software repository http://repository.egi.eu/
Type	Service
Purpose	The EGI Software Repository Portal provides a unified point of access for the Middleware Distributions, the Community Repositories and the operational tools developed by EOSC. The software distributed by this service passes through a quality verification process that covers functionality tests under production environments. The repository is divided into two main categories: the Unified Middleware Distribution (UMD), which distributes traditional middleware, and the Cloud Middleware Distribution (CMD) which distributes specific middleware for OpenStack and OpenNebula integration components developed by Cloud Technology Providers and needed to run the EOSC federation. The CMD is composed of two different distributions: CMD-OS for OpenStack and CMD-ONE for OpenNebula.
Function	This component downloads, verifies and organizes into repositories, software as RPMs and DEBs bundles. It moves the submitted bundles (aka product releases) into the appropriate areas based on the verification outcome: Unverified -> Verified -> StageRollout -> UmdStore -> Production The hosted software is finally distributed to the public through a collection of YUM or APT compatible repositories.
High Level Architecture	<p>The service is composed of the following components:</p> <ul style="list-style-type: none"> ● Request tracker: acts as the user-interface for submitting and managing product releases ● Backend: downloads and processes the submitted software products ● Composer: constructs the UMD or CMD releases out of the submitted product releases ● Front-end: populates the produced UMD or CMD repositories ● Repository areas: implements the storage for the hosted software items and their respective repositories <p>The following figure describes the high-level architecture of the Repository. Components in orange are internal to the EGI Repository, while those in blue are external.</p>

	<pre> graph TD External[External/institutional repositories] --> EGI[EGI Repository] subgraph EGI_Repository [EGI Repository] RT[Request Tracker (RT)] --> Backend[Backend] Backend --> Frontend[Frontend] Backend --> RA[Repository areas] Composer[Composer] <--> RA end Verification[Verification Team (human)] --> RT </pre>
Dependencies	<p>This service depends on the Verification Team: responsible unit (human) for performing validation processes to the submitted software product releases</p>
Interfaces	<p>Below the internal & external interfaces are listed:</p> <p>Internal:</p> <ul style="list-style-type: none"> ● RT -> Backend: XML-RPC API ● Backend -> RT: Rest API ● Backend -> Frontend: Rest API combined with an XML-RSS Feed ● The rest of the internal components are communicating by using the shared database. <p>External:</p> <ul style="list-style-type: none"> ● External tools may submit releases using RT Rest API ● Users may download software from the Frontend using standard YUM or APT tools.
Data	<p>The metadata are stored in an internal MySQL database engine. The data (packages) are natively stored in the filesystem.</p>

Needed improvement	<p>We foresee the following improvements:</p> <ul style="list-style-type: none"> ● improve the front-end part of the service in terms of: <ul style="list-style-type: none"> ○ usability ○ searching ○ quality of information ● Increase the automatization level of the software quality assurance process ● support additional distribution methods, e.g. containers, virtual machines ● investigate the possibility of using the AppDB service as a front-end module of the EGI repository
--------------------	---

eduTEAMS

Identification	eduTEAMS https://www.geant.org/Innovation/eduteams
Type	A service operated by GEANT that builds on top of eduGAIN to support research collaborations to deploy their own AAI.
Purpose	The service offers a suite of tools aimed at research collaborations to manage their users, membership and access policies, to access a variety of services in a federated fashion.

Function	<p>Main functions are:</p> <ul style="list-style-type: none">● eduTEAMS Proxy & Identity Hub: support for the OIDC and SAML protocols. It can connect SAML Identity Providers, OIDC Providers, SAML Service Providers, OIDC Resource Providers enabling teams to use their preferred identity sources and services regardless of the authentication protocol used. The eduTEAMS Proxy is responsible for aggregating the user attributes from various identity sources, enforce community● eduTEAMS Discovery service provides a web interface for users to search and select their preferred identity provider. It is an essential component of the platform, directly connected with the eduTEAMS Proxy.● eduTEAMS Metadata Service (MDS) aggregates the metadata of all the SAML Identity and Service providers that are connected on the platform. It does so by aggregating the metadata feed of eduGAIN, while allowing the platform administrators to configure also other local or remote metadata sources.● eduTEAMS Membership Management Services (MMS) enables users to create virtual organisations (VO), 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. Users can choose between 3 options for their VO: CManage, HEXAA and Perun. All three are supported and available on the eduTEAMS platform.
----------	---

<p>High Level Architecture</p>	
<p>Dependencies</p>	<p>Satosa, Membership Management Services,</p>
<p>Interfaces</p>	<ul style="list-style-type: none"> ● SAML ● OAuth2 ● OIDC ● LDAP, SQL ● HTTP <p>Because eduTEAMS follows the AARC BPA and because of the support of SAML and OAuth2 protocols eduTEAMS can interoperate with B2Access and Check-in services.</p>

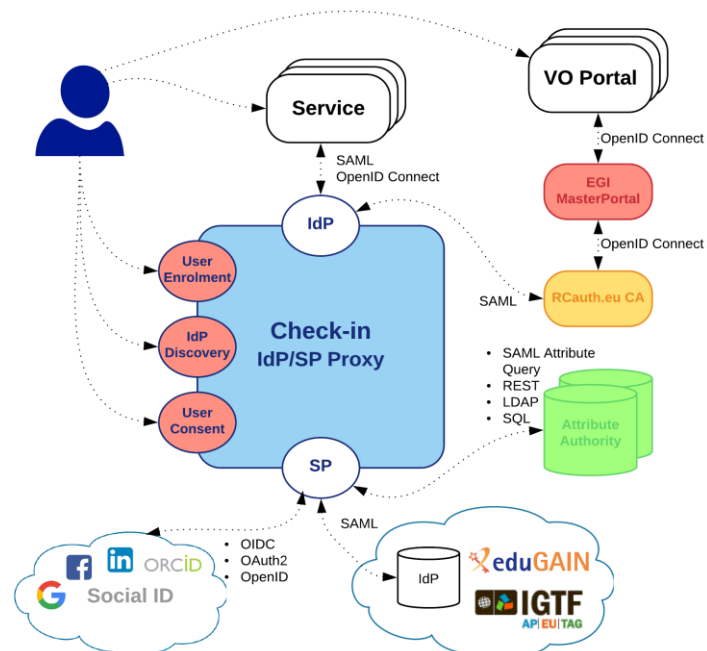
Data	Only authentication/authorisation assertions are supported in different format: username and password, X.509, SSH key, SAML token, OAuth2 and OIDC.
Needed improvement	Plans are to further enhance the UI, integrate a step-up service (for those require it) and in general continuously improve the service based on community feedback.

EGI Check-in

Identification	EGI Check-in
Type	The EGI Check-in service is an Identity and Access Management solution that makes it easy to secure access to services and resources.
Purpose	Through Check-in, users are able to authenticate with the credentials provided by the IdP of their Home Organisation (e.g. via eduGAIN), as well as using social identity providers, or other selected external identity providers. Check-in provides an intuitive interface for communities to manage their users and their respective groups, roles and access rights. For communities operating their own group management system, Check-in has a comprehensive list of connectors that allows integrating their systems as externally managed Attribute Authorities.
Function	<p>Main functions are:</p> <ul style="list-style-type: none"> ● IdP/SP Proxy acts as a Service Provider towards the external Identity Providers and, at the same time, as an Identity Provider towards the Service Providers (e.g. GGUS, AppDB, etc). Through the IdP/SP proxy, users are able to sign in with the credentials provided by the IdP of their university or research institute that participates in eduGAIN, as well as using social identity providers, or other selected external identity providers, such as Google, Facebook, LinkedIn, and ORCID. To achieve this, the proxy supports different authentication and authorisation standards, such as SAML 2.0, OpenID Connect (OIDC) 1.0 and OAuth 2.0. ● IdP Discovery Service (Where Are You From – WAYF) allows for users to select their preferred IdP. ● User Enrolment and Group/VO Management supports the management of the full life cycle of user accounts in Check-in. This includes the initial user registration, the acceptance of the terms of use of the infrastructure, account linking, group and VO management, delegation of administration of VOs/Groups to authorised users and the configuration of custom enrolment flows for VOs/Groups via an intuitive web interface. For VOs,

operating their own Group/VO Management system, the Check-in service has a comprehensive list of connectors that allows integrating their systems as externally managed Attribute Authorities.

High Level Architecture



Dependencies

The IdP/SP Proxy is based on SimpleSAMLphp and MITREid Connect. The User Enrolment and Group/VO Management component is based on COnamange and Perun for selected communities.

Interfaces

- SAML2
- OAuth2
- OIDC
- LDAP

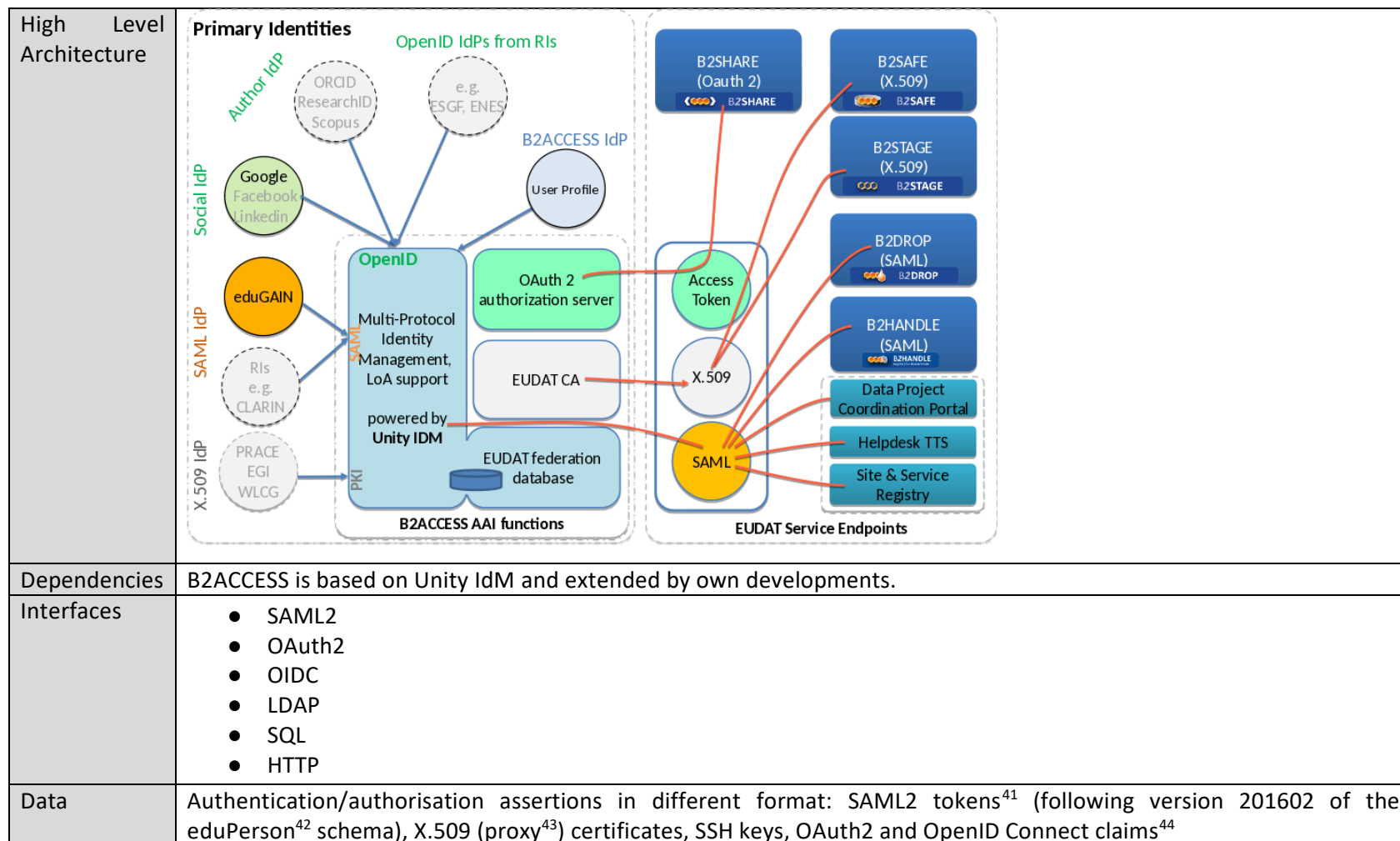
	<ul style="list-style-type: none"> • SQL • HTTP <p>Since Check-in follows the AARC BPA and because of the support of SAML and OAuth2 protocols Check-in can interoperate with B2Access and eduTEAMS services.</p>
Data	For the data internal to the component, this field describes the representation method, initial values, use, semantics, and format.
Needed improvement	The production operation of the EGI Check-in service involves technological upgrades of the underlying framework and libraries in order to take advantage of new features and robustness, as well as continuous optimisation of the architecture and automation of new tasks to ensure the uninterrupted and performant operation.

B2ACCESS

Identification	B2ACCESS
Type	B2ACCESS is an authentication and user management platform for EUDAT CDI.
Purpose	B2ACCESS follows the Identity Management as a Service (IDaaS) approach by externalising the user authentication from protected services. It allows users to access (Web or non-Web Browser based) infrastructure services while authenticating with their home organisation provided credentials. Since B2ACCESS supports multiple authentication protocols (for example, X.509, OAuth 2.0/OIDC, SAML, LDAP and PAM), several types of credentials can be used.
Function	<p>B2ACCESS specifically offers the following main functions:</p> <ul style="list-style-type: none"> • IdP/SP proxy component of B2ACCESS acts as an intermediary and significantly reduces the complexity of trust and authentication management between the two entities (IdPs and SPs). Alongside simplified trust management, B2ACCESS plays an important role in connecting the IdPs and SPs that may not rely on similar authentication protocols (for example, SAML based IdP and OIDC based SP). In that case, B2ACCESS performs user authentication from its IdP in a protocol agnostic fashion and transparently generates the types of credentials based on the SP requirements (for example, access token for the OIDC SP or SAML assertion for the SAML SP). • User discovery sub-system allows the user to select the preferred identity provider, for the external authentication. • User and group management is one of the core functions of B2ACCESS and is managed through its versatile

administrator Web GUI. It allows the B2ACCESS administrators to manage the groups and the users therein. The user provisioning can be automatic (through user registration form) or manual (under the administrator Web user interface) by dragging the users from one group to the other. The groups are hierarchically organised, and the privileges assigned to the users are inherited from parent groups. In addition, each group (and its users) are managed independently by its administrators. Furthermore, B2ACCESS uses MVFLEX Expression Language (MVFL) to let administrators define specific rules for the groups. This is an advanced feature and has been very useful in creating dynamic user attributes in (sub-)groups.

- **Form management sub-system** enables B2ACCESS administrators to create user registration forms for the new users. The form is invoked upon successful user authentication. It is very common that an infrastructure relying on B2ACCESS changes its policy and requires its 'existing' users to provide new attribute(s); in that case B2ACCESS provides enquiry forms to extract that additional information from the users. There are also invitation forms (not publicly available) to register a specific set of users.
- **Self-service user homepage** (separate from administrator Web GUI) allows users to manage their profile containing all their information/attributes, and to update their credentials under the home page.
- **User Import Management (UIM)** used to import user attributes from external attribute providers/authorities in a configurable manner. Currently SAML, LDAP and OIDC based external attribute providers are supported.



⁴¹ <https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf> (see Section 2.3.3)

⁴² <http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201602.html>

⁴³ <https://tools.ietf.org/html/rfc3820>

Needed improvement	The production operation of the B2ACCESS service involves technological upgrades of the underlying framework and libraries. With this updates new features enhance the usability and robustness of the service. Keeping underling libraries up to date will close potential security issues too.
--------------------	--

Perun

Identification	Perun
Type	Perun is an identity and access management system (IAM) designed to build strong identity and authorisation layers on top of an existing infrastructures.
Purpose	Perun offers complete support for VO management and whole user life-cycle form enrolment to suspension/leaving the VO. Users and groups can be managed directly in the tool by the VO manager or by users themselves as a part of registration flow. Alternatively, the users can be synchronised from existing external source like LDAP, VOMS or SQL database. Perun manages access to the services by supporting provisioning and deprovisioning of authorisation data. It can manage access to the web based and also non-web based services.
Function	<p>Main functions are:</p> <ul style="list-style-type: none"> ● Virtual organization management allows the creation of virtual organisations and the assignment of VO managers ● User management allows: <ul style="list-style-type: none"> ○ creating customisable registration forms an enrolment flows for virtual organisations ○ manually or automatically approving user's applications ○ setting up various membership expiration ○ assigning additional information to the user entry in Perun ○ user and administrator notifications ● Group management allows: <ul style="list-style-type: none"> ○ organising users into groups ○ users to apply for group membership ○ delegating rights to manage group members

⁴⁴ https://openid.net/specs/openid-connect-core-1_0.html#Claims

	<ul style="list-style-type: none"> ○ synchronising members between groups ○ customized expiration of group membership ● Resource management allows: <ul style="list-style-type: none"> ○ allocating resources to virtual organizations/projects ○ setting up various access rights or other specific configuration for each organization/group ○ setting up resource for real users or service identities ● Service management enables: <ul style="list-style-type: none"> ○ configuring services directly on resources ○ monitoring status of the configuration propagation ○ sending the new configuration every time some change occurs - provisioning/deprovisioning ○ using existing infrastructure, like LDAP, together with Perun
High Level Architecture	<p>The diagram illustrates the High Level Architecture of Perun, organized into three main functional areas:</p> <ul style="list-style-type: none"> Propagation: This section handles the distribution of configuration. It includes a SLAVE component that receives data from a SEND component. The SEND component is connected to GEN (Generator), which in turn feeds into the Engine. The Engine is connected to the Dispatcher. RPC (Remote Procedure Call): This central section manages the core logic. It features a Registrar that interacts with the Auditer. The Auditer is connected to the Core and the Controller. The Core is also connected to the Notification and Cabinet components. The Controller and Notification components are interconnected. Interfaces: This section provides the user-facing and system-level interfaces. It includes Java lib, Perl lib, CLI, JavaScript lib, WEB GUI, PHP lib, and Java RPC lib. <p>External components like LDAPc are shown connected to the Core in the RPC section.</p>
Dependencies	<p>Perun is based on enterprise technologies:</p> <ul style="list-style-type: none"> ● Java

	<ul style="list-style-type: none"> ● Spring - Supports building flexible JVM-based systems and applications ● Google Web Toolkit - Used to build web user interface of Perun ● Jenkins - Leading open-source continuous integration software <p>To use SQL database as a backend, Oracle and PostgreSQL database engines are supported. Perun can synchronise groups and members with external sources like:</p> <ul style="list-style-type: none"> ● XML and CSV files ● any SQL DB supporting JDBC ● VOMS ● LDAP servers, including Active Directory <p>The Source code is available on GitHub⁴⁵</p>
Interfaces	<p>Perun provides the following interfaces:</p> <ul style="list-style-type: none"> ● RPC using JSON (or VOOT) ● Java library ● PERL library ● PHP library ● command line interface (Perun CLI⁴⁶) ● LDAP
Data	<p>Perun maintains information about the managed virtual organisations, groups, users, resources and services in the form of Perun attributes⁴⁷.</p>
Needed improvement	<p>Main points for further development:</p> <ul style="list-style-type: none"> ● Extend number of attributes available through LDAP interface, which serves as common integration point with IdP/SP proxies. ● Add support for more complex life-cycles of users within the VOs. ● Expand user documentation.

⁴⁵ <https://github.com/CESNET/perun>

⁴⁶ https://wiki.metacentrum.cz/wiki/Perun_CLI

⁴⁷ <https://wiki.metacentrum.cz/wiki/Attributes>

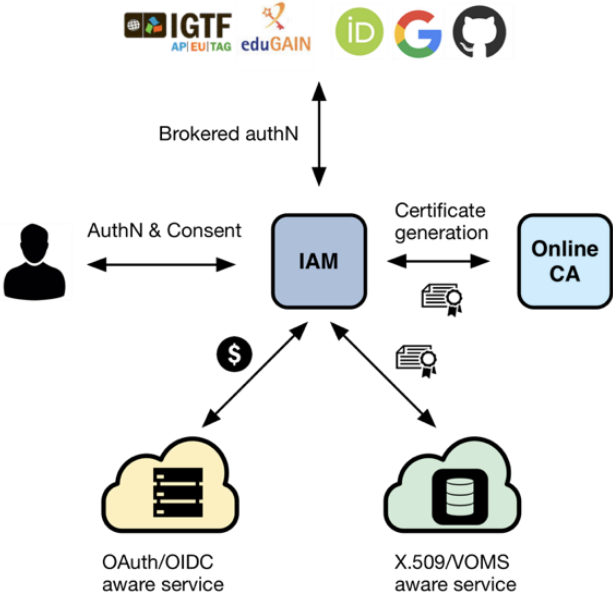
WaTTS

Identification	WaTTS
Type	WaTTS (the INDIGO-DataCloud Token Translation Service) is a plugin-based Token Translation Service
Purpose	WaTTS accepts federated user identities (via OpenID Connect) and uses a plugin scheme to generate credentials for allowing users to access services that do not support federated identities.
Function	<p>Existing translation plugins are available for:</p> <ul style="list-style-type: none"> ● SSH ● S3 storage (commercial DDN appliance) ● OpenNebula cloud middleware ● X.509 online certificate authorities (CAs) <p>The primary function of WaTTS within EOSC-hub is the translation of federated identities into X.509 certificates. To this end, WaTTS acts as an implementation of a so-called MasterPortal for the RCauth Online CA. A user can request a certificate after authenticating into WaTTS using any OpenID Connect Provider. For this, the user is redirected to the RCauth CA WAYF. There, he/she may choose which home IdP to use for authenticating to RCauth. Upon successful authentication at RCauth, the user's browser is redirected back to WaTTS, where he/she is given a proxy of his certificate. The full certificate is stored inside a myProxy secure credential store.</p>

	<pre> family_name no yes yes yes given_name no yes yes yes name yes yes yes yes eduPersonEntitlement yes no no no groups yes yes no no email yes no no no email_verified yes no no no organisation_name no yes no no picture no no no user-provided preferred_username no no no no profile no no yes updated_at no yes no no </pre> <p>This data received by WaTTS is passed on to the plugin chosen by the user. WaTTS stores no data beyond the lifetime of a session (few minutes). It is up to the plugin to store data. For stored credentials (e.g. SSH-keys X.509 certificates) the plugins stores the key material plus a unique identifier (sub, iss). The unique identifier is required to return the stored credential to the user.</p>
Needed improvement	WaTTS will be extended to support high availability deployment.

INDIGO IAM

Identification	INDIGO IAM
Type	The Identity and Access Management Service provides a layer where identities, enrolment, group membership and other attributes and authorization policies on distributed resources can be managed in a homogeneous way.
Purpose	Simplify authentication and authorization management for research communities
Function	<p>Identity hub:</p> <ul style="list-style-type: none"> • A central point of authentication supporting Identity Federations (EduGAIN), X.509 certificates and social logins (e.g., Google) • User-driven account linking functionality <p>Collaboration management:</p>

	<ul style="list-style-type: none"> • Enrolment and registration functionalities, so that users can join groups/collaborations according to consolidated flows • Group membership and attribute assignment management, to provide group/collaboration administrators the ability to manage <p>Provisioning:</p> <ul style="list-style-type: none"> • Standard identity provisioning interfaces to expose user and group membership information to relying services <p>Authorization</p> <ul style="list-style-type: none"> • OAuth Scope and group-based authorization together with the integration with a XACML-based engine that allows to define and enforce authorization policies on distributed resources
High Level Architecture	 <p>The diagram illustrates the High Level Architecture of the IAM (Identity and Access Management) service. At the top, several logos are displayed: IGTF (APIEUITAG), eduGAIN, ID, G, and GitHub. The central component is the IAM service, represented by a blue box. It interacts with several other components:</p> <ul style="list-style-type: none"> Brokered authN: A vertical double-headed arrow connects the IAM service to the logos at the top. AuthN & Consent: A horizontal double-headed arrow connects the IAM service to a user icon on the left. Certificate generation: A horizontal double-headed arrow connects the IAM service to an Online CA (Certificate Authority) on the right. OAuth/OIDC aware service: A diagonal arrow with a dollar sign icon points from the IAM service to a cloud icon representing an OAuth/OIDC aware service. X.509/VOMS aware service: A diagonal arrow with a certificate icon points from the IAM service to a cloud icon representing an X.509/VOMS aware service.
Dependencies	IAM is a Java 8 Spring Boot-based service which leverages the MitreID Connect OpenID Connect server implementation library.

Interfaces	<ul style="list-style-type: none"> ● OAuth2 ● OpenID Connect ● SAML/XACML ● SCIM ● Other ReSTful HTTP APIs
Data	IAM keeps its state in a MySQL/MariaDB database instance. Authentication and authorization information are exposed via OAuth, OpenID Connect, SCIM and SAML/XACML interfaces
Needed improvement	Foreseen developments aim at providing improved support for LoA and multitenancy.

EGI Online storage

Identification	EGI Online Storage- https://www.egi.eu/services/online-storage/
Type	EGI Online Storage is a service allowing to effectively and reliably store and share data in a distributed environment.
Purpose	Online Storage allows researchers to store data in a reliable and high-quality environment and share it across distributed teams. Data can be accessed through different standard protocols and can be replicated across different providers to increase fault-tolerance. Online Storage gives users complete control over the data they share.
Function	<p>Main characteristics:</p> <ul style="list-style-type: none"> ● Assign global identifiers to files ● Access highly-scalable storage from anywhere ● Control the data that is shared ● Organise data using a flexible hierarchical structure
High Level Architecture	<p>There is no centralized endpoint, and the different tools provide:</p> <ul style="list-style-type: none"> ● Object storage ● File storage ● Block storage
Dependencies	<ul style="list-style-type: none"> ● EGI Check-In, X509, OIDC

	<ul style="list-style-type: none"> ● EGI DataHub ● Grid SE, BDII, VOMS ● OpenStack swift
Interfaces	<ul style="list-style-type: none"> ● GridFTP ● SRM ● HTTP/Webdav ● POSIX ● HTTP REST API
Data	N/A
Needed improvement	<ul style="list-style-type: none"> ● Exhaustive integration with CheckIn ● Improved integration with native APIs of cloud services ● Auto discovery of cloud sites' capabilities ● Expansion to commercial providers ● Integration with B2* suite

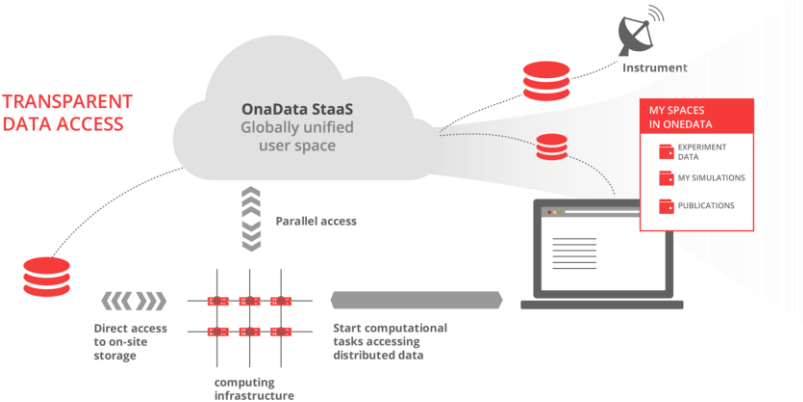
EGI DataHub

Identification	EGI DataHub - https://datahub.egi.eu
Type	EGI DataHub is a platform allowing to access (and share) public or private datasets from compute services.
Purpose	<ul style="list-style-type: none"> ● Discovery of data via a central portal. ● Access to data conforming to required policies which may be: 1) unauthenticated open access; 2) access after user registration or 3) access restricted to members of a Virtual Organization (VO. This access may be via a GUI (e.g. a webpage) or an API (e.g. programmatic access to the data) ● Replication of data from data providers for resiliency and availability purposes. Replication may take place either on--demand or automatically. ● Authentication and Authorization Infrastructure (AAI) integration between the EGI DataHub and with other EGI components and with user communities existing infrastructure ● File catalogue to track replication of data: logical and physical file
Function	<p>The EGI DataHub provides the following functionalities:</p> <ul style="list-style-type: none"> ● Federated authentication ● Federation of distributed providers

	<ul style="list-style-type: none"> ● Access to open datasets, Open data management ● Data management (sharing, access, discovery, replication)
High Level Architecture	<ul style="list-style-type: none"> ● EGI DataHub is a Onezone: https://datahub.egi.eu ● A default Oneprovider is available: https://plg-cyfronet-01.datahub.egi.eu ● User communities can connect their Oneproviders, deployed on different service providers, to the EGI DataHub Onezone
Dependencies	<ul style="list-style-type: none"> ● Onedata connected to site-specific storage ● EGI Online Storage (optional) ● EGI CheckIn
Interfaces	<ul style="list-style-type: none"> ● Web interface ● HTTP REST API ● POSIX ● CMDI
Data	N/A
Needed improvement	<ul style="list-style-type: none"> ● Supporting requirements from the user communities (see Onedata section) ● Finalizing tasks to operate and offer a production-grade service

Onedata

Identification	Onedata (https://onedata.org)
Type	Distributed virtual filesystem and data management platform
Purpose	In large scale hybrid cloud deployments it is often the case that data maintained in private clouds has to be processed on-demand in public clouds. While submitting remote jobs is today fairly straightforward, and can be automated using several orchestration platforms, making data available for processing in remote clouds is a significant challenge, which Onedata aims at addressing.
Function	<p>Onedata provides the following functionalities:</p> <ul style="list-style-type: none"> ● data access ● data discovery ● data federation ● data transfer (replication)

<p>High Level Architecture</p>	<ul style="list-style-type: none"> ● open data management <p>The Onedata architecture is based on the idea of unifying globally distributed storage resources and data sets, visible in the same way across all supported interfaces.</p>  <p>The diagram illustrates the Onedata architecture. At the top, a cloud labeled 'OnaData StaaS Globally unified user space' is connected to an 'Instrument' (satellite) and a laptop. A box titled 'MY SPACES IN ONEDATA' lists 'EXPERIMENT DATA', 'MY SIMULATIONS', and 'PUBLICATIONS'. Below the cloud, 'Parallel access' is shown with multiple arrows pointing to a 'computing infrastructure' block. To the left, 'Direct access to on-site storage' is shown with arrows pointing to a storage icon. Below the computing infrastructure, an arrow points to the text 'Start computational tasks accessing distributed data'.</p> <p>The main functional components include:</p> <ul style="list-style-type: none"> ● Onezone - the federation and authentication service, each Onezone instance (e.g. EGI DataHub) provides a single-sign on to a network of connected storage providers ● Oneprovider - is the main data management component of Onedata, which is deployed in the data centers and is responsible for provisioning the data and managing transfers ● Oneclient - provides the access to the virtual filesystem on a VM or host directly via a Fuse mountpoint
<p>Dependencies</p>	<p>Onedata itself does not provide any storage, which has to added to Oneprovider component by means of one of supported storage technologies:</p> <ul style="list-style-type: none"> ● POSIX (including NFS) ● Ceph ● S3 ● OpenStack Swift ● GlusterFS
<p>Interfaces</p>	<p>The following interfaces are provided by Onedata:</p> <ul style="list-style-type: none"> ● HTTP Rest API (https://onedata.org/#/home/api/latest)

	<ul style="list-style-type: none">• CDMI• POSIX
Data	N/A
Needed improvement	The main advancements within the framework of EOSC-hub project are envisioned in the area of supporting GridFTP storages, and integrating with other EUDAT services such as B2FIND and B2STAGE.