



EOSC-hub

D10.4 EOSC Hub Technical Architecture and standards roadmap v2

Lead Partner:	INFN
Version:	1
Status:	FINAL
Dissemination Level:	Public
Document Link:	https://documents.egi.eu/document/3495

Deliverable Abstract

This document describes the EOSC-hub contribution to the definition of the EOSC Technical Architecture, which is currently being developed by the EOSC architecture Working Group. It is based on the concepts of service interoperability and end-to-end composition of services and foresees the definition of a reference architecture in which EOSC building blocks and the main functions, interfaces, APIs and standards are identified. This architecture is expected to facilitate access to services, lower the barriers to integrate and composes services and promote the usage of services between adjacent communities.

As a basis for the proposed architecture, service categories have been introduced, mapping their functions, relationships and organisation to the kinds of services required for the federating core of EOSC and the external EOSC service portfolio. The concept of the end-to-end composition of services has been presented,



highlighting the most common integration scenarios. Leveraging the defined service categories and on the concepts of service interoperability and composition, a reference EOSC Technical Architecture has been defined identifying a hierarchical structure where the first level relies on service categories (Federation & Access enabling,, Common and Thematic), the second level introduces functional categories, that groups technical functions to facilitate their identification, and the third is made of the technical functions that has been called building blocks.

EOSC-hub is working on defining the building blocks of the architecture for each service type and specified a common approach to complete this task. It foresees the identification of the main building blocks/technical functions in each service category and, for each of those, the definition of a technical specification that includes a high-level architecture, suggested EOSC standards and APIs and interoperability guidelines. As a consequence, interoperability between services compliant with the EOSC specifications will be easier to be achieved.

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>Name</i>	<i>Partner/Activity</i>	<i>Date</i>
From:	Giacinto Donvito, Diego Scardaci and Mark Van De Sanden	INFN/WP10, EGI.eu/WP10 & SurfSARA/WP10	01/08/2019
Moderated by:	Malgorzata Krakowian	EGI.eu/WP1	
Reviewed by	Daan Broeder Baptiste Grenier	KNAW EGI.eu	15/09/2019
Approved by:	AMB		

DOCUMENT LOG

<i>Issue</i>	<i>Date</i>	<i>Comment</i>	<i>Author</i>
v.1	01/08/2019	Full draft	Diego Scardaci (EGI.eu), Giacinto Donvito (INFN) and Mark Van De Sanden (SurfSARA) Lukas Dutka (Cyfronet) Giuseppe Fiameni (CINECA) Heinrich Widmann (DRKZ) Ignacio Blanquer (UPV) Enol Fernandez (EGU.eu) Joao Pina (LIP) Marica Antonacci (INFN) Marcin Plociennik (PSNC) Jens Jensen (STFC) Michal Prochazka (CESNET) Licia Florio (GEANT)
v.2	20/08/2019	New version answering to reviewers' comments	Diego Scardaci (EGI.eu), Giacinto Donvito (INFN) and Mark Van De Sanden (SurfSARA)
v.3	27/08/19	Proofing Section 5 abstract exec summary, intro.	Owen Appleton (EGI.eu)

TERMINOLOGY

<https://wiki.eosc-hub.eu/display/EOSC/EOSC-hub+Glossary>

<i>Terminology/Acronym</i>	<i>Definition</i>
Access Enabling services	Delivering features allowing customers to easily exploit EOSC resources such as discovery, ordering and workflow enabling services (e.g. the EOSC Portal).
AAI	Authentication and Authorisation infrastructure
Building block	Technical functions that can be offered by one or more services. A building block is defined through a technical specification that includes an high-level architecture, suggested EOSC standards and APIs and interoperability guidelines
Common services	Providing generic capabilities usable by any science discipline each supporting aspects of the data lifecycle from creation to processing, analysis, preservation, access and reuse. Examples of services belonging to this category are multi-disciplinary services for data discovery, processing, workflow management and orchestration, data management, etc.
CMDB	Configuration Management Database
Federation services	Needed to operate the EOSC (e.g. a common helpdesk, accounting information gathering, monitoring)
HPC	High Parallel Computing
HTC	High Throughput Computing
IaaS	Infrastructure as a Service
Interoperability	Ability of two or more services to work together to deliver a feature for users.
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
PaaS	Platform as a Service
Reference architecture	In the field of software architecture or enterprise architecture, reference architecture provides a template solution for architecture for a particular domain. It provides a common vocabulary with which to discuss implementations, often with the aim of stressing commonalities.
Service composability	Ability to compose services to create new workflow.
Thematic services	Community-specific capabilities including research core data, data

	products, scientific software, and pipelines. Examples of thematic services are: data resources and software tools to access study and compare the data; data brokering services tailored to the needs of specific scientific communities.
UR	Accounting Usage Record
VM	Virtual Machine

Contents

1 Introduction	10
2 Landscape	12
2.1 EC Implementation Roadmap	12
2.2 EOSC Architecture Working Group	13
2.3 Past work on the EOSC Technical Architecture	14
2.3.1 EOSC Pilot Service Architecture	14
2.3.2 EOSC-hub Technical Architecture v1	15
3 The EOSC Portfolios and the EOSC Federating Core	17
4 Service Composability	19
4.1 Fostering the service interoperability	19
4.2 Federating thematic services in the EOSC	21
5 Defining the EOSC Technical Architecture	22
5.1 Reference Architecture	22
5.1.1 Approach to define building blocks	23
5.1.2 Technical Specification template	25
5.2 Proposed EOSC Technical Architecture	25
5.2.1 EOSC Access Enabling and Federation services	26
5.2.2 EOSC Common services	31
5.2.3 EOSC Thematic services	34
5.2.4 EOSC Portal	35
5.2.5 Architecture governance	37
6 Use cases to drive the identification and the specification of the building blocks	38
7 Relationship with the EOSC Architecture Working Group	41
8 Technical specifications for Federation Services	42
8.1 AAI	42
8.2 Accounting	47
9 Technical specifications for Common Services	49
9.1 Cloud IaaS VM Management	49
9.2 Platform as a Service (PaaS) solution	52
9.3 Metadata cataloguing and indexing	55
10 Conclusions and next steps	59

Executive summary

This document describes the EOSC-hub contribution to the definition of the EOSC Technical Architecture which is currently being developed by the EOSC architecture Working Group. It is based on the concepts of service interoperability and end-to-end composition of services and foresees the definition of a reference architecture where all the EOSC main functions, interfaces, APIs and standards are identified. This reference architecture will increase the added value provided by EOSC and foster its uptake, facilitating access to services, lowering barriers to integrate and composes services and promoting the usage of services between adjacent communities.

This work has taken into account the surrounding landscape and has followed the recommendations on the EOSC architecture of the European Commission described in the Staff Working Document on the *Implementation roadmap for the European Open Science Cloud (EOSC)*¹ and the mandate of the EOSC Architecture Working Group (WG)², recently launched by the EOSC governance. This paved the way for a refinement and better focusing of the scope of the EOSC architecture work within EOSC-hub. EOSC-hub is member of the EOSC Architecture Working Group and will contribute to the discussion within this WG. This deliverable must be seen in the light of the EOSC-hub contribution to the EOSC Architecture WG. Past work on this topic has been also analysed, notably the EOSC Service Architecture proposed by the EOSCpilot project³.

The proposed architecture is organised according to service categories: Federation & Access enabling, Common and Thematic services. As a basis to describe the architecture, service categories have been introduced, mapping their functions, relationships and organisation to the kinds of services required for the federating core of EOSC and the external EOSC service portfolio. The concept of end-to-end service composition has been presented, highlighting the most common integration scenarios and how services belonging to different categories can cooperate to create added-value solutions for research. EOSC-hub effort to foster service interoperability and the impact of the service composability on federating thematic services into the EOSC has also been depicted.

Leveraging the service categories and on the concepts of service interoperability and composition, a proposal for a reference Technical Architecture for EOSC has been defined identifying a hierarchical structure. The first level of this hierarchy relies on the subdivision in categories and allows to differentiate services according to their function within EOSC: Federation and Access enabling are key services to operate the EOSC (e.g. the EOSC Portal or the accounting infrastructure), Common services offer add-value features on top of EOSC resources (computing, storage, data, etc) and can be reused by a multitude of other services, Thematic services implement discipline specific features and are provided directly by scientific communities. The

¹ Commission Staff Working Document - Implementation roadmap for the European Open Science Cloud: https://ec.europa.eu/research/openscience/pdf/swd_2018_83_f1_staff_working_paper_en.pdf

² <https://www.eoscsecretariat.eu/working-groups/architecture-working-group>

³ <https://www.eoscpilot.eu/>

second level of the hierarchy introduces the functional categories that groups technical functions to facilitate their identification (e.g. Authentication and Authorisation or Monitoring for federation services, Cloud Compute and Metadata management for common services). In the case of thematic services, the functional categories are identified per scientific discipline. The third level is made of the technical functions that have been called building blocks. Examples of building blocks are AAI and accounting infrastructure for federation services, Cloud Infrastructure as a Service (IaaS) Virtual Machine (VM) management, a Platform as a Service (PaaS) solution or a Data Repository for common services, scientific workflows for thematic services.

EOSC-hub is working on defining the building blocks of the architecture for each service type and specified a common approach to complete this task. It foresees the identification of the main **building blocks/technical functions** in each service category. As described above, the typology of the building blocks changes according to the category they belong to. Then, for each of those building blocks, a **technical specification** that includes a **high-level architecture, suggested EOSC standards and APIs and interoperability guidelines** will be defined. As a consequence, thanks to the provided guidelines, interoperability between services offering the same technical function(s) and following the EOSC specifications will be easier to achieve, examples are the Authentication and Authorisation Infrastructure (AAI) services compliant with the AARC blueprint architecture and guidelines or monitoring and/or accounting systems able to exchange/share information and provide integrated views to the EOSC customers and service providers. This approach is tailored to the varied environment seen in EOSC, where many solutions to satisfy a given technical requirement already exist. Furthermore, having well defined EOSC endorsed standards and APIs and related interoperability guidelines for each of the identified building blocks will foster the end-to-end composition of services, lowering the barriers to make services interoperable. Indeed, other building blocks/services offering different technical functions can interoperate thanks to the EOSC interfaces, described in the technical specification (e.g. it would be easier for a thematic services integrating a common services if clear interoperability guidelines are available).

The EOSC interoperability guidelines that are being defined in the context of this work will take into account and will be based on existing community practices, well-known standards and interfaces. They should be defined by all relevant EOSC stakeholders (communities, e-infrastructures, etc.) in a collaborative manner and their adoption should not be mandatory but a natural consequence of the advantages, for a service, generated by being compliant.

In the proposed architecture, identifying building blocks and the related technical specifications for all the service categories has proved to be complex and long work; therefore we decided to follow an iterative approach starting from the functions that are more requested by the EOSC use cases⁴. Also the technical specifications, initially prepared by the technical experts within the EOSC-hub project, will be iteratively improved collecting feedback from external people with expertise in the area and involving them in the maintenance and evolution of such specifications.

⁴ EOSC-hub is taking into account in this work requirements collected from EOSC Pilot Scientific Demonstrator (see [D5.6 Evaluation Report of service pilots](#)), EOSC-hub Thematic Services (see [D7.2 First report on Thematic Service architecture and software integration](#)), EOSC Competence Centers (see [D8.1 Report on progress, achievements and plans of the Competence Centres](#)) and EOSC use cases identified through the EOSC Portal (see the [EOSC-hub Community Requirements Database](#)).

EOSC-hub already identified a considerable number of building blocks per service category and completed the technical specifications of the most relevant. However, we consider fundamental involvement of other relevant stakeholders in this work to have a real impact on the research world. For example, we think that including other technical experts in refining technical specifications and finding consensus around them to be essential. For this reason, we started a process to share our approach and collect feedback. The first step was a webinar where we presented this work⁵, followed by a formal collection of feedback and we are planning to organise a workshop by the end of this year involving the largest expected EOSC user groups.

Finally, EOSC-hub intends to propose the contribution to the definition of the EOSC technical architecture described in this document, including the related approach to define the EOSC technical specification for building blocks, to the EOSC Architecture WG for its adoption in the wider EOSC environment, as soon as this WG will be fully operative. EOSC-hub would also like to collaborate with the WG on further refining the proposed architecture taking into account requirements and suggestions from the largest possible set of service providers and user communities.

⁵ <https://www.eosc-hub.eu/events/eosc-hub-proposal-eosc-technical-architecture>

1 Introduction

The aim of the work presented in this document is increasing the added value provided by EOSC and fostering its uptake through the definition of a reference Technical Architecture for EOSC that facilitates access to services, lower barriers to integrate and composes services and promotes the usage of services between adjacent communities. This is achieved identifying key technical functions, named building blocks in the rest of the document, for each of the EOSC service category (Federation, Access Enabling, Common and Thematic) and defining related technical specifications that include an **high-level architecture, suggested EOSC standards and APIs and interoperability guidelines**. In this way, EOSC ‘compliant’ services will offer well-established and documented interfaces for usage and integration, based on well-known standard or APIs, facilitating:

- their exploitation from user communities willing to create new scientific services that could rely on well-established and documented interfaces for the integration. An example of exploitation of EOSC services is when a community creates a new scientific workflow re-using EOSC federation and common services, like AAI, accounting, Cloud orchestrator and/or data management solutions.
- the combined usage of EOSC services, indeed the adoption of well-known standards and interfaces will very-likely reduce the cost to integrate services. For example, two accounting infrastructures can be made easily interoperable if they use the same standard usage record format, in such case accounting data extracted from them can be merged and presented in a unique view. Another example is about data processing and data management services implementing compliant interfaces that enable a jointly usage by a thematic services.

As a consequence, less mature or small scientific communities can leverage on EOSC services for a series of IT functions and focus on their scientific work, access to scientific services will be open to new communities thanks to the documented interfaces and new scientific workflows can be created combining existing applications.

This deliverable focuses on the EOSC Technical Architecture. The work on standard roadmaps mentioned in the title will be reported on the D10.1 and D10.2 EOSC-hub Technical Roadmap v1 and v2.

The document is organized as follows:

- Section 2 describes the landscape around the work on the EOSC Technical Architecture definition showing the connection of our work with the EC EOSC implementation roadmap and the EOSC Governance. Information on the past work on defining the EOSC Technical Architecture is also provided.
- Section 3 describes EOSC service categories and their organisation into EOSC portfolios as a basis for defining the architecture.
- Section 4 introduces the concept of end-to-end composition of services and how EOSC-hub is fostering the interoperability of services.

- Section 5 presents the proposed EOSC Technical Architecture describing a hierarchical structure and a functional view. A common approach to identify and detail each building block is depicted.
- Section 6 shows how requirements collected by several EOSC use cases are driving this work.
- Section 7 maps our outcomes with the objectives of the EOSC Architecture WG.
- Sections 8 and 9 present examples of EOSC technical specification for federation and common services.
- Finally, section 10 draws conclusions and describes next steps.

2 Landscape

This section describes the landscape around the work on the EOSC Technical Architecture definition showing the connection of this EOSC-hub effort with the EC EOSC implementation roadmap and the EOSC Governance.

Furthermore, a brief analysis of the past work on this topic is presented, notably the contribution for the definition of the EOSC Architecture of the EOSCpilot project.

2.1 EC Implementation Roadmap

In March 2018, the European Commission released a Commission Staff Working Document on the implementation roadmap for the European Open Science Cloud (EOSC)⁶ where a model was proposed that *describes a pan-European federation of data infrastructures built around a federating core and providing access to a wide range of publicly funded services supplied at national, regional and institutional levels, and to complementary commercial services. The model includes six actions lines: (a) architecture, (b) data, (c) services, (d) access and interfaces, (e) rules and (f) governance.*

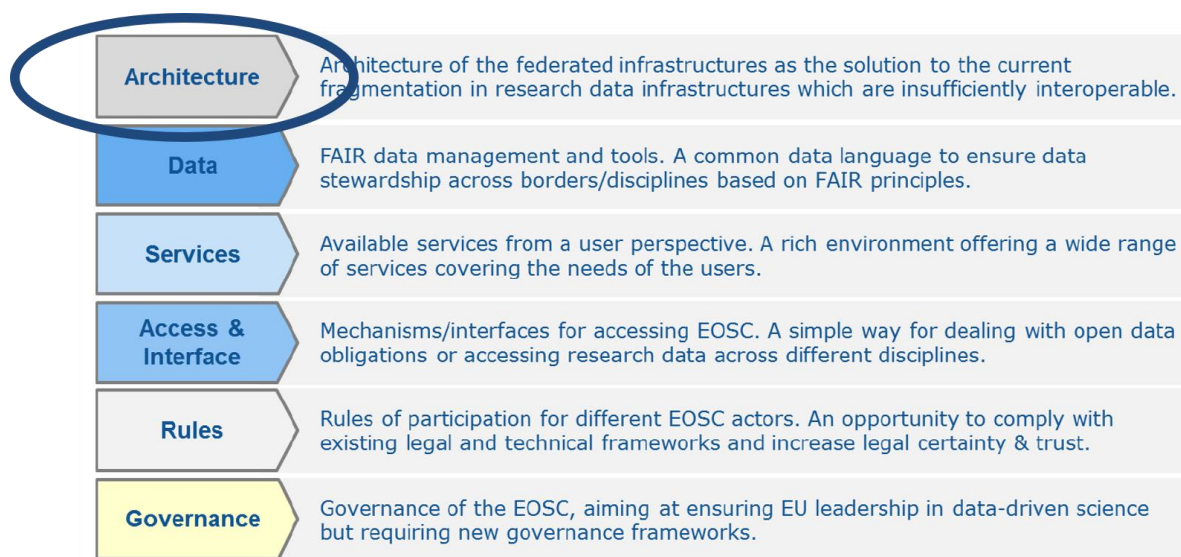


Figure 1. The six action lines of the EOSC implementation roadmap.

The Architecture action line, the most relevant for the work described in this document, foresees the creation of a *federation of existing and planned research data infrastructures, adding a soft overlay to connect them and making them operate as one seamless European research data infrastructure*. The EOSC architecture should *comprise a federating core and a variety of federated research data infrastructures committed to providing services as part of the EOSC offered through the EOSC hub*. Services are categorised in horizontal services, *such as a portal, authentication and authorisation and security services, allowing users to access the computing, data and services of*

⁶ Commission Staff Working Document - Implementation roadmap for the European Open Science Cloud: https://ec.europa.eu/research/openscience/pdf/swd_2018_83_f1_staff_working_paper_en.pdf

pan-European and disciplinary research data infrastructures, and in generic (also called common) or thematic services, for data storage, management and analytics, simulation and visualisation, distributed computing, etc. Furthermore, the document underlines that EOSC services should come from existing European data infrastructures. The EOSC hub would relay the resources and the services of data infrastructures funded at EU, national and regional level, and should be accessible from a central portal (e.g. the EOSC Portal), EOSC would provide a single, coherent access channel to EOSC services at European level that meets researchers' needs for data sharing, management and computing. Finally, it is mentioned that the federation of the services should be gradual and based on simple guidelines consistent with existing good practices.

A first attempt on defining the EOSC Federating Core is presented in the EOSC-hub briefing paper "EOSC Federating Core Governance and Sustainability"⁷, while the organisation of the EOSC services in portfolios according to the service categories depicted in the Commission Staff Working Document is described in the EOSC-hub D2.6 "First Service roadmap, service portfolio and service catalogue"⁸. Both concepts are shortly summarised in section 3. The work presented in this document leverages such definitions and builds the technical architecture for EOSC on top of them.

2.2 EOSC Architecture Working Group

The EOSC Governance⁹, to fully implement its structure, defined five working groups (WGs)¹⁰ to ensure a community-sourced approach to the current challenges of the EOSC:

- **Landscape:** Mapping of the existing research infrastructures which are candidates to be part of the EOSC federation;
- **FAIR:** Implementing the FAIR data principles by defining the corresponding requirements for the development of EOSC services, in order to foster cross-disciplinary interoperability;
- **Architecture:** Defining the technical framework required to enable and sustain an evolving EOSC federation of systems;
- **Rules of participation:** Designing the Rules of Participation that shall define the rights, obligations governing EOSC transactions between EOSC users, providers and operators;
- **Sustainability:** Providing a set of recommendations concerning the implementation of an operational, scalable and sustainable EOSC federation after 2020.

The activity of the Architecture WG is strictly related to the work EOSC-hub is doing on defining the EOSC technical architecture. This can be deduced by its main objective¹¹: [the WG] *proposes the technical framework required to enable and sustain an evolving EOSC federation of systems. Such a technical framework may include standards, APIs and protocols that will facilitate interoperable services delivered by diverse providers.*

⁷ <https://documents.egi.eu/document/3479>

⁸ <https://documents.egi.eu/document/3470>

⁹ <https://www.eoscsecretariat.eu/eosc-governance>

¹⁰ <https://www.eoscsecretariat.eu/eosc-working-groups>

¹¹ <https://www.eoscsecretariat.eu/working-groups/architecture-working-group>

The need for defining an EOSC interoperability layer to enable the end-to-end composition of services delivered by various providers is mentioned in the WG mandate. To achieve this objective the WG is intended to describe and/or define:

- *EOSC core services and their interfaces;*
- *EOSC open source APIs for reuse by thematic services;*
- *EOSC portal components and federated catalogues of service offerings;*
- *the EOSC data description standards;*
- *Standards and best practices necessary to ensure the evolution of EOSC and the widening of its user base to the industry and the public sectors.*

The EOSC-hub work on technical architecture has been shaped with the same objectives in mind; this is clearly described in section 7 where we mapped the outcomes of our work to the Architecture WG objectives. Through its representative in the WG, EOSC-hub is expected to propose the EOSC technical architecture described in this document and the related approach to defining EOSC standard building blocks (see later for details) to the WG for its adoption in the wider EOSC environment. EOSC-hub would also like to collaborate with the WG on further refining the proposed architecture, taking into account requirements and suggestions from the largest possible set of service providers and user communities.

2.3 Past work on the EOSC Technical Architecture

2.3.1 EOSC Pilot Service Architecture

The EOSCpilot project was the first initiative that worked on the definition of the EOSC technical architecture. The architecture model described in the deliverable EOSC-Pilot D5.4 Final EOSC Service Architecture¹² is based on 47 classes of services *needed to develop and operate a system suitable to support the EOSC mission and goal*. These classes of services were organised in architecture from both a user and a functional perspective.

In the functional architecture, services were split into five categories:

- **Front-end services**, for implementing the part of the overall service with which users will interact directly, namely portals or APIs;
- **Security & Trust**, aimed at guaranteeing that the overall system (and the services) operate securely and according to standards;
- **Open Science, Data Management, Analytics**, aimed at providing their users with user- and open-science-friendly facilities, enabling users to focus on science tasks;
- **EOSC System Governance & Management**, dedicated to supporting the operation and management of the overall EOSC System;
- **Compute & Cloud Platforms**, offering generalist resources like virtual machines and containers as well as network transport connectivity. In addition, all the platforms and software that do not belong to the other categories falls here.

The following figure shows an overall view of the function architecture proposed by EOSCpilot.

¹² <https://eoscipilot.eu/content/d54-final-eosc-service-architecture>

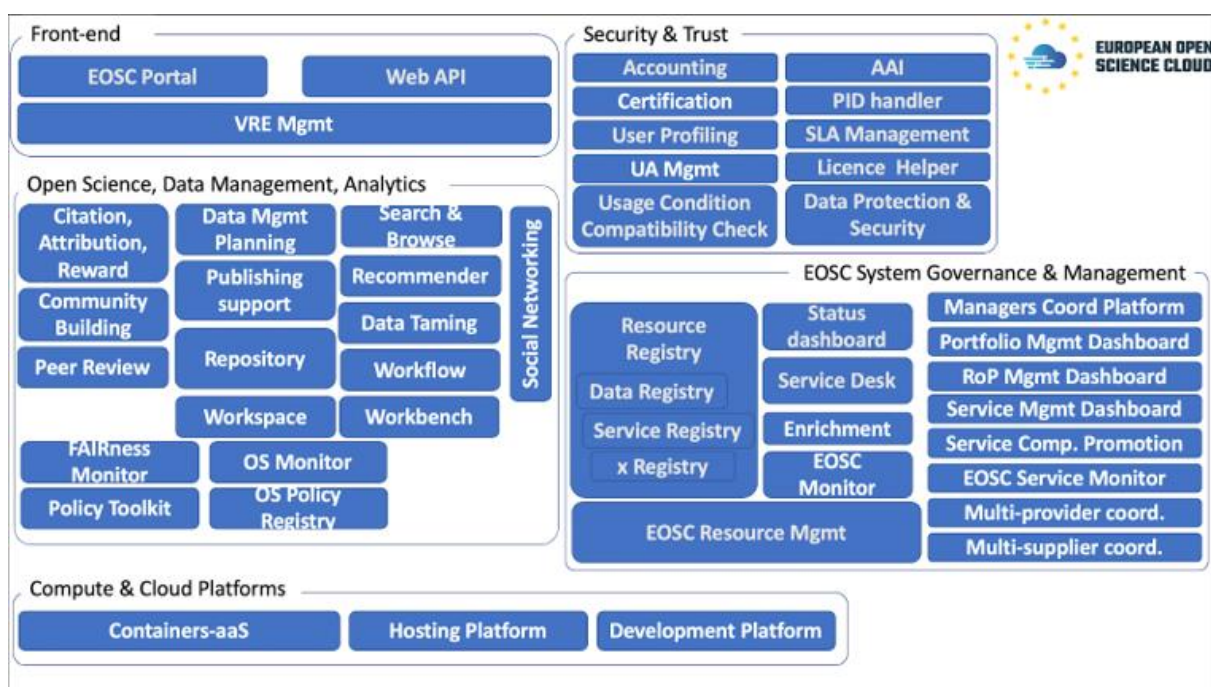


Figure 2. EOSCpilot - proposed EOSC function architecture

There are many similarities between the EOSCpilot functional architecture and the work described in this document. Indeed, the EOSC technical architecture described in the next sections is also based on a classification of the services according to their functions, similar to the one proposed by EOSCpilot. Leveraging on the EOSCpilot experience, EOSC-hub refined the service classification, also taking into account the Commission Staff Working Document on the implementation roadmap for the EOSC, and went further ahead:

1. clarifying the interactions between the different service classes (the first level on the hierarchy in the proposed EOSC technical architecture).
2. defining an approach to create EOSC technical specifications and interoperability guidelines for each service/feature offered by EOSC.
3. proposing EOSC technical specifications and interoperability guidelines for key EOSC services/features.

2.3.2 EOSC-hub Technical Architecture v1

The first EOSC-hub deliverable on EOSC technical architecture¹³ established the groundwork for the work described in this document. It identified the EOSC service types and their relationships and introduced the concept of end-to-end compositions of services describing the effort of the project on fostering the service interoperability both promoting the adoption of well-known standard and with ad-hoc integration activities driven by user requirements.

The document also presented the procedures to extend the EOSC service offer, federating/on boarding new services, and a deep analysis on the main standards, APIs and protocols used by the

¹³ EOSC-hub D10.3 Technical Architecture v1: <https://documents.egi.eu/public/ShowDocument?docid=3417>

services belonging to a specific technical area. Such analysis was a needed preparatory phase to start the definition of the EOSC technical specifications.

The work presented in this document further analysed the concept of end-to-end compositions of services and built architecture that, from one side, leverages on the defined EOSC service types and relationships, and, from the other side, fosters service interoperability providing specifications and guidelines to develop and integrate services.

3 The EOSC Portfolios and the EOSC Federating Core

The proposed EOSC Technical Architecture is based on the different classes of EOSC services and on their interactions. Then, an introduction on such service categories and on their functions and relationships is necessary before describing the architecture.

As depicted in Figure 3, EOSC services are organised in two service portfolios:

- **EOSC Service Portfolio:** the external services which EOSC-hub either provides from its partners or onboards from the community to contribute to the larger portfolio of researcher-benefitting services within EOSC. The EOSC Service Portfolio contains:
 - **Thematic services:** community-specific capabilities including research core data, data products, scientific software, and pipelines. Examples of thematic services are: data resources and software tools to access, study and compare the data; data brokering services tailored to the needs of specific scientific communities;
 - **Common services:** they provide generic capabilities usable by any science discipline each supporting aspects of the data lifecycle from creation to processing, analysis, preservation, access and reuse. Examples of services belonging to this category are multi-disciplinary services for data discovery, processing, workflow management and orchestration, data management, etc.
- **Hub Portfolio:** the internal services contributing to the federating core of EOSC, both for internal operation of the EOSC Hub and to offer as components to be integrated into the services of the EOSC Service Portfolio. They enable the other EOSC elements to deliver (greater) value to researchers across Europe. They can be further split in
 - **Access-enabling services:** delivering features allowing customers to easily exploit EOSC resources such as discovery, ordering and workflow enabling services
 - **Federation services:** needed to operate the EOSC e.g. a common helpdesk, accounting information gathering, monitoring

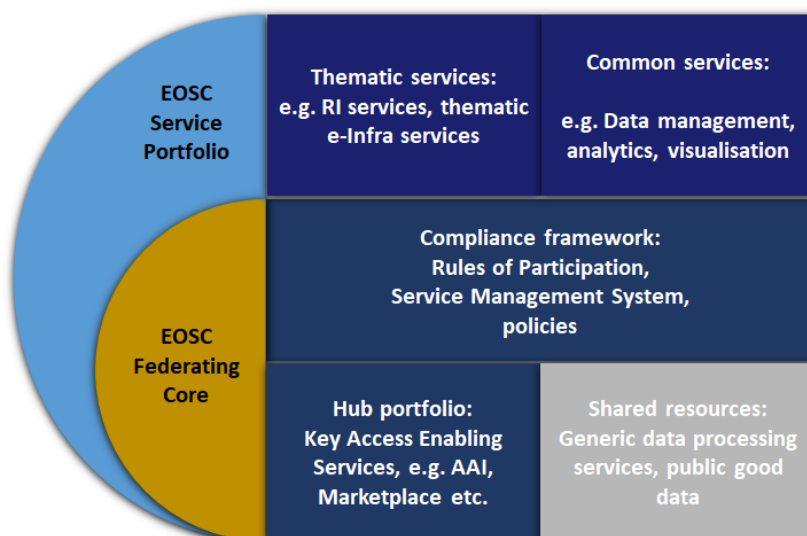


Figure 3. EOSC Service Portfolios and EOSC Federating Core

Thematic services can be integrated with the services in the Hub portfolio to facilitate the users' access (e.g. the EOSC Portal and Marketplace) or to avoid re-implementing basic features, like authentication and authorisation, accounting, monitoring, etc. They can also adopt common services that already address some of their technical needs. Common services can also leverage on services of the Hub portfolio to deliver some of their functions. The integration of thematic, common, access enabling and federation services can be fostered through a large adoption of open and standard interfaces.

The Hub Portfolio is one of the key elements of the **EOSC Federating Core** together with the Compliance framework - made of Rules of Participation, EOSC Service Management System and other policies - and the Shared resources, a set of generic data processing and managing services, commodity services, compute and storage resources and public good data managed and offered centrally by EOSC. More information about the EOSC Federating Core and the EOSC service portfolios are available in the EOSC-hub briefing paper "EOSC Federating Core Governance and Sustainability"¹⁴ and in the EOSC-hub D2.6 "First Service roadmap, service portfolio and service catalogue"¹⁵.

In the context of the Technical Architecture, we then discuss three categories relevant to our technical work:

- Federation and access enabling services
- Common services
- Thematic Services

Federation and access enabling are key services needed to operate the EOSC (e.g. the EOSC Portal or the accounting infrastructure). Common services offer add-value features on top of EOSC resources (computing, storage, etc.) and can be reused by a multitude of other services. Thematic services implement discipline specific features and are provided directly by scientific communities.

¹⁴ <https://documents.egi.eu/document/3479>

¹⁵ <https://documents.egi.eu/document/3470>

4 Service Composability

The end-to-end composition of the services can be considered one of the most important added values provided by EOSC. Indeed, the service composability would allow EOSC service providers and users to select various services offered by EOSC and compose them according to their needs to create added-value solutions for research.

Typical service combinations are:

- A thematic service adopts some EOSC federation services to implement basic features (AAI, monitoring, accounting).
- A thematic service adopts common services that provide features to better exploit compute, storage and data resources including those offered by distributed infrastructures.
- An EOSC user creates new scientific workflows integrating, for example, a data repository and some analytics services together.

The adoption of standard interfaces makes some of the services of the EOSC service catalogues already interoperable, these sub-classes of composable services need to be identified and made accessible through the EOSC Portal. Furthermore, in response to newly emerging needs from communities, other services can be made interoperable through integration activities. EOSC should provide technical guidelines (in terms of suggests EOSC standards and APIs) and technical support to both integrate services and facilitate the combined usage of (already) interoperable services.

Enabling the service composability would allow to lower the barriers for developers of the thematic services to reuse common, federation or access enabling services to implement basic features (AAI, accounting, monitoring, etc.) and exploiting in the best way compute, storage and data resources. Indeed, they, from one side, can focus on working on increasing the scientific added value of their services, and from the other side, rely on well-established and EOSC-compliant services for implementing the basic features. A large part of these reusable services will come from the experiences of the main European e-infrastructures and other relevant initiatives (such as those involved in the project, EGI, EUDAT and INDIGO-DataCloud).

4.1 Fostering the service interoperability

Interoperability is a key concept to enable the end-to-end composition of services in EOSC. EOSC-hub is already working on fostering the interoperability in EOSC in a dual way:

- identifying services that can already work together because they support the same standards and/or interfaces. An example is shown in Figure 4 where the CREODIAS DATA HUB¹⁶, one of the DIAS platform funded by the EC for handling Copernicus Data, is working with the Sentinel Hub¹⁷, a tool that uses Copernicus data to create maps. The services are

¹⁶ <https://marketplace.eosc-portal.eu/services/cloudferro-data-collections-catalog>

¹⁷ <https://marketplace.eosc-portal.eu/services/sentinel-hub>

already interoperable because they both support the OGC WMS standard¹⁸. Therefore, specific integration work is not needed and EOSC should provide technical support to the community willing to exploit these services in a combined manner.

- integrating federating, common and thematic services according to identified users' requirements. An example is shown in Figure 5, the DODAS analytics services has been integrated with the EGI Federated Cloud, to use its cloud resources, and the EGI Check-in services, to implement the AAI. In such a case, the integration required some development. The solution made of these integrated services is used by the CMS community. The solution can be reused by other communities without any further integration work.

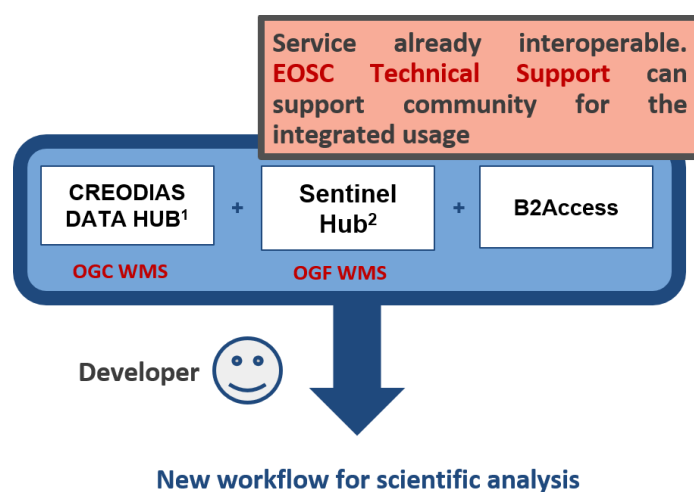


Figure 4. Examples of composition of services supporting the same standard (e.g. OGC WMS).

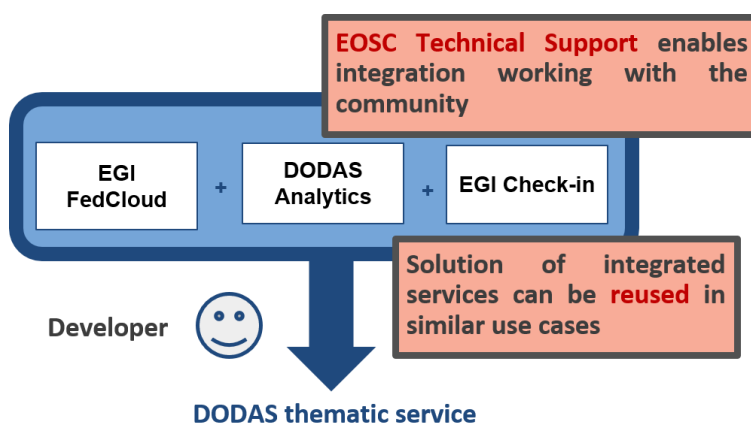


Figure 5. Examples of composition of services obtained through an integration activity.

¹⁸ <https://www.opengeospatial.org/standards/wms>

Both approaches allow identifying and extending the set of services that can work together /can be composed. These solutions can be offered to all the EOSC users that should be able to recognise and reuse them, also thanks to the technical support offered by EOSC.

This work to identify and make services interoperable would be easier if EOSC interfaces for integration, possibly based on well-known standards and API, would have been already available together with clear instructions or procedures to allow services to interact and work together. Interfaces and instructions to make interoperable a given service or a technical feature can be called **EOSC interoperability guidelines**. The full set of such interoperability guidelines can be considered a valid implementation of the EOSC interoperability layer mentioned in the mandate of the EOSC Architecture WG. The model of technical architecture presented in section 5 is based on these concepts and is thought to foster the service interoperability and, then, the end-to-end composition of services.

The EOSC interoperability guidelines that are being defined in the context of this work will take into account and will be based on existing community practices, well-known standards and interfaces. All relevant EOSC stakeholders (communities, e-infrastructures, etc) should be able to describe and promote their standards and practices for their inclusion in the EOSC guidelines. Adoption of these guidelines by providers will not be forced, making them mandatory, but should be a natural consequence of the advantages, for a service, generated by being compliant such as offering access through well-known interfaces, low cost to interoperate with other EOSC services, etc.

4.2 Federating thematic services in the EOSC

When a provider of a thematic service decide to join the EOSC, it should be able to consult the EOSC service offer and, consequently, decide (or not) to adopt/integrate EOSC services. EOSC will present to the provider its service portfolios (the Hub portfolio and the EOSC portfolio) as a sort of à-la-carte menu from which the provider can, first of all, understand the benefits of adopting a certain service and, then, can assess the technical feasibility and the related cost of the integration. This technical assessment will be possible only if the selected EOSC services offer well-established and documented interfaces for the integration, the EOSC interoperability guidelines. Furthermore, the integration cost will be very-likely reduced if such interfaces are based on well-known standard and interfaces.

After the analysis of the EOSC service offer, the provider of the thematic service can decide which EOSC services to adopt. This choice can be very different for each provider; some providers can decide to join the EOSC with no integration except listing their services in the service catalogue and/or marketplace. Others can opt for a tighter integration, adopting services from both the Hub and EOSC portfolios. For example thematic services can decide to adopt only the Marketplace and the AAI from the Hub Portfolio, while another thematic service can use a plethora of services from both portfolios (Marketplace, AAI, Accounting, and Monitoring and Helpdesk from the Hub Portfolio, a cloud orchestrator and a data management tool from the EOSC Service Portfolio).

5 Defining the EOSC Technical Architecture

This section details the process used by EOSC-hub to define the EOSC Technical Architecture. The architecture presented is a reference architecture where service categories, building blocks and related interfaces are identified. It focuses on the concepts of service interoperability and composition introduced in the previous section, fostering the definition and the adoption of EOSC standards and interfaces. EOSC-hub is proposing an implementation of this reference architecture as described in section 5.2.

5.1 Reference Architecture

As stated above, the EOSC Technical Architecture presented in this section is reference architecture. In the field of software architecture or enterprise architecture, reference architecture provides a template solution for architecture for a particular domain. It includes a common vocabulary with which to discuss implementations, often with the aim of stressing commonalities. A reference architecture often consists of a list of functions, some indication of their interfaces (or APIs) and interactions with each other and with functions located outside of the scope of the reference architecture¹⁹.

Reference architectures can be defined at different levels of abstraction, in the context of EOSC, EOSC-hub decided to work at the infrastructure/technical level. As part of this work, we are also defining a common vocabulary that can be used to define both existing services and those joining EOSC catalogue in the future. The architecture includes functions, interfaces, APIs and standards as technical concepts, with the final aim of fostering interoperability and, ultimately, service composability. It is based on a hierarchical structure with three levels. These are:

1. Category (the service categories introduced earlier).
2. Functional categories within the main category.
3. Individual building blocks usable in fulfilling these functions.

An overview is seen in Figure 6.

¹⁹ S. Angelov, P. Grefen and D. Greefhorst, "A classification of software reference architectures: Analyzing their success and effectiveness," 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, Cambridge, 2009, pp. 141-150.

doi: 10.1109/WICSA.2009.5290800

URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5290800&isnumber=5290660>

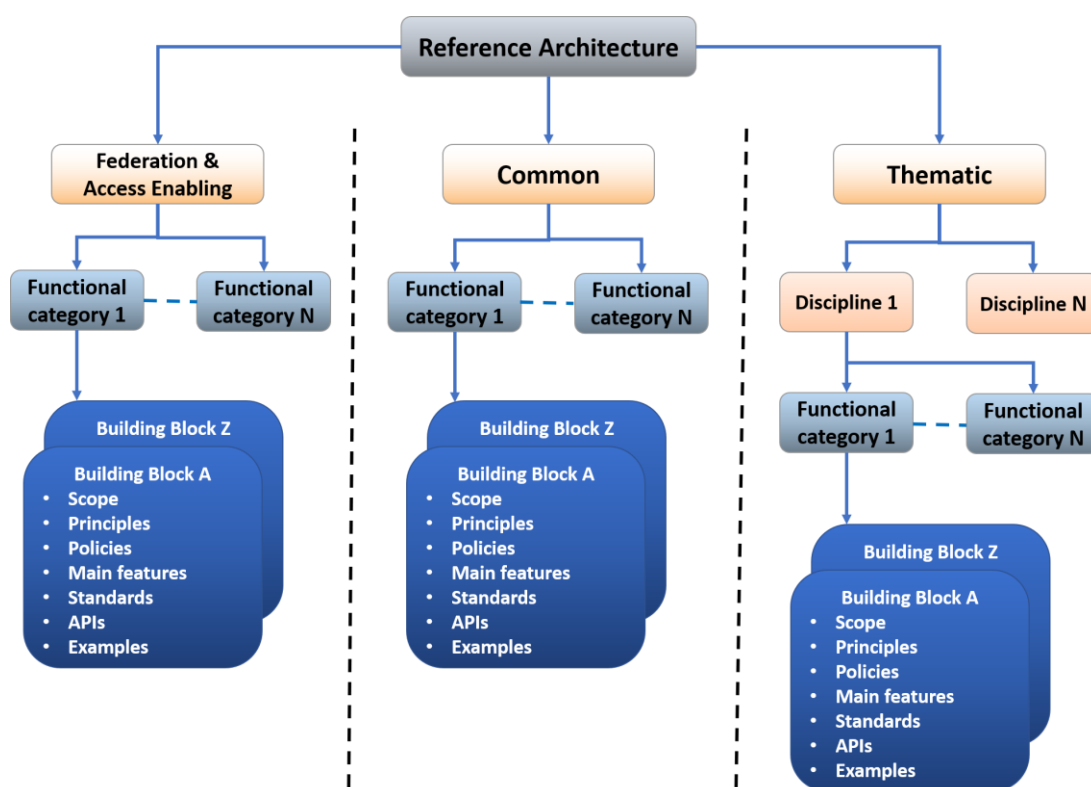


Figure 6. Hierarchical structure in the EOSC reference architecture.

The subdivision in categories allows differentiating services according to their function within EOSC: The top level categories (Federation & Access Enabling, Common and Thematic) have already been introduced. The second level of the hierarchy introduces the functional categories that groups technical functions to facilitate their identification. To take an example, within Federation and Access enabling services, we may see Authentication & Authorisation or Monitoring as functional categories. In the case of thematic services, the functional categories are identified per scientific discipline.

Beneath this, we see the individual building blocks that implement technical functions. To continue the example, within the Authentication & Authorisation functional category, we see the AAI building block.

The reference architecture described in this section gives flexibility on defining the second and third level of the hierarchy, functional categories and building blocks. Section 5.2 presents the implementation of this reference architecture proposed by EOSC-hub where functional categories and building blocks are started to be defined for each service category.

5.1.1 Approach to define building blocks

EOSC-hub is working on defining the building blocks of the architecture for each service type and has specified a common approach to complete this task. It foresees the identification of the main **building blocks/technical functions** in each service category and, for each of those, defining a **technical specification** that includes an **high-level architecture, suggested EOSC standards and**

APIs and interoperability guidelines. This method would allow providers offering services implementing the technical function of a given building block to be compliant with the related EOSC technical specification. As a consequence, thanks to the provided guidelines, interoperability between services offering the same technical function(s) and following the EOSC specifications will be easier to achieve. Hence sets of services implementing the same building block and compliant with the EOSC specification can be made able to work together with less effort, to deliver a given technical function in the EOSC environment. Examples of these service families can be AAI services compliant with the AARC blueprint architecture and guidelines or monitoring and/or accounting systems able to exchange/share information and provide integrated views to the EOSC customers and service providers. This approach is tailored to the varied environment seen in EOSC, where many solutions to satisfy a given technical requirement already exist.

Furthermore, the definition of EOSC standards and APIs along with related interoperability guidelines for each of the identified building blocks will foster the end-to-end composition of services. Being compliant with a specification for a given building block, would allow a service to interoperate with other services offering the same function (as described above) and, conversely, building blocks/services offering different technical functions can interoperate thanks to the EOSC interfaces, described in the technical specification (e.g. it would be easier for a thematic service integrating a common service if clear interoperability guidelines are available).

EOSC interoperability specifications are not intended to be mandatory, but being compliant with them would be an added value for services. Indeed they could interoperate with other services with less effort and reduced cost. Therefore, providers willing to expand their user base by making their services composable will be inclined to support such specifications.

In this approach, identifying building blocks and the respective technical specifications could be a complex and long work, so the consortium has agreed to follow an iterative approach, starting from the functions that are more requested by the EOSC use cases²⁰. Technical specifications, initially prepared by the technical experts within the EOSC-hub project, should also be iteratively improved, collecting feedback by external people with expertise in the area and involving them in the maintenance and evolution of such specifications. The same is true for the list of building blocks: they will evolve and change in the future, adding/removing functions depending on the user requirements and on the projects/service providers that may join the EOSC in the future. This will be an ongoing, continuous activity that should be continued within EOSC after the end of the project.

²⁰ EOSC-hub is taking into account in this work requirements collected from EOSC Pilot Scientific Demonstrator (see [D5.6 Evaluation Report of service pilots](#)), EOSC-hub Thematic Services (see [D7.2 First report on Thematic Service architecture and software integration](#)), EOSC Competence Centers (see [D8.1 Report on progress, achievements and plans of the Competence Centres](#)) and EOSC use cases identified through the EOSC Portal (see the [EOSC-hub Community Requirements Database](#)).

5.1.2 Technical Specification template

We have defined a template to collect information about each of the identified building blocks and define a technical specification, regardless of the service category they belong to. It is structured as follows:

- Introduction: short description of the building block highlighting its main functions.
- High-level Service Architecture: reference architecture of the building block, highlighting the interfaces towards the other building blocks. It does not refer to any specific service.
- Adopted Standard: list with references of the main adopted standards and protocols/API.
- Interoperability guidelines: describe how services implementing this building block can be made interoperable.
- Examples of solutions implementing this specification: list of already available services that are compliant with this specification.

The complete template is available in Appendix I.

5.2 Proposed EOSC Technical Architecture

Figure 7 shows the functional view of the proposed EOSC technical architecture, as implementation of the reference architecture described in the previous section, where the interactions between services belonging to different categories are highlighted.

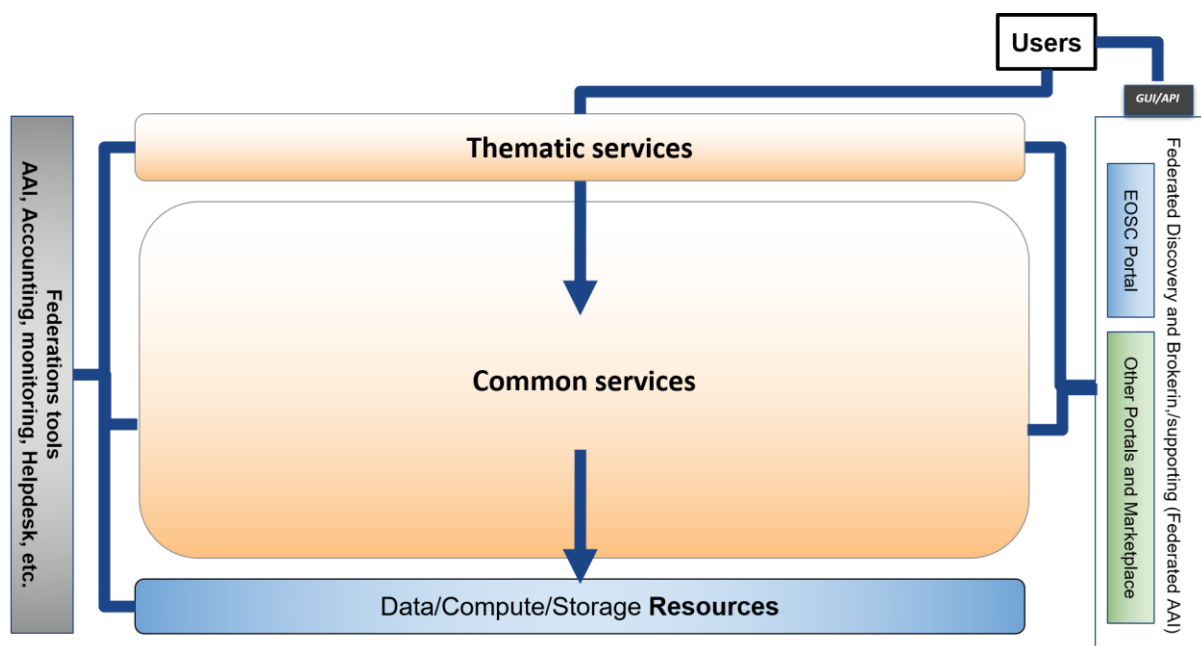


Figure 7.EOSC Technical Architecture - Functional view.

EOSC users can exploit EOSC Thematic and Common services directly or through the GUI or API of an access enabling services like the EOSC Portal (or others portals and Marketplaces). Thematic services can leverage Common services for added value features on top of data, compute and storage resources. Federation tools support all these services providing basic features like authentication and authorisation, accounting, monitoring, etc. Pledged shared resources centrally

managed by EOSC, including both commodity services and service capacity, are part of the Resources and complement other EOSC resources directly managed by other service providers.

This functional view will be better detailed in the following sections of the document with information on the already identified and defined building blocks per service category.

5.2.1 EOSC Access Enabling and Federation services

In the Access Enabling and Federation service categories, a **building block is any key access-enabling and federation function needed to operate the EOSC**. Services offering these features according to the EOSC specification could be onboarded on the Hub service portfolio described in section 3.

We already identified an initial list of building blocks for this category, leveraging the experiences from some of the largest European e-infrastructures that are involved in the project. In this initial phase, we have a one-to-one mapping between functional categories and building blocks, this may change in the future.

This list is detailed in the table below.

Functional categories/Building blocks	Short description
EOSC Portal	<p>The EOSC Portal provides a European-level delivery channel, connecting the demand-side (the EOSC Customers) and the supply-side (the EOSC Providers) to allow researchers to conduct their work in a collaborative, open and cost-efficient way for the benefit of society and the public at large. In particular it delivers the following functions:</p> <ul style="list-style-type: none"> ● Enable different kinds of users, with different skills and interests, to discover, access, use and reuse a broad spectrum of EOSC Resources (services, datasets, software, support, training, consultancy, etc.) for advanced data-driven research ● Support interdisciplinary research and facilitate Resource discovery and access at the institutional and inter-institutional level ● Allow researchers and institutions to focus on value creation through sharing and reuse as opposed to duplicating Resources and increase excellence of research and European competitiveness ● Improve the provisioning of access to integrated and composable products and services from the EOSC Catalogue ● Facilitate the composition of services and products to support multi-disciplinary science for example with high-level community-specific interfaces for running workflows involving EOSC services ● Help Providers gain additional insight into potential User groups outside their traditional constituencies ● Give Providers the possibility to offer Resources under homogeneous terms of use, acceptable use policies, and in

	<p>different configuration options, so that Users are guided in the choice.</p> <p>Use case. The Portal is particularly relevant to support on-demand access to EOSC through Business-to-User (B2U) and Business-to-Business (B2B) transactions.</p> <ul style="list-style-type: none"> ● B2U is applicable for consumer-oriented Resources appealing to a large potential User pool. B2U transactions will address the digital needs of individual researchers and short- and medium-term research projects. Because of the potential large user base, B2U transactions will be most suitable for those Resources supporting automated or semi-automated provisioning, a short acquisition process, requiring a low-level of specialisation, and which can be easily compared and chosen without requiring expert support. ● On the other hand, B2B applies predominantly to the acquisition of bespoke solutions and/or of large quantities of EOSC Resources involving potentially multiple Providers. B2B suits the needs of research performing organisations and research infrastructures which need to cater for the long-term needs of a large pool of end users. <p>The EOSC Portal Concept 2.0²¹ provides extensive information on potential use cases and a participatory model for resource providers, which are provided with the choice of selecting different EOSC participation levels.</p>
AAI	<p>The EOSC AAI aims to enable seamless access to multiple research data and services in EOSC in a secure and user-friendly way. It also provides authorisation management for access control. It is based on the AARC blueprint architecture.</p> <p>The EOSC AAI follows the architectural and policy recommendations defined in the AARC project²². As such, it enables interoperability across different Service Provider(SP)-Identity Provider(IdP)-Proxy services, each of which acts as a bridge between the community-managed proxies (termed Community AAIs) managing the researchers' identity and the generic services offered by Research Infrastructures and e-Infrastructures (termed R/e-Infrastructures or Infrastructures). This is the "community-first" approach to the AARC Blueprint Architecture²³, which enables researchers to sign in with their community identity via their Community AAI. Community-specific services are connected to a single Community AAI, while Infrastructure Services are connected to a single Infrastructure Proxy. Lastly, generic services may be connected to more than one Community AAI. Each Community AAI in turn serves as a bridge between external identity providers and the proxies to the e-infrastructure services. Specifically, Community AAIs connect to eduGAIN as service providers but act as identity providers from the</p>

²¹ <https://wiki.eosc-hub.eu/display/EOSC/EOSC+Portal>

²² <https://aarc-community.org>

²³ <https://aarc-project.eu/guidelines/aarc-g045/>

	<p>services point of view, thereby allowing users to use their credentials from their home organisations. Complementary to this, users without an account on a federated institutional Identity Provider are still able to use social media or other external authentication providers for accessing services.</p> <p>Research communities can leverage the EOSC AAI services for managing their users and their respective roles and other authorisation-related information. At the same time, the adoption of standards and open technologies, including SAML 2.0, OpenID Connect, OAuth 2.0 and X.509v3, facilitates interoperability and integration with the existing AAls of other e-Infrastructures and research communities.</p> <p>Use Cases. Access to all EOSC shared resources and access enabling services (e.g. the Portal, the Helpdesk, EOSC data and compute and storage resource tier) will require federated authentication and authorisation.</p>
Helpdesk	<p>The helpdesk is the tool that supports Incident and Service Request Management to restore normal/agreed service operation within the agreed time after the occurrence of an incident, and to respond to user service requests. The service works as a unified ticketing system, by connecting individual providers' helpdesks to the central helpdesk instance, offering a standalone service interface.</p> <p>Use case. The helpdesk tool is necessary to support Incident and Service Request Management of the resources provided by EOSC. The helpdesk can be implemented as a distributed platform linking together the helpdesks of suppliers offering resources to EOSC. The linking of existing helpdesks allows streamlining of support processes involving multiple suppliers, and in particular facilitates the work of the support teams that, through linking, are able to use existing in-house tools.</p>
Monitoring	<p>Monitoring provides the capability to check the status of service end-point interfaces and aggregate such information for the production of service reports. In particular, it should provide a scalable framework for monitoring the status, availability and reliability of end-points. It provides monitoring of services, visualisation of their status, dashboard interfacing, notification and generation of availability and reliability reports. Third parties can gather monitoring data from the system through a complete API.</p> <p>Use case. Monitoring information supports Service Report Management, and is consumed to produce Service Reports, i.e. the documents that provide the details of the performance of a service against the service targets defined in service level agreements (SLAs) – often based on key performance indicators (KPIs). Typical users are the EOSC service suppliers.</p>
Accounting	<p>Accounting is about collecting, aggregating, storing and displaying EOSC resource usage data produced by the providers participating in EOSC, for example from the providers of Shared Resources. It gathers usage information from the individual resource providers and aggregates it</p>

	<p>centrally in a secure, GDPR-compliant manner. Accounting is necessary for providing control over resource consumption by the funders, and reduces the overhead of each separate resource provider defining accounting information models, architecture and setup. Accounting is a key service of the EOSC federating core that will support its business models, and provides transparency on which resources are being used. The correlation of usage data to service identifiers, scientific product identifiers and user identifiers, supports the development of metrics that relate scientific impact to the extent a researcher and/or project has been embracing open science practices.</p> <p>Use case. Accounting of resource usage is required for any EOSC customers (e.g. platform operators and research infrastructure managers) to enable aggregated information on usage of scientific products and services used from the EOSC portfolio.</p>
<p>Federated Configuration Management DataBase (CMDB)</p>	<p>A configuration database is a database used by an organisation to store information about hardware and software assets (commonly referred to as Configuration Items). This database acts as a data warehouse for the organisation and also stores information regarding the relationship between its assets. The CMDB provides a means of understanding the organisation's critical assets and their relationships. At a federation or EOSC-level, it is a database drawing selected configuration information from provider CMDBs, which is needed at the EOSC or federation level.</p> <p>Use case. The availability of an EOSC CMDB is relevant to EOSC shared resource suppliers, and is requested by the IT configuration management process. It allows the management of the provision of services owned and managed by the EOSC governance. It is envisaged that the management of resources published in EOSC just for the purpose of improving their discoverability, will be delegated to the respective providers and will not be registered in an EOSC CMDB.</p>
<p>Order management</p>	<p>Order management is a process allowing the portal operators to handle orders received through the EOSC Portal. It implements interfaces towards service provider order management processes to support orders that should not be centrally processed in EOSC.</p> <p>Use case. Managing orders from the EOSC Portal.</p>
<p>Operations Portal</p>	<p>The Operations Portal refers to the set of control dashboards that support the work of EOSC infrastructure managers in charge of supervising the overall status, allocation and accessibility of the EOSC shared resources. It provides central operations management of federated resources. The Operations Portal offers a portfolio of management tools to support communications, customer relationship management, infrastructure oversight, and metrics gathering.</p> <p>Use case. The Operations Portal can support multiple service management activities like incident management and order management if used as a back-office tool of the EOSC Portal.</p>
<p>Service Portfolio</p>	<p>The Service Portfolio Management Tool (SPMT) allows lifecycle</p>

Management Tool	<p>management of the services provided through EOSC. SPMT allows providers to capture, store and maintain key information about their services, and to easily publish that data into an EOSC-mandated service catalogue, such that that hosted on EOSC-Portal.</p> <p>Use case. The tool is used by providers with one or more services which they which to deliver through EOSC. It simplifies their management of information about the services, simplifies delivery of this information to those managing onboarding to EOSC, and this simplifies the publishing of these services in a public catalogue.</p>
Collaboration software & platforms	<p>Tools needed to operate a 'Hub' or federating core for EOSC. These include collaborative documentation and document creation and management systems, issue management for task tracking and communication tools to manage remote collaborations.</p> <p>Use case. Collaborations between EOSC users and/or service providers.</p>
Security monitoring	<p>Provide features to monitor the security of the EOSC services and resources.</p> <p>Use case. Identify security threats in the EOSC.</p>
Messaging	<p>A real-time messaging service allowing to exchange messages between independent applications.</p> <p>Use case. Enabling asynchronous communication between EOSC services.</p>
Software quality assurance	<p>A tool allowing to deliver quality 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 instead of manual-based validation mechanisms.</p> <p>Use case. Automated validation of software quality.</p>

Technical specifications for all these building blocks are under preparation and will be published for feedback²⁴ as soon as they are ready. The maturity level of the technical specifications of these building blocks varies; an example of an already mature specification is that for AAI, which is described later in the document. We intend to have mature specifications for all the building blocks of this category by the end of 2019.

The EOSC Portal is as special case within this category. It is currently being further enhanced and developed by a large collaboration that includes EOSC-hub, OpenAIRE Advance and key partners from the former eInfraCentral project. More information is available in the EOSC Portal concept

²⁴ Feedback will be collected through a public consultation that will be launched in the EOSC-hub web-site.

paper.²⁵ The outcomes of this collaboration will be adopted by this work to technically specify the EOSC Portal. Some details about the status of this activity are reported in section 5.4.

5.2.2 EOSC Common services

In the Common services, **a building block is a technical function that offers added value on top of EOSC resources (computing, storage, etc.) and that can be adopted by multiple thematic services.** Examples of building blocks for this category are Infrastructure as a Service (IaaS) Virtual Machine (VM)/Container management, Cloud Orchestration, metadata management, making scientific artefacts FAIR, etc. A building block in the Common Services category can be implemented and, then, offered by one or more common services.

In this category the number of relevant building blocks can be huge, so we must split the work into sub-areas or functional categories. We used the different technical areas EOSC-hub is working on as a basic functional division to start the process of identifying the building blocks:

- HTC/HPC Compute
- Cloud Compute (including Containerisation and Orchestration)
- PaaS Solutions
- Data Platforms for Processing
- Data Publishing and Open Data
- Data Preservation/Curation/Provenance
- Metadata Management and Data Discovery
- Workflow management, user interfaces and Data analytics

Other functional categories could be added by other initiatives according to their expertise. For example, OpenAIRE suggested the addition of a ‘Scholarly Communication’ category and proposed building blocks for this area during the last EOSC-hub technical workshop in Amsterdam²⁶.

How the work was split into technical functions is shown in the following figure. Within the Common Services category, a set of functional categories/technical areas were depicted (only four areas are shown as a sample) to simplify the identification of the building-blocks.

²⁵ <https://wiki.eosc-hub.eu/display/EOSC/EOSC+Portal>

²⁶ <https://indico.eji.eu/indico/event/4675/overview>

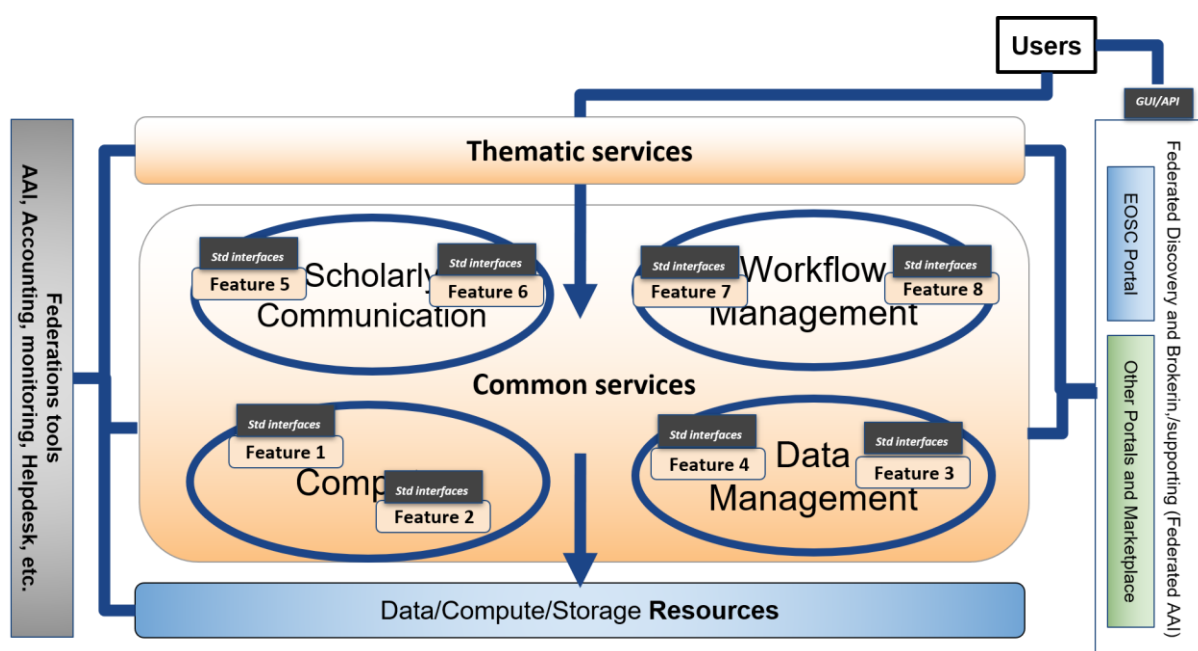


Figure 8. EOSC Technical Architecture. Building blocks per functional category.

For the Common Services, we agreed to prioritise the preparation of the technical specifications for the building blocks that are more relevant to users according to the use case analysis. The current list of identified building blocks, organised per technical area, is in the following table.

Functional categories	Building blocks
HTC/HPC Compute	<ul style="list-style-type: none"> • Multitenant job submission • Multitenant container based job submission • HTC / HPC clusters on demand
Cloud Compute (including Containerisation and Orchestration)	<ul style="list-style-type: none"> • IaaS: VM Management • IaaS: Orchestration • IaaS: Containers
PaaS Solutions	<ul style="list-style-type: none"> • PaaS Solution for Cloud service automation and federation of hybrid Cloud resources
Data Platforms for Processing	<ul style="list-style-type: none"> • Transparent data processing using POSIX in distributed and hybrid cloud environments including Docker, Kubernetes and Jupyter (Notebooks and Hub) • Data Ingestion and transfer for processing in hybrid cloud environment • Metadata Management in processing workflows • QoS based data access optimization and tight integration with preservation services • Authorization based on attributes from IdP • Results sharing and experiment repeatability

	<ul style="list-style-type: none"> ● Distribution of software for the processing tasks
Data Publishing and Open Data	<ul style="list-style-type: none"> ● Data Repository
Data Preservation/Curation/Provenance	<ul style="list-style-type: none"> ● Data Preservation ● Tracking of provenance metadata ● Data Curation
Metadata Management and Data Discovery	<ul style="list-style-type: none"> ● Data Discovery and Access ● Metadata cataloguing and indexing ● Annotation service ● Cloud based IoT Platforms interoperability
Workflow management and user interfaces and Data analytics	<ul style="list-style-type: none"> ● Portals ● Big data analytics ● ML/DL analytics services
Scholarly Communication	<ul style="list-style-type: none"> ● Data Management Plans ● Digital Preservation ● Overlay platforms: Peer-review ● Anonymization ● Aggregator ● Broker ● Entity Registry ● Metadata validation ● Annotation ● Usage stats ● VRE: RI Services for experiments

Examples of completed technical specification for building blocks belonging to the Common Services are presented later in this document.

The picture 9 shows how the functional view of the EOSC technical architecture will appear when the first set of building blocks and the related technical specifications are well defined. Thematic services could easily exploit building blocks offered by common services through the EOSC standard interfaces (purple arrows in the figure). Also common services implementing such building blocks can be made interoperable in an easier way thanks to the EOSC standard interface (red arrows in the figure) offering a combined usage to the thematic services. In this scenario, service composability would be easier to attain and the cost of integration work will be reduced with respect to the current situation.

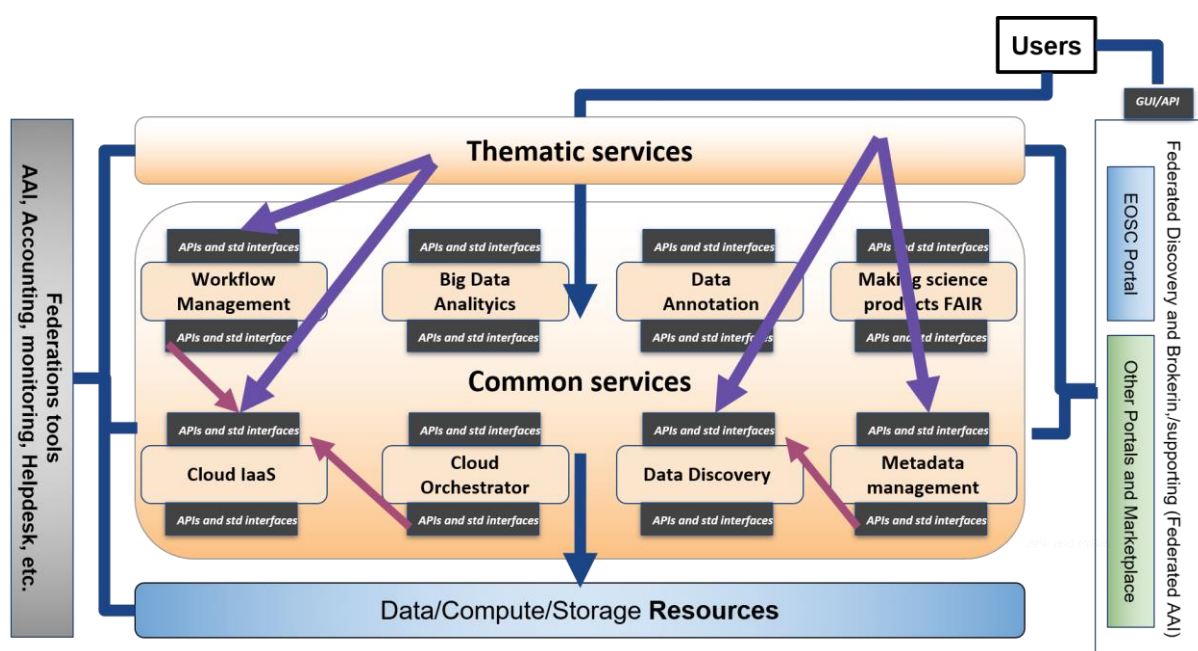


Figure 9. EOSC Technical Architecture. Interactions between thematic and common services.

5.2.3 EOSC Thematic services

As previously written, we want to apply the same process for Thematic services, to identify and create technical specifications for their building blocks. In this category, **a building block is a technical function that is discipline-oriented and that can be reused in multiple services within one thematic domain.**

Discipline oriented building blocks need to be identified and specified by experts of the related disciplines. Then, EOSC-hub will start the work of detailing this category with the communities participating in the project. However, community oriented projects need to be involved to further enhance this activity.

The following picture shows how the EOSC technical architecture will appear when the first set of building blocks (and the related technical specifications) for the Thematic Services will have been identified.

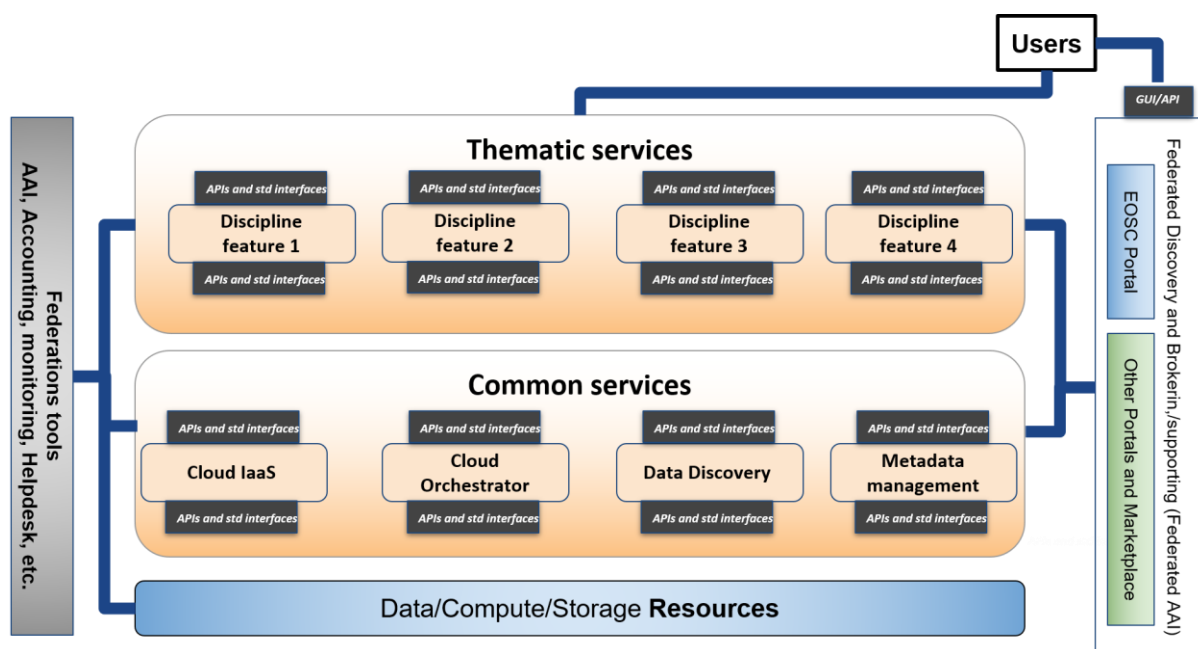


Figure 10. EOSC Technical Architecture. Building blocks for thematic services.

5.2.4 EOSC Portal

The EOSC Portal is intended to become the *single, coherent access channel to EOSC services at European level that meets researchers' needs for data sharing, management and computing* mentioned in European Commission Staff document introduced in section 2. It is currently being further enhanced and developed by a large collaboration that includes EOSC-hub, OpenAIRE Advance and key partners of the eInfraCentral project in the context of the EOSC Portal collaboration agreement (until December 2019). More information is available in the EOSC Portal concept paper.²⁷ The outcomes of this collaboration will be adopted by this work to technically specify the EOSC Portal. From 2020, EOSC-hub and OpenAIRE Advance are expected to continue the collaboration involving the new project that will be funded under the INFRAEOSC-06 “Enhancing the EOSC portal and connecting thematic clouds” call²⁸.

The following section is an extract of the architecture section of the EOSC Portal concept paper.

5.2.4.1 Architecture

The EOSC Portal concept paper describes the EOSC Portal architecture with a number of internal components and dependencies from external services.

²⁷ <https://wiki.eosc-hub.eu/display/EOSC/EOSC+Portal>

²⁸ <https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/infraeos-06-2019-2020>

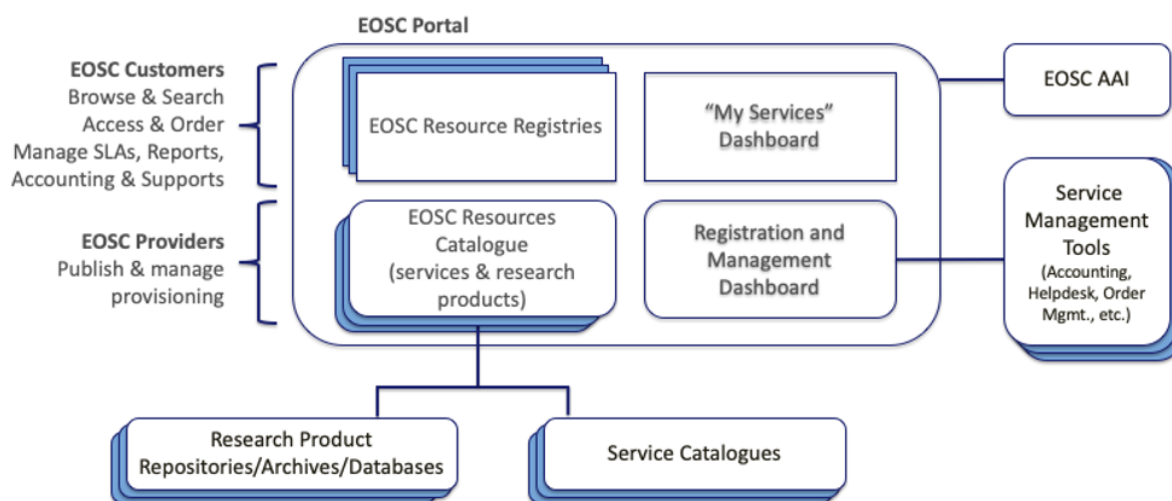


Figure 11. Internal components of the EOSC Portal and dependencies on external services

- **Internal components**

- **EOSC Resource Registry:** *from an EOSC Customer point of view, browsing, searching, access and ordering will be possible through the EOSC Resource Registry. According to the EOSCpilot Glossary 1.0²⁹, the EOSC Resource Registry provides the descriptions of live / ready-to-use EOSC Resources offered by the EOSC System. Resources will include services and scientific products that are produced by scientists, like data, software, publications, tools and experiments. These are published for discovery and reuse with metadata and links to other products via dedicated sources, e.g. repositories, archives, databases. To facilitate their discovery, cross-discipline or thematic metadata aggregators are today available and widely used by scientists. The EOSC Portal will integrate with scientific product catalogues capable of serving the needs of researchers from different disciplines. Dedicated Registries will be possible in order to present the EOSC offer to specific Users groups.*
- **EOSC Resource Catalogue:** *from a EOSC Provider point of view, the Catalogue comprises “the list of all live EOSC Resources that can be requested by EOSC System Users”. Resources are described by metadata that is either directly inputted after successful validation by the Provider, or through APIs [...].*
- The **My Services Dashboard** will allow EOSC users to manage the services they are ordered and accessed.
- The **Registration & Management Dashboard** will provide capabilities supporting the Provider in the on-boarding and validation procedure, and additional functions for integrated service management of the contributed Resources within EOSC. The Additional capabilities for the Providers willing to in that choose a high-level partnership.

- **External dependencies.** The EOSC Portal will interact with external services, namely:

²⁹ <https://eoscpilot.eu/eosc-glossary>

-
- *The **EOSC AAI service**, conforming to the AARC blueprint architecture and operational guidelines, supporting: (1) uniform representation of unique Users identifiers, (2) a standardised way of expressing group membership/role/information and Resource capabilities, (3) non-web browser based access, (4) delegation, (5) release of mandatory Users attributes according to the REFEDS research and scholarship entity category, (6) operational security, incident response and traceability - REFEDS Sirtfi, (7) privacy requirements for processing personal information following the GEANT Data Protection Code of Conduct, (8) rules and conditions that govern access to and use of Resources following the WISE Baseline Acceptable Use Policy, and (9) assurance information following the REFEDS assurance framework and IGTF/AARC assurance profiles.*
 - ***External catalogues, repositories, databases and archives**, providing metadata on services and other products (e.g. datasets, software, applications). Interoperability will be enforced with the adoption of the EOSC Catalogue Framework (see the concept paper for more information).*
 - ***External tools for service management** (accounting repositories, helpdesk, order management tools etc.), which will exchange ticket information, usage information and order information that are managed externally. Interoperability will be ensured through a Service Integration and Access Management interoperability framework that is being developed by the EOSC-hub project.*

5.2.5 Architecture governance

Different governance models can be envisaged per the EOSC service typologies embedded in the architecture. Indeed, the influence of the EOSC governance on defining technical specifications for the different service types can vary from fully authority for federation and access enabling services to simple endorsement of the specifications provided by the communities for thematic services, while a hybrid approach can be foreseen for common services.

Recommendations on the governance models to apply to the identified service categories are expected from the EOSC Architecture WG.

6 Use cases to drive the identification and the specification of the building blocks

The presented EOSC Technical Architecture foresees the definition of a technical specification for each of the building blocks identified per service category. It appears clear that the total number of building blocks can become quite big, and then a way to identify and focus on the most relevant functions is needed.

The natural way to prioritise the building blocks is taking into account the user requirements. For this reason, we are using as references the analysis done on a multitude of use cases in the context of the EOSCpilot and EOSC-hub projects.

All the sources of information we are taking into account are listed in the following table. This list will be extended at any time according to the suggestions we will receive from other relevant EOSC stakeholders.

Sources	Description	References
EOSCpilot Science Demonstrators	EOSCpilot selected fifteen science demonstrators, across different scientific domains with the purpose of providing insight on technical and policy needs, and cross-infrastructure integration requirements, and to get indications on how the EOSC Service portfolio should be structured. Some examples of communities supported in the context of this activity are: Photon and Neutron, EPOS, PanCancer, Fusion, WLCG, LOFAR, etc.	D5.6 Evaluation Report of service pilots ³⁰
EOSC-hub Thematic Services	Mature thematic services (TRL8 or TRL9) from large communities federating in the EOSC and integrating several generic, federation and access-enabling services. Involved	D7.2 First report on Thematic Service architecture and software integration ³¹

³⁰ <https://eoscpilot.eu/content/d56-evaluation-report-service-pilots>

³¹ <https://documents.egi.eu/document/3412>

	communities are: CLARIN, CMS, ENES, GEOSS, OpenCOASTs, WeNMR, DARIAH, LifeWatch and several Earth Observation services (including the CREODIAS DIAS platform).	
EOSC-hub Competence Centres	<p>Competence Centres design, integrate and disseminate new, community specific services and service platforms.</p> <p>Each Competence Centre (CC), fosters the use of advanced digital capabilities and resources of EOSC by early adopter research communities in order to support data- and computing-intensive science.</p> <p>Competence Centres are driven by well-established and mature research infrastructure or international scientific collaborations: ELIXIR, Fusion, Marine, EISCAT-3D, EPOS-ORFEUS, LOFAR, ICOS and Disaster Mitigation.</p>	D8.1 Report on progress, achievements and plans of the Competence Centres ³²
Use cases identified through the EOSC Portal	Since the launch of the EOSC Portal in November 2018, many communities ordered EOSC services through the Service Catalogue and Marketplace. These communities were supported by the EOSC-hub technical support activity (T10.3) and the analysis of their use cases	Community Requirements Database ³³

³² <https://documents.egi.eu/document/3485>

³³ <https://wiki.eosc-hub.eu/display/EOSC/Community+requirements+DB>

	where stored in the Community Requirement Database.	
--	---	--

Collected requirements are used to identify the most relevant building blocks but also to properly shape the related technical specifications. Indeed, user requirements give suggestions on the main functions a building block should deliver and on the most common integration scenarios. The latter would allow understanding which (standard/EOSC) interfaces a building block should implement to satisfy the needs of the largest number of use cases. The following figure shows hypothetical user requirements for the interoperability guidelines of various building blocks.

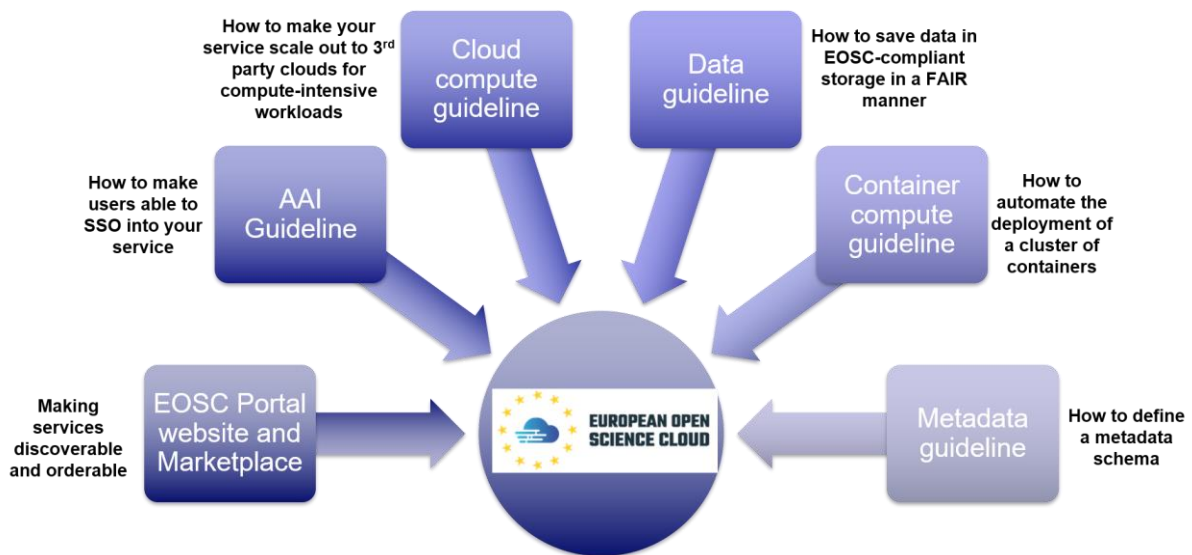


Figure 12. User requirements and interoperability guidelines for building blocks.

7 Relationship with the EOSC Architecture Working Group

As described in Section 2, the EOSC-hub work on the EOSC Technical Architecture is intended to become an important input for the activity of the EOSC Architecture Working Group and it has been shaped taking into account the WG mandate. As a result, the outputs of our work can become valuable for the WG and support it in achieving its objectives.

The following table shows the mapping between the Architecture WG sub-objectives and what EOSC-hub is expected to deliver.

EOSC WG objectives - The WG will describe and/or define:	Outcome of the EOSC-hub activity on the EOSC technical architecture
EOSC core services and their interfaces	<ul style="list-style-type: none"> • Definition of the EOSC Access Enabling and Federation services and interfaces
EOSC open source APIs for reuse by thematic services	<ul style="list-style-type: none"> • Interoperability guidelines for Common services (EOSC APIs and standards) • Interoperability guidelines for Thematic services (EOSC APIs and standards)
EOSC portal components and federated catalogues of service offerings	<ul style="list-style-type: none"> • Outcomes of the collaboration between EOSC-hub, OpenAIRE and key partners from eInfraCentral on the EOSC Portal design and development
The EOSC data description standards	<ul style="list-style-type: none"> • To be described in the technical specification of the metadata management building block
Standards and best practices necessary to ensure the evolution of EOSC and the widening of its user base to the industry and the public sectors	<ul style="list-style-type: none"> • Interoperability guidelines for Common services (EOSC APIs and standards) • Interoperability guidelines for Thematic services (EOSC APIs and standards)

EOSC-hub would like to establish a fruitful collaboration with the WG to further refine the proposed architecture taking into account requirements and suggestions from the largest possible set of service providers and user communities.

8 Technical specifications for Federation Services

This section presents some examples of technical specifications for Federation Services proposed by EOSC-hub.

8.1 AAI

Introduction

The EOSC AAI enables seamless access to research data and services in EOSC in a secure and user-friendly way.

High-level Service Architecture

The EOSC AAI follows the architectural and policy recommendations defined in the AARC project³⁴. As such, it enables interoperability across different SP-IdP-Proxy services, each of which acts as a bridge between the community-managed proxies (termed Community AAI) managing the researchers' identity and the generic services offered by Research Infrastructure and e-Infrastructures (termed R/e-Infrastructures or Infrastructures). This is the “community-first” approach to the AARC Blueprint Architecture³⁵, which enables researchers to sign in with their community identity via their Community AAI. A high-level view of the EOSC AAI is provided in the figure below.

³⁴ <https://aarc-community.org>

³⁵ <https://aarc-project.eu/guidelines/aarc-g045/>

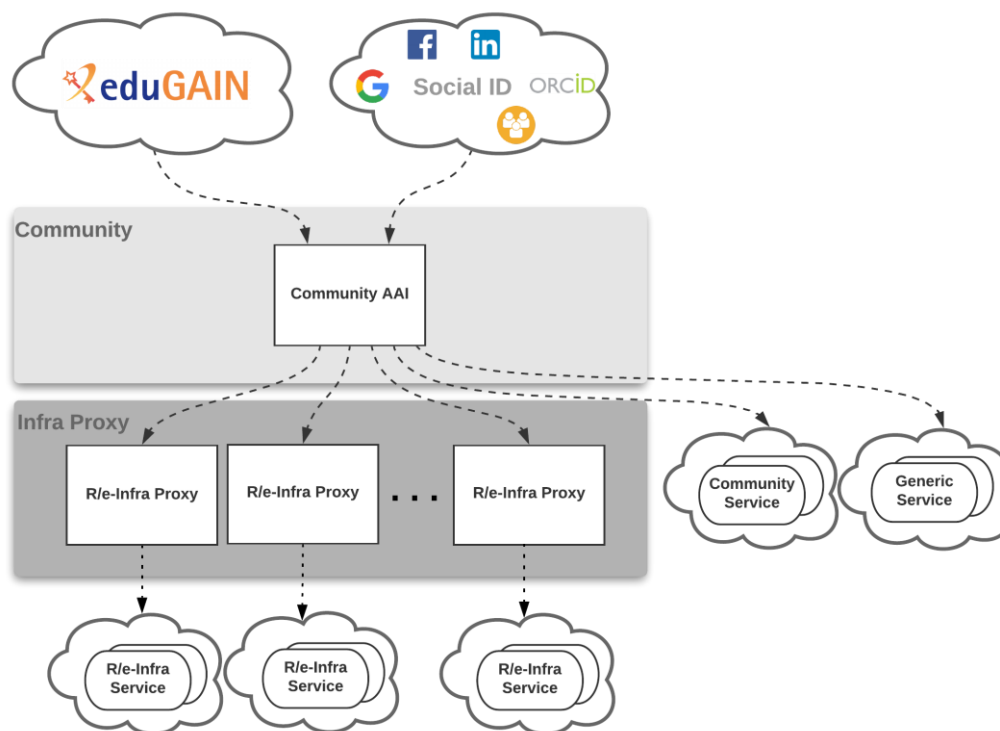


Figure 13. AARC Community first approach.

Community-specific services are connected to a single Community AAI, while Infrastructure Services are connected to a single Infrastructure Proxy. Lastly, generic services may be connected to more than one Community AAI. Each Community AAI in turn serves as a bridge between external identity providers and the proxies to the e-infrastructure services. Specifically, Community AAIs connect to eduGAIN as service providers but act as identity providers from the services point of view, thereby allowing users to use their credentials from their home organisations. Complementary to this, users without an account on a federated institutional Identity Provider are still able to use social media or other external authentication providers for accessing services.

Research communities can leverage the EOSC AAI services for managing their users and their respective roles and other authorisation-related information. At the same time, the adoption of standards and open technologies, including SAML 2.0, OpenID Connect, OAuth 2.0 and X.509v3, facilitates interoperability and integration with the existing AAIs of other e-Infrastructures and research communities.

Adopted standards

Standard	Short description	References
Security Assertion Markup Language (SAML) 2.0	OASIS standard for exchanging authentication and authorisation data between parties.	https://www.oasis-open.org/standards#samlv2.0
OAuth 2.0	Standard for authorisation that enables delegated access to server resources on behalf of a resource owner	"The OAuth 2.0 Authorization Framework", RFC 6749, https://www.rfc-editor.org/info/rfc6749
OpenID Connect 1.0	Identity layer on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner	"OpenID Connect Core 1.0", https://openid.net/specs/openid-connect-core-1_0.html
X.509	ITU-T standard for a public key infrastructure (PKI), also known as PKIX (PKI X509)	"Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, https://www.rfc-editor.org/info/rfc5280 "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile", RFC 3820, https://www.rfc-editor.org/info/rfc3820
Lightweight Directory Access Protocol (LDAP)	Provides access to distributed directory services that act in accordance with X.500 data and service models.	https://tools.ietf.org/html/rfc4511

Protocol/API	Short description	References
OAuth 2.0 Token Introspection	Protocol that allows authorised protected resources to query the authorisation server for determining the set of metadata for a given OAuth2 token, including its current validity.	https://tools.ietf.org/html/rfc7662

OAuth 2.0 Token Exchange	Protocol for requesting and obtaining security tokens from OAuth 2.0 authorization servers, including security tokens employing impersonation and delegation.	https://tools.ietf.org/id/draft-ietf-oauth-token-exchange-14.html
OAuth 2.0 Device Authorization Grant	Enables OAuth 2.0 clients on input-constrained devices to obtain user authorisation for accessing protected resources without using an on-device user-agent.	https://tools.ietf.org/html/draft-ietf-oauth-device-flow-15
System for Cross-domain Identity Management (SCIM) 2.0	Open API for managing identities	<p>SCIM: Core Schema , RFC7643, https://tools.ietf.org/html/rfc7643</p> <p>SCIM: Protocol, RFC7644, https://tools.ietf.org/html/rfc7644</p> <p>SCIM: Definitions, Overview, Concepts, and Requirements, RFC7642, https://tools.ietf.org/html/rfc7642</p>

Interoperability guidelines

Technical interoperability guidelines

- The attributes used to express user information should follow the REFEDS R&S attribute bundle, as defined in [\[REFEDS-R&S\]](#)
- VO/group membership and role information, which is typically used by relying parties for authorisation purposes, should be expressed according to [\[AARC-G002\]](#)
- Capabilities, which define the resources or child-resources a user is allowed to access, should be expressed according to [\[AARC-G027\]](#)
- Affiliation information, including (i) the user's affiliation within their Home Organisation, such as a university, research institution or private company, and (ii) affiliation within the Community, such as cross-organisation collaborations, should be expressed according to [\[AARC-G025\]](#)
- Assurance information used to express how much relying parties can trust the attribute assertions about the authenticating user should follow:
 - REFEDS Assurance framework (RAF) [\[RAF-version-1.0\]](#)
 - Guideline on the exchange of specific assurance information [\[AARC-G021\]](#)
 - Guideline for evaluating the combined assurance of linked identities [\[AARC-G031\]](#)

- Guideline Expression of REFEDS RAF assurance components for identities derived from social media accounts [[AARC-GO41](#)]
- Guidelines for expressing the freshness of affiliation information, as defined in [[AARC-G025](#)]
- OAuth2 Authorisation servers should be able to validate tokens issued by other trusted Authorisation servers. Extending existing flows, such as the OAuth2 Token Exchange flow [[OAuth2-Token-Exchange-draft](#)], will need to be considered for enabling the validation of such externally issued tokens.

Policy interoperability guidelines

- For the EOSC AAI, compliance with the GÉANT Data Protection Code of Conduct version 1 (DPCoCo-v1) [[DPCoCo-v1](#)] is implicit, since it reflects the Data Protection Directive and means compliance with applicable European rules (see [[AARC-G040](#)]). To explicitly declare compliance with DPCoCo-v1, the privacy notice of each EOSC AAI service should include a reference to DPCoCo-v1.
- The entities of the EOSC AAI registered with eduGAIN should meet the Sirtfi [[Sirtfi-v1.0](#)] requirements and express Sirtfi compliance in their metadata in order to facilitate coordinated response to security incidents across organisational boundaries.
- To reduce the burden on the users and increase the likelihood that they will read the AUP as they access resources from multiple service and resource providers, the EOSC AAI services should adopt the WISE Baseline AUP model [[WISE-AUP](#)].

Examples of solutions implementing this specification

AAI services:

- [B2ACCESS](#)
- [Check-in](#)
- [eduTEAMS](#)
- [INDIGO-IAM](#)

Identity and Access Management:

- [Perun](#)
- [Comanage](#)
- [HEXAA](#)

Token Translation Services:

- [WaTTS](#)
- [MasterPortal](#)
- [RCauth.eu](#)

Procedure to integrate a service with the EOSC Hub AAI

- [B2ACCESS](#)
- [Check-in](#)
- [eduTEAMS](#)
- [INDIGO-IAM](#)

- [Perun](#)
- [WaTTS](#)
- [MasterPortal](#)
- [RCauth.eu](#)

8.2 Accounting

Introduction

The Accounting system collects, aggregates, stores, and displays compute (serial and parallel jobs), storage, and cloud resource usage data collected from Resource Centres of the EOSC infrastructure.

Accounting information is gathered from distributed sensors into a central Accounting Repository where it is processed to generate summaries that are made available through the Accounting Portal. The Accounting Repository has a database backend, and needs to ensure the exchange of accounting information with peer e-Infrastructures. The Accounting Portal receives and stores the site, user, and user groups (e.g. VO) level aggregated summaries generated by the Accounting Repository and provides views via a web portal, for example, by grouping sites in a country on custom time intervals. The databases are organized into a CPU record database, a User record database, and a topology database.

High-level Service Architecture

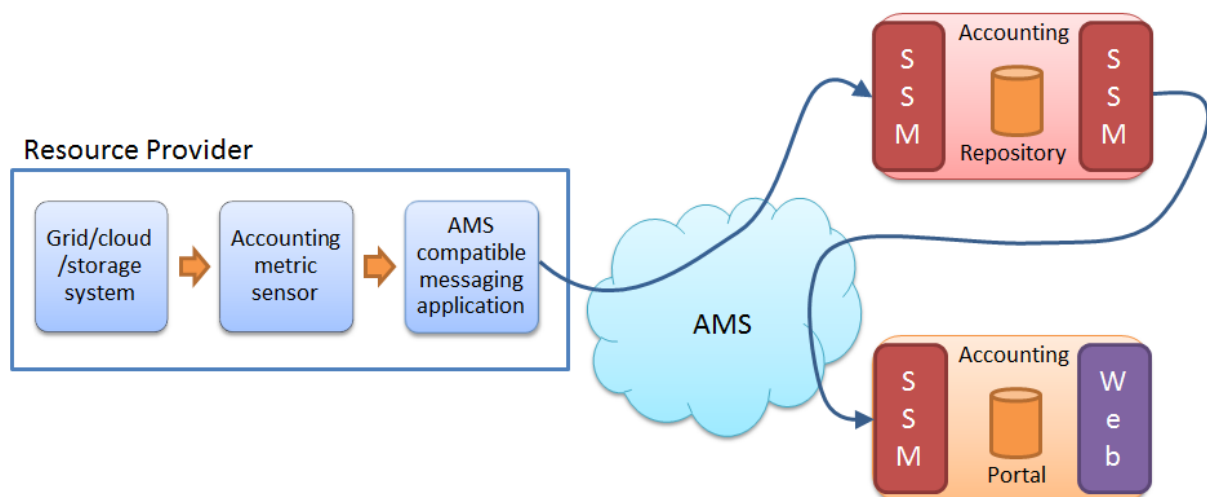


Figure 14. Accounting. High-level architecture.

Computing centres that are providing resources to the EOSC infrastructure have a sensor (a stand-alone script or program, or a built in function of their resource system) to gather accounting metrics and format them into one of the record formats that are accepted by the Accounting Repository. These are then sent using an ARGO Messaging Service (AMS, introduced later) compatible messaging application to the Accounting Repository, which stores and processes the data to produce aggregations that are then sent to the Accounting Portal for display.

Adopted standards

Standard	Short description	References
APEL Job Message	Standard used within WLCG and EGI for exchanging grid accounting metrics.	https://wiki.egi.eu/wiki/APEL/MessageFormat#Job_Records
Cloud Usage Record	Standard adopted by the EGI Federated Cloud for exchanging cloud accounting metrics.	https://wiki.egi.eu/wiki/Federated_Cloud_Accounting#Cloud_Usage_Record
OGF StAR	Open Grid Forum standard for Storage Accounting Records, used to exchange storage consumption data.	http://cds.cern.ch/record/1452920/files/GFD.201.pdf

Protocol/API	Short description	References
ARGO Messaging Service (AMS)	A Publish/Subscribe Service, which implements the Google PubSub protocol. It provides an HTTP API that enables Users/Systems to implement message oriented service using the Publish/Subscribe Model over plain HTTP.	http://argoeu.github.io/messaging/v1/

Interoperability guidelines

The Accounting Repository can accept records produced by any program so long as they are in the correct format (see standards for usage records above) and are sent via Argo Messaging Server. Resource providers need to be registered in a configuration management database (e.g. GOCDB³⁶) or be individually authorised to publish via AMS.

Examples of solutions implementing this specification

SSM³⁷ can be used as the messaging component. The APEL client³⁸ can be used for collecting usage metrics from grid batch systems. There are 3rd-party scripts available for certain cloud and storage systems.

³⁶ <https://goc.egi.eu/>

³⁷ <https://github.com/apel/ssm>

³⁸ <https://github.com/apel/apel>

Procedure to integrate a service with the EOSC Hub Accounting

For sending accounting records it is recommended to use SSM to handle the interfacing with AMS, but if it is desired to create a 3rd-party service then using the ARGO AMS Library³⁹ can be used to simplify the integration somewhat.

For creating accounting records, the Accounting Repository will accept records formatting to the standards above.

9 Technical specifications for Common Services

This section presents some examples of technical specifications for building blocks of Common Services proposed by EOSC-hub.

9.1 Cloud IaaS VM Management

Introduction

Services of Cloud IaaS VM Management provide on-demand API-based access to computing resources as Virtual Machines that can run user-defined arbitrary software (including operating systems and applications). Services in this category also allow management of block storage that can be associated to the VMs and network management to provide connectivity between VMs and external networks.

³⁹ <https://github.com/ARGOeu/argo-ams-library>

High-level Service Architecture

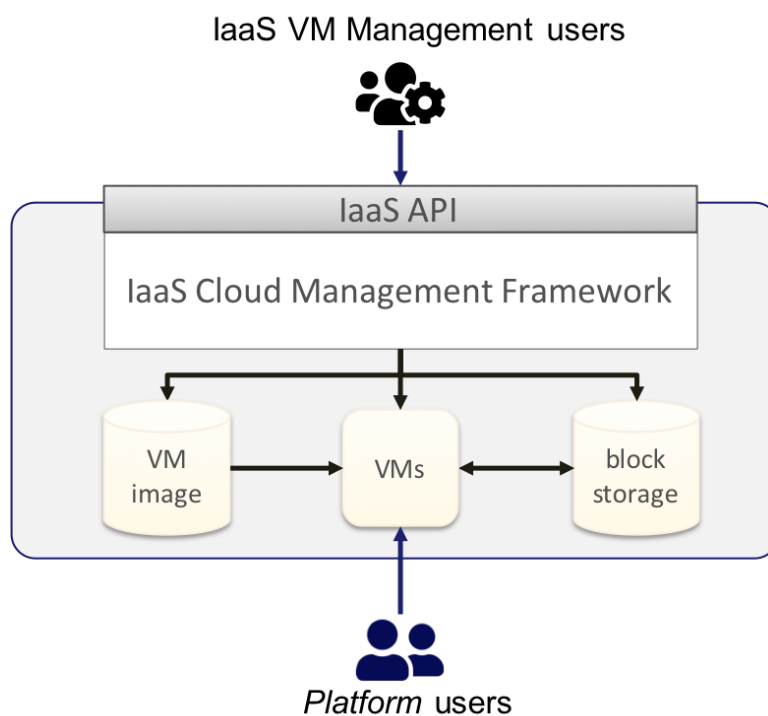


Figure 15. Cloud IaaS VM Management. High level architecture.

IaaS VM Management services allow users to manage VMs that are instantiated from VM images and can be associated with permanent block storage. The VMs can execute any kind of workload, including new services or platforms that are accessed by platform users, which may be different from the IaaS VM Management users that manage the IaaS resources.

Adopted standards

Standard	Short description	References
Open Virtualization Format (OVF)	Packaging format for software solutions based on virtual systems (VM image format)	OVF 2.1.1

Protocol/API	Short description	References
OpenStack	OpenStack is an Open Source cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and	OpenStack API

	provisioned through APIs with common authentication mechanisms.	
Amazon EC2/EBS/VPS & AWS VPN	Amazon Elastic Compute Cloud (EC2), Elastic Block Storage (EBS), Virtual Private Cloud (VPS) and AWS Virtual Private Network (AWS VPN) provide management of Virtual Machines and associated block storage and network features	AWS EC2 API
Azure Virtual Machines/Disks/VNet	IaaS VM management services from Microsoft Azure	Azure Virtual Machines API
Google Cloud Compute Engine	IaaS VM management service from Google Cloud Platform	Google Cloud Compute Engine API
OGF OCCI	Open community-lead specifications delivered through the Open Grid Forum. OCCI is a Protocol and API for all kinds of Management tasks, focused on IaaS	OCCI Specification
DMTF CIMI	The CIMI specification describes the model and protocol for management interactions between a cloud Infrastructure as a Service (IaaS) provider and the consumers of an IaaS service.	CIMI 2.0.0

Interoperability guidelines

Interoperable service in this category must:

- **Provide API access** for on-demand management of VMs and associated resources. Open and/or Standard APIs are preferred. Services that provide the capability to manage VMs through graphical dashboards but limit API access to users cannot be considered interoperable. See table above for a non-comprehensive list of APIs that may be supported by the service.

AAI interoperability

- Services should provide access to users authenticated with one of the EOSC-hub AAI federated identity protocols (OpenID Connect and/or SAML).

Orchestration interoperability

- Services should expose APIs that are supported by the IaaS Orchestrator services of EOSC-hub.

Federation interoperability:

- Services in this category that need to be federated into a cloud federation should provide API-based access to:
 - Management of VM images, i.e. allow creating (upload) and deleting VM images from which VMs can be instantiated.
 - Access usage information of individual VMs and block storage so accounting records can be generated for integration into the EOSC-hub central services.

Examples of solutions implementing this specification

EOSC-hub services:

- [EGI Cloud Compute](#)

OpenSource implementations:

- [OpenStack](#)
- [OpenNebula](#)

9.2 Platform as a Service (PaaS) solution

Introduction

A PaaS solution allows 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 templating language.

The PaaS layer features advanced federation and scheduling capabilities ensuring the transparent access to the different IaaS environments, both the “traditional” cloud management frameworks, like Openstack, OpenNebula, AWS and Azure, and the more innovative container orchestration platforms like Apache Mesos or Kubernetes.

The selection of the best cloud provider to fulfil the user request is performed considering criteria like the user’s SLAs, the services availability and the data location.

High-level Service Architecture

The high-level **reference architecture** is depicted in the diagram below.

The architecture can be broken down into the following main categories of components:

- Core services:
 - *API server*, providing REST endpoints to submit and handle the deployment requests;
 - *Workflow Engine*, that manages the deployment workflow;
 - *Message Bus*, providing a way of integrating services loosely and based on notifications (events).

- Plugins
 - *Cloud connectors, implementing the interfaces with the relevant Cloud Management Frameworks.*
 - *Container orchestration connectors, implementing the interfaces that abstract the interaction with the relevant container orchestration platforms, e.g. Mesos, Kubernetes.*
 - *HPC integration connectors, implementing the interfaces to interact with the HPC services; the envisaged interaction is based on REST APIs provided by gateway hosted by the HPC site, e.g. using QCG APIs or SLURM APIs.*
 - *Storage services connectors, implementing the interfaces to interact with the relevant storage management services; the interaction is based on REST APIs provided by the storage services themselves.*

Moreover, the following dependencies towards the Federation Services:

- EOSC AAI, to ensure federated access to the services and resources;
- AppDB-IS⁴⁰ or AMS (optional): information published by the sites can be used by the PaaS tools exploiting the already available collectors;
- EOSC Monitoring (optional): information about the health status of the services can be usefully exploited by the PaaS orchestrators in order to select the best sites for scheduling the user requests;
- Marketplace (optional): information collected in the Marketplace can be consumed by the PaaS tools.

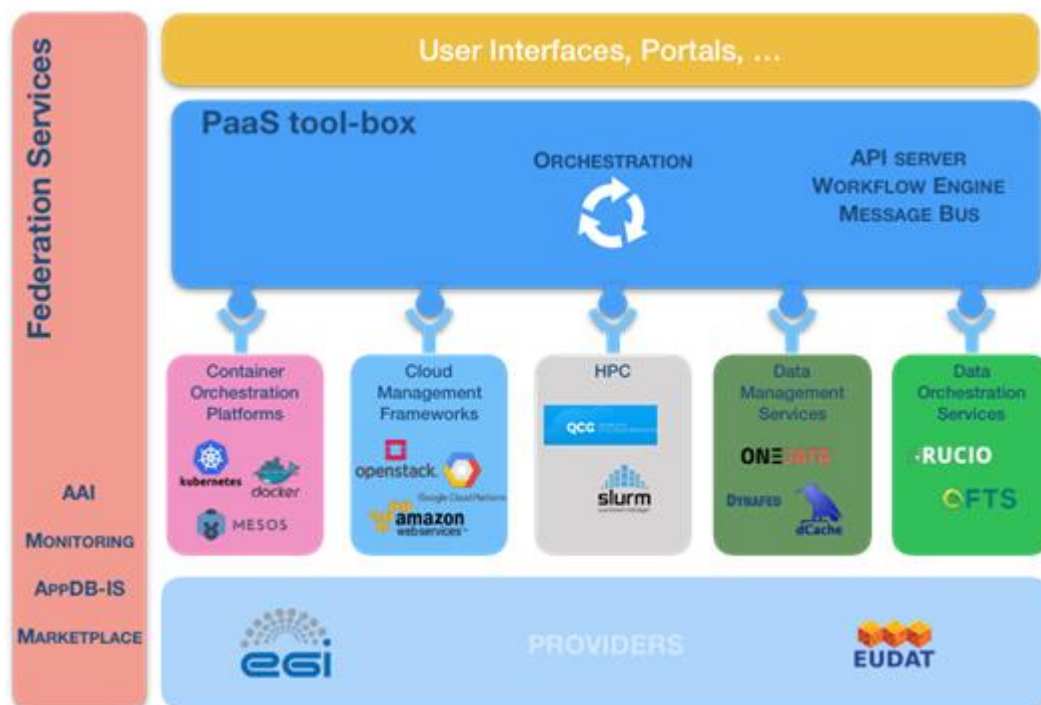


Figure 16. High level architecture of the PaaS solution

⁴⁰ <https://wiki.appdb.egi.eu/>

Adopted standards

Standard	Short description	References
TOSCA	OASIS open standard that defines the interoperable description of services and applications hosted on the cloud and elsewhere; including their components, relationships, dependencies, requirements, and capabilities, thereby enabling portability and automated management across cloud providers regardless of underlying platform or infrastructure.	http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/csprd01/TOSCA-Simple-Profile-YAML-v1.0-csprd01.html
Oauth2.0 Authorization Framework	The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.	https://tools.ietf.org/html/rfc6750
REST	REST, or REpresentational State Transfer, is an architectural style for providing information exchange interfaces between computer systems on the web, making it easier for systems to communicate with each other. REST-compliant systems, often called RESTful systems, are characterized by how they are stateless, resource-oriented and separate the concerns of client and server.	https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest

Interoperability guidelines

The adoption of the TOSCA standard can help reaching a good level of interoperability among different services in this area. However, this is a necessary but not sufficient condition since the

full interoperability would require the adoption of the same TOSCA custom types (in addition to the normative ones) and of the same REST API specifications.

Currently there is not an official standard for the PaaS orchestration APIs.

Examples of solutions implementing this specification

- INDIGO PaaS Orchestrator: <https://github.com/indigo-dc/orchestrator>
- Infrastructure Manager: <https://www.grycap.upv.es/im/index.php>

9.3 Metadata cataloguing and indexing

Introduction

Metadata Cataloguing and Indexing comprises the entire metadata ingestion workflow, i.e. 1) Metadata harvesting from community repositories 2) metadata mapping on common schema including curation and validation and 3) upload and indexing of metadata records in the metadata catalogue, to enable *Data Discovery and Access*.

High-level Service Architecture

The technical implementation of metadata cataloguing comprises usually three modules as shown in the figure below:

1. One (meta)data provider metadata must be available and harvestable using a known metadata schema and format and being harvestable using a standardised transfer protocol (e.g. OAI-PMH⁴¹).
2. One service provider site doing incremental metadata harvesting and metadata ingestion should be set up.
3. On service provider site normalisation, homogenisation and mapping of the specific community standards onto a generic, common and unified metadata schema is performed. The metadata mapping should be adapted to the needs of metadata provider and should include metadata validation and curation. This module is usually combined with (2)
4. Finally the mapped records are uploaded into the central metadata catalogue and indexed to allow faceted search in the discovery portal.

⁴¹ <https://www.openarchives.org/pmh/>

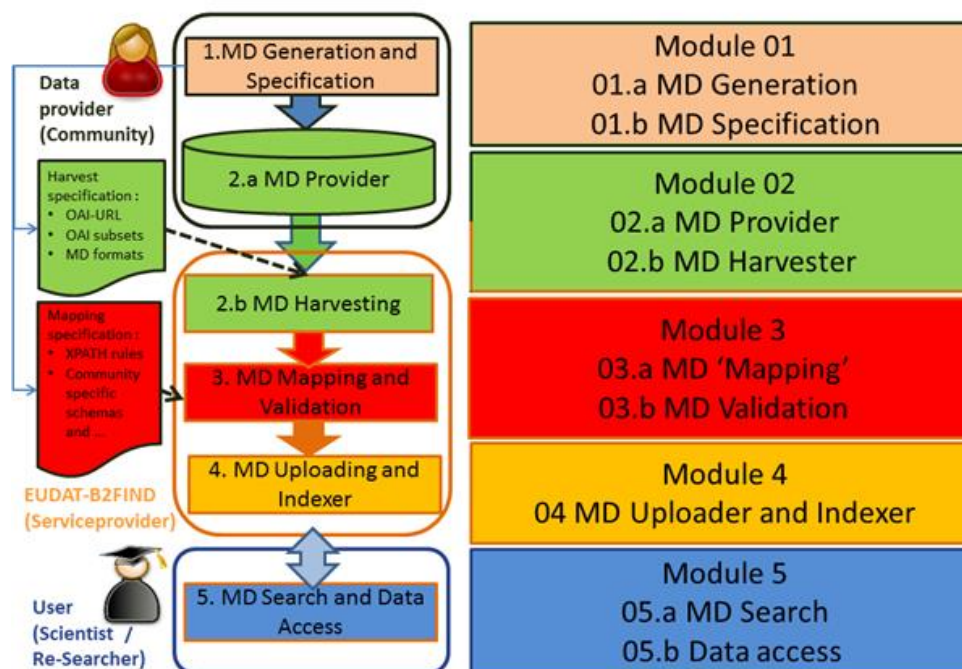


Figure 17. Metadata Cataloguing and Indexing. High level architecture.

Adopted standards

Standard	Short description	References
Community specific metadata schemas and standards	Central, cross-domain Metadata aggregators collect community specific formatted metadata. E.g. B2FIND supports harvesting of multiple data formats (as XML, JSON) and metadata schemas (e.g. MarcXML, DataCite, Dublin Core, ISO 19115, CMDI, DDI and others).	A list of some domain specific metadata standards can be found at http://b2find.eudat.eu/guidelines/providing.html#mdformats
DataCite Metadata Schema 4.1.	Common and widely used Metadata Schema, on which services like OpenAire and EUDAT- B2FIND are based.	https://schema.datacite.org/meta/kernel-4.1/ http://b2find.eudat.eu/guidelines/mapping.html#b2fmdschema
Controlled Vocabularies	ISO 639-1 codes as a standardized vocabulary used to identify languages	https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

	or the Taxonomy developed by EUDAT-B2FIND which specifies the different research disciplines present in B2FIND	https://cryptpad.fr/pad/#/1/edit/KDecbjauKCtZclOmZAbbWg/L4aEiGrzJISbRSXrFutOb0Cd/
--	--	---

Protocol/API	Short description	References
OAI-PMH	The Open Archives Initiative Protocol for Metadata Harvesting provides an application-independent interoperability framework to collect metadata from repositories.	http://www.openarchives.org/OAI/openarchivesprotocol.html
ResourceSync	The ResourceSync specification describes a synchronization framework for the web consisting of various capabilities that allow third-party systems to remain synchronized with a server's evolving resources.	http://www.openarchives.org/rs/1.1/resourcesync
misc. REST APIs	Full REST APIs are used to collect metadata formatted as JSON. E.g. the referenced REST API is used to 'harvest' from Herbadrop's repository	https://helpdesk.eudat.eu/Ticket/Attachment/122586/63597/RESTAPI_HowTo_SearchUserGuide_V3.pdf
CSW / OGC	Catalogue Service for the Web (CSW) is used to collect metadata from OGC catalogues	http://www.opengeospatial.org/standards/cat

Interoperability guidelines

In general the preconditions to publish metadata should be clearly described stated by the discovery service provider by 'Guidelines for data providers', as e.g. guidelines of OpenAire or of EUDAT-B2FIND (see <http://b2find.eudat.eu/guidelines>). This allows not only research communities, but also generic data storage repositories and metadata aggregators to make their data searchable in a simple way by following the guidelines.

Examples of solutions implementing this specification

Examples of cross-domain discovery services using this approach are:

- Google datasearch (<https://toolbox.google.com/datasetsearch>), which crawls mainly schema.org, but does not support any specific (meta)data curation and validation and does not care about open data access ('dark data', not really FAIR).

- EUDAT-B2FIND (<http://b2find.eudat.eu/>), the central indexer of EOSC-hub, provides an interdisciplinary discovery portal for research data with faceted search and comprises extensive metadata mapping, validation and curation in a FAIR manner.

Procedure to integrate a service with the EOSC Metadata Cataloguing and Indexing

To provide metadata to the Metadata Cataloguing and Indexing service the following preconditions must be fulfilled: 1) metadata provider must be set up (e.g. OAI-PMH provider), 2) Metadata must be implemented in a standardised format and agreed schema and made available and accessible for harvesting, also some mandatory fields (e.g a title and data identifier) must be provided. In the next stage refinement and enrichment of the metadata is done iteratively.

10 Conclusions and next steps

This deliverable presented the EOSC-hub proposal for the EOSC Technical Architecture that consists of the definition of a reference architecture, where all the EOSC main functions, interfaces, APIs and standards are identified. A common approach to identify key technical functions/building blocks for service category has been defined and started to be applied. As a result, several building blocks have already been identified and technical specifications are available for some of them.

As a next step, we have started a process to share our approach and collecting feedback. Initially a webinar was held where this work has been presented, then a formal feedback collection will start in the next few weeks and we are planning to organise a workshop by the end of this year involving the largest expected EOSC user groups.

Feedback is also needed on the technical specifications we are defining with the expertise available within the project. For every specification, we are intending to open a forum with technical experts from other initiatives with the double aim to improve the specification and find consensus around it.

Finally, we will liaise with the EOSC Architecture WG to continue this activity taking into account suggestions and requirements from the EOSC governance and the largest possible set of service providers and user communities.