EOSC Future

D4.1b

Back-Office design, functional and technical specifications

The EOSC Future project is co-funded by the European Union Horizon Programme call INFRAEOSC-03-2020, Grant Agreement number

Version 1.0
June 2023

D4.1b / Back-Office design, functional and technical specifications

Lead by EGI Foundation
Authored by Diego Scardaci (EGI Foundation), Mark Dietrich (EGI), Paolo Manghi (OpenAIRE), George Papastefanatos (Athena), Thanassis Mantes (Athena), Roksana Rozanska (Cyfronet), Dimitris Pierrakos (Athena), , Alessia Bardi (CNR), Kostas Koumantarous (GRNET), Pavel Weber (KIT), Themis Zamani (GRNET)

Reviewed by Raimundas Tuminauskas (POSZAN) & Athanasia Spiliotopoulou (JNP)

Dissemination Level of the Document

Public

Abstract

This document presents the final specification of the functionalities, interactions (data flows and APIs), and resulting architecture of the EOSC Core Back-Office. The documents extends and refines the functional architecture designed in D4.1a, as a part of the overall EOSC Platform, after an analysis of the main requirements gathered from the users of the back-office: providers of thematic or horizontal services and resources, portal or resource catalogue providers (thematic, regional, project-based, etc.), EOSC funders and other relevant stakeholders (e.g. the EC or the EOSC Association) and providers of the EOSC Portal front-office functionalities.

This final specification should be considered as a fundamental document to perform the handover towards the initiatives that will deliver and enhance the EOSC Platform after the end of the EOSC Future project (e.g. the EOSC Procurement and the project that will be awarded in the call HORIZON-INFRA-2023-EOSC-01-04).

Version History

Version	Date	Authors/Contributors	Description
V0.1	01/02/2023	Diego Scardaci (EGI Foundation), Mark Dietrich (EGI), Paolo Manghi (OpenAIRE)	First draft
V0.2	15/05/2023	Diego Scardaci (EGI Foundation), Mark Dietrich (EGI), Paolo Manghi (OpenAIRE), George Papastefanatos (Athena), Roksana Wilk (Cyfronet), Dimitris Pierrakos (Athena), Cyril L'Orphelin (CNRS), Alessia Bardi (CNR), Adrian Coveney (STFC), Kostas Koumantarous (GRNET), Pavel Weber (KIT), Themis Zamani (GRNET)	Full draft ready for internal review
V0.3	31/05/2023	Diego Scardaci (EGI Foundation), Mark Dietrich (EGI), Paolo Manghi (OpenAIRE), George Papastefanatos (Athena), Roksana Wilk (Cyfronet), Dimitris Pierrakos (Athena), Cyril L'Orphelin (CNRS), Alessia Bardi (CNR), Adrian Coveney (STFC), Kostas Koumantarous (GRNET), Pavel Weber (KIT), Themis Zamani (GRNET)	Full draft ready for external review
V0.4	05/06/2023	Diego Scardaci (EGI Foundation)	Final draft
V1.0	12/06/2023	Diego Scardaci (EGI Foundation), Paolo Manghi (OpenAIRE), Ron Dekker (TGB), Mike Chatzopoulos (ATHENA)	Final Version submitted to EC

Copyright Notice

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

Acronym	Definition
AAI	Authentication and Authorisation Infrastructure
AMS	Argo Messaging Service
API	Application Program Interface
A/R	Availability and Reliability
CMDB	Configuration Management Data Base
CRIS	Current Research Information System
CRM	Customer Relationship Management
CRUD	Create Read Update Delete
EC	European Commission
EPOT	EOSC Portal Onboarding Team
HDFS	Hadoop Distributed File System
HPC	High Processing Compute
ISNI	International Standard Name Identifier
OMS	Order Management System
ORCID	Open Researcher and Contributor ID
PID	Persistent Identifier
REST API	Representation State Transfer Application Program Interface
SLA	Service Level Agreement
SOMBO	Service Order Management Back-Office
UI	User Interface

1 Executive Summary

This document presents the final specification of the functionalities, interactions (data flows and APIs), and resulting architecture of the EOSC Core back-office, including interfaces to the front-office components to integrate with the EOSC Portal.

In this revision of the deliverable, the functional architecture for the EOSC Core back-office is presented embedded in the overall EOSC Platform architecture that has been developed by the project (TCB and technical WPs) leveraging a well-known methodology to describe software architecture, the C4 Model (see D3.3a, D3.2b and D3.3b). This architecture was designed after an analysis of the main requirements gathered from the users of the back-office: providers of thematic or horizontal services and resources, portal or resource catalogue providers (thematic, regional, project-based, etc.), EOSC funders and other relevant stakeholders (e.g. the EC or the EOSC Association) and EOSC Portal front-office functionalities. This analysis allowed us to identify a list of key features to be defined and the main EOSC Core back-office enabling functional components. These enabling functional components were mapped onto the items of EOSC Future's high-level technical roadmap which are relevant to the Core back-office. Their main features were also described.

At this stage of the project, all the identified functional components have been deployed into production adopting, extending and integrating key outputs of past relevant EOSC initiatives (e.g. EOSC-hub, OpenAIRE-Advance, EOSC Enhance) that acted as background components for the EOSC Core back-office functional components. Details on the functional architecture of each of the EOSC Core back-office components are provided in the deliverable.

Leveraging the functional architecture, a technical architecture was drawn, connecting such components through interfaces and APIs with the aim of creating a homogeneous and coherent EOSC Core back-office able to guarantee a uniform experience to its users. The technical roadmap that was defined for each background component in the first version of this deliverable has been almost completely implemented.

The specifications reported in this document are the final specifications for the WP4 design and development activities. They reflect the output of the continuous requirement gathering and feedback collection process we run during the project that allowed us to enhance the functional and technical architectures during the project's lifetime to satisfy the needs of a wider group of research communities.

This final specification should be considered as a fundamental document to perform the handover towards the initiatives that will deliver and enhance the EOSC Platform after the end of the EOSC Future project (e.g. the EOSC Procurement and the project that will be awarded in the call HORIZON-INFRA-2023-EOSC-01-04).

2 Introduction

The EOSC Core Back-Office serves EOSC providers and offers a series of functionalities to allow them to access, share and manage their EOSC resource. The EOSC Core Back-Office is a key part of the EOSC Core Platform.

This deliverable reports the output of the work to define the functional and technical specifications of the EOSC Core back-office. The intended audience of this document is made of EOSC technical architects and service providers of the EOSC Core services, other EOSC providers and the European Commission. It is organised as follows:

- Section 3 presents the requirements for the EOSC Core back-office components gathered from its main stakeholders and the project's high-level technical roadmap. From these requirements, a list of enabling functional components is derived.
- Section 4 describes the EOSC Core back office's functional architecture, identifying the background components (delivered by past projects and initiatives) that support the development of the enabling functional components and outlining how they should collaborate. It also provides details for each functional component on how it has evolved during the project lifetime.
- Section 5 describes the EOSC Core back office's technical architecture. For each component of the EOSC Core back office, a technical architecture is presented.
- Section 6 draws conclusions and next steps.

3 Requirements for the EOSC Core Back-Office components and Functional Components

3.1 Stakeholders Requirements and Enabling Functional Components

The EOSC Core for back-office is mainly intended to serve EOSC providers, facilitate access to EOSC Core services and allows providers to share and manage their EOSC resources. It also offers a series of enabling functionalities for the EOSC Portal front-office components.

The main stakeholders that will use the EOSC Core back-office components are:

- EOSC providers of thematic services and resources (e.g. clusters and research infrastructures);
- EOSC providers of horizontal services and resources (e-infrastructures, but also clusters and research infrastructures);
- Providers of thematic and regional portals/catalogues willing to offer their services and resources through EOSC;
- Other EC-funded projects (e.g. the EOSC-07 projects) and initiatives delivering services and resources for EOSC;
- EOSC funders and other relevant stakeholder (e.g. the EC or the EOSC Association) to get information about the usage and uptake of EOSC and its services;
- Providers of the EOSC Portal front-office functionalities.

The requirements of these stakeholders for the EOSC Core for back-office are diverse. The most relevant, collected from EOSC Future and past projects and initiatives (e.g. EOSC-Hub, OpenAIRE-Advance, EOSC Enhance), are listed in Table 3-1 [1] [2] [3] [4] [5] . They are considered features that should be offered by the EOSC Core back-office and we named them accordingly.

Table 3-1: EOSC Core back-office features from stakeholder requirements

Feature name	Requirement description
Resource onboarding	Register and manage their services and resources in the EOSC Resource Catalogue
Portal/resource catalogue onboarding	Connect their resource catalogues to the EOSC Resource Catalogue
EOSC Order Management	Enable services to be accessed and ordered by EOSC users
Composability	Ability to group any EOSC resources (services, data, etc.) into ready-to-use solutions, order, and execute them seamlessly as a whole
Interoperability	Ability to identify EOSC resources able to work together
Open Science statistics	Get information about research impacts or trends for countries, institutions, research infrastructures (e.g. number of open access publications/datasets and the number of repositories and journals)
EOSC statistics	Get information about EOSC usage and EOSC resource ordering and access
EOSC service accounting	Get indicators for the usage of EOSC Core and Exchange services
EOSC research product usage statistics	Get indicators about downloads and views of EOSC research products (publications, datasets, or software)
EOSC support	Manage user requests about EOSC Core and Exchange services
EOSC monitoring	Get near real-time status of the EOSC services

However, not all these requirements are relevant to all the identified stakeholders and each requirement can be relevant to different stakeholders for various reasons. The following table crosses the features described above with the EOSC Core back-office stakeholder groups, providing more insights about how each feature will be used by each stakeholder.

The stakeholders presented above were further grouped in four sets:

- EOSC Providers: thematic and horizontal
- Providers of portals or resource catalogues: thematic, regional, project-based portals, etc.

- EOSC Funders and other relevant stakeholders
- Providers of EOSC Portal Front-Office functionalities

Table 3-2: EOSC Core Back-Office requirements per stakeholders

Stakeholders / Features	EOSC Providers (thematic + horizontal)	Providers of portals or resource catalogues providers (thematic, regional, project based, etc.)	EOSC Funders and other relevant stakeholders	Providers of EOSC Portal Front-Office functionalities
Resource Onboarding	Onboarding and management of their EOSC resources			Publishing the resources in the EOSC Portal
Portal/Catalogue Onboarding		Onboarding and management of their resource catalogue		Publishing the resources in the catalogues in the EOSC Portal
EOSC Order Management	Making their services orderable through EOSC		access to the list of most popular services in order management	Exposing ordering functions for EOSC resources
Composability	Fostering the combined usage of their services with other EOSC resources			Enriching EOSC resource descriptions with attributes to facilitate and, when possible, orchestrate the integration with other EOSC resources
Interoperability	Declare compatibility with one or more Interoperability Guidelines		Identify standards/interfaces/best practices widely used in EOSC	Enriching EOSC resource descriptions with attributes to facilitate the identification of interoperable EOSC resources
Open Science Statistics	Getting open science statistics about the EOSC resources they provide		Measurements of the uptake and impact of EOSC	Enriching EOSC resource descriptions in the portal with open science statistics
EOSC statistics	Getting information about the number of accesses and orders to their resources		Getting information about EOSC usage and the most-used EOSC resources	Enriching EOSC resource descriptions in the portal with statistics about accesses and orders
EOSC service accounting	Getting specific indicators for each of their EOSC resources to have a measurement of their usage		Getting specific indicators for each of their EOSC resources to have a measurement of their usage and an estimation of the costs of providing these resources	Enriching EOSC resource descriptions in the portal with specific indicators per resource
EOSC research product usage statistics	Getting usage statistics on their EOSC research products		Getting EOSC research product usage statistics to have a measurement of the EOSC impact	Enriching EOSC research product descriptions with usage statistics
EOSC support	Managing support requests received via EOSC			Providing an access point for support in the EOSC Portal
EOSC monitoring	Checking the status of their resources and being alerted in case of issues		Getting an overall view of the quality of the resources offered via EOSC	Showing quality measurements of the EOSC resources to EOSC users

The requirements outlined above, together with EOSC Future's high-level technical roadmap (presented in the section 3.2) drove the design and the implementation of the EOSC Core back-office presented in this document.

As a next step towards defining the EOSC Core back-office functional specifications, the features identified above were grouped into a set of enabling functional components listed in the following table. An additional

functional component was added to allow asynchronous communication between different EOSC Core back-office components. This is fundamental to enable the coordination and exchange of information between them.

The next section of this document will present the EOSC Core back-office components' functional architecture for these enabling components.

Table 3-3: EOSC Core back-office enabling functional components and supported features

Enabling Functional Components	Supported Features
EOSC resource catalogue	Resource onboarding Portal/Catalogue onboarding
EOSC service order management & composition	EOSC order management Composability
EOSC Interoperability Framework Registry	Interoperability
EOSC and open science statistics	Open science statistics EOSC statistics EOSC service accounting EOSC research product usage statistics
EOSC helpdesk	EOSC support
EOSC monitoring	EOSC monitoring
EOSC messaging	Coordination and exchange of information between EOSC Core back-office components

3.2 Requirements from the High-level Technical Roadmap

Technical developments in EOSC Future have been driven by the project's high-level technical roadmap. This roadmap was built taking into account requirements collected by several research communities, e-infrastructures and other relevant stakeholders as well as a common vision of EOSC that emerged from the joint effort of the EOSC Architecture WG and past EOSC projects (EOSC Enhance, EOSC-hub, OpenAIRE-Advance, etc.). The roadmap was organised into areas with a focus on providing an improved user experience to research communities in Europe. The roadmap has evolved during the project lifetime according to the new requirements that have been gathered from users that started to use the EOSC Platform.

This section presents the items of the project's high-level technical roadmap that are related to the EOSC Core back-office components (WP4) and how these have been translated into design choices and implementation activities for the back-office components. In particular, each element of the technical roadmap that is relevant for EOSC Core back-office was linked to one or more of the enabling functional components identified in the previous section. These enabling functional components are expected to deliver the related roadmap item.

The output of this analysis is described in the table below. It drove the functional and technical specifications described in the next sections of this document.

Table 3-4: EOSC Future technical roadmap and EOSC Core back-office enabling functional components

Area	EOSC Core back-office functional capabilities for EOSC Users			EOSC Core back-office enabling functional components
	M6	M18	M30	
User Experience - Resource Sharing and Discovery	Researchers can see and reach all thematic and regional portals from the EOSC Portal. Researchers can see services from the thematic clusters through EOSC.	A researcher searching the EOSC Portal sees a comprehensive set of resources from multiple communities and clusters. Usage statistics for datasets (views, downloads) will be collected and made available. Researchers using resources through EOSC will have common elements such as AAI, support, monitoring, and accounting. Researchers using a thematic portal see resources (services, data) pulled in from the central EOSC registry.	Researchers can gauge the quality and suitability of resources based on usage statistics and feedback from other services.	EOSC resource catalogue EOSC statistics and Open Science statistics EOSC helpdesk EOSC monitoring
User Experience - Resource Allocation	Researchers can request resources offered by INFRAEOSC-07 projects through the EOSC Marketplace.	Researchers requesting resources through the EOSC Marketplace portal can request access to EC-funded resources.	Requesting resources through EOSC includes the possibility to access commercial or centrally funded resources.	EOSC resource catalogue EOSC service order management & composition
User Experience - Resource Composability	Researchers can see example cases of complex workflows using multiple resource providers. Researchers can see a rich range of horizontal resources and request access to them.	Researchers can orchestrate data analysis on computing resources provided by multiple e-Infrastructure resource providers and transfer back the output to their storage system.	Researchers can compare and select resources based on how easy they are to compose and connect to.	EOSC resource catalogue EOSC service order management & composition EOSC Interoperability Framework Registry
EOSC-Exchange	Resources onboarded through prior projects (e.g. EOSC-hub, EOSC Enhance, eInfraCentral) remain available. Resource requests are passed to the provider.	EOSC-Exchange highlights which services can be easily integrated or composed. The onboarding process is extended to include validation of data sources to align to community (FAIR) metadata guidelines.	Resource requests integrated with procurement as well as provisioning functions.	EOSC resource catalogue EOSC service order management & composition EOSC Interoperability Framework Registry

		Resource requests can lead to automatic provisioning of resources.		
EOSC-Core	Provider portal & resource registry Allows direct onboarding (web + API) but as transfer of provider and resource records from other registries and vice-versa.	Provider portal & resource registry Increased automatic validation tools and automatic flagging of resources which are likely to require review.	Provider portal & resource registry Built-in or integrated management/workflow engine to support management of applications, review of records, auditing and quality control as well as automatic provider communication.	EOSC resource catalogue
	Monitoring & accounting Allows checking of services based on availability of their web pages/endpoints. Can support better integration via specific metrics. Accounting supports usage tracking to support virtual access reimbursement by the EC.	Monitoring & accounting Automated/self-service integration of monitoring probes and metrics offered to providers. Monitoring can track availability, reliability and accounting of usage based on parameters in provider and resource profiles (location, sector, organisation type).	Monitoring & accounting Automated monitoring includes automated thresholds, raising issues or alarms in the service management system based on results. Accounting is related to capacity data offered by providers.	EOSC monitoring EOSC statistics and open science statistics
	Helpdesk Helpdesk covers core services. Customisation based on new/updated technology and connected to the service management system.	Helpdesk Helpdesk covers core services and can redirect tickets to providers who have their own helpdesk. Helpdesk can be tested by providers who wish to use it.	Helpdesk Helpdesk-as-a-service available as optional add-on during onboarding. Integrated with central helpdesk functions.	EOSC Helpdesk
	Order processing Order management is set up and includes integration of providers. Already used for services inherited from EOSC-hub and others that previously enabled it.	Order processing Order management can deploy and provide resources from selected providers, including from 07 projects.	Order processing Order processing for thematic services can bundle orders for horizontal or basic services needed to deliver the thematic service and include deployment and provisioning.	EOSC service order management & composition
	EOSC Portal metrics dashboard The dashboard provides statistics about services and access requests and is fully integrated with the EOSC Portal.	EOSC Portal metrics dashboard The dashboard is enriched with additional information on EOSC resource providers and on activities of researchers in EOSC.	EOSC Portal metrics dashboard The dashboard provides statistics related to the combined/integrated usage of EOSC resources.	EOSC statistics and open science statistics

4 Functional Architecture

The design of the Functional Architecture for the EOSC Core back-office took into account the requirements described in the previous section as well as the current EOSC landscape. Components developed by past initiatives were selected to fulfil the EOSC Core back-office enabling functional components identified by the requirements, and evolution roadmaps were defined for each of these. Furthermore, integration activities were planned to create a unified EOSC Core Back-office platform able to guarantee a uniform experience to its users.

4.1 Background Components and Overall Functional Architecture

A thorough analysis of the landscape and of the main outputs of past relevant EOSC initiatives, done during the proposal preparation and in the first months of the project, allowed the identification of a set of components/services/tools that could properly act as background components for the EOSC Core back-office enabling functional components identified in section 3. Such background components already covered part of the stakeholder requirements while they needed extensions to satisfy other requirements.

In collaboration with the other technical WPs (WP3, WP5, WP7) and under the coordination of the TCB, WP4 contributed to the development of the overall architecture of the EOSC Platform leveraging a well-known methodology to describe software architecture, the C4 Model (see D3.3a, D3.2b and D3.3b). As a result of this effort, an overall functional architecture of the EOSC Platform has been developed (see Figure 4.1).

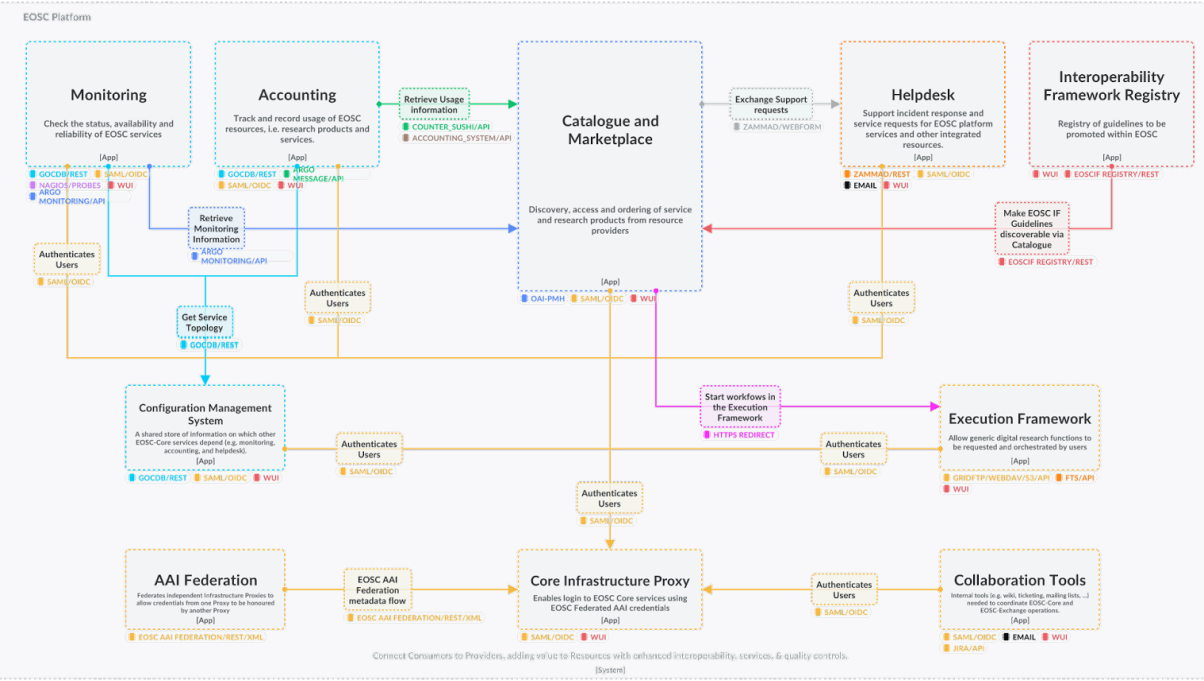

Figure 4.1: EOSC Platform functional architecture embedding EOSC Core back-office components.

This architecture embeds all the EOSC Core back-office components and other components (e.g. the Front-end and the MP) that all together composes the EOSC Platform.

The EOSC Core back-office functional architecture has been affected by this overall work on the architecture. In particular the EOSC Platform Architecture:

- introduced a hierarchy of the EOSC Platform components with the introduction of the *Container* category. An *EOSC Platform Container* includes multiple components that are working together to implement use cases;
- revised the name of the different components to guarantee uniformity in the whole EOSC Platform.

Table 4-1 shows how the EOSC Core back-office enabling functional components, identified in the previous section, are mapped to related EOSC Platform Containers and components. It also provides a short description of what each functional component is expected to deliver in terms of main features.

Table 4-1: EOSC Core back-office enabling functional components: functional requirements and background components

EOSC Platform Containers (see Figure 4.1)	**EOSC Platform Components**	**Enabling Functional Components**	**High-level Functional Requirements**
Catalogue and Marketplace	• Service Catalogue • Service Provider Dashboard • Research Product Catalogue • Research Products Provider Dashboard	EOSC resource catalogue	The EOSC resource catalogue maintains an overall list of EOSC resources (providers, services, data sources, research products) to enable a global discovery.
Catalogue and Marketplace	• Order Management System	EOSC service order management & composition	The Service order management & composition function enables resource ordering and management and facilitate the resource composition. It also offers instruments to facilitate and, when possible, automate the integration of EOSC resources to enable composition.
Interoperability Framework Registry	• Interoperability Framework Registry	EOSC Interoperability Framework Registry	Registry of Interoperability Guidelines is operational allowing the Guidelines to be managed by their owners. Registry allows a provider to declare the compliance of its resources with one or more Interoperability Guidelines present in the Registry.
Accounting	• Accounting for Services • Accounting for Research Products	EOSC and open science statistics	Various set of statistics have to be implemented: • Open science statistics, including research impact or trends for countries, institutions, research infrastructures (e.g. number of open access publications/datasets) • EOSC statistics, dashboards with statistics of usage of the EOSC Portal and service orders • EOSC service accounting, indicators about the usage of EOSC Core and Exchange services in order to transparently report service costs to the European Commission • EOSC research product usage statistics, indicators collected from EOSC data source services about downloads and views of research products (publications, datasets, or software)
Helpdesk	• Helpdesk	EOSC helpdesk	The EOSC Helpdesk works as a unified ticketing system by managing the requests related to different Core services. It can be also provided as a service for EOSC Exchange service providers through three main usage scenarios: • Direct usage: providers use dedicated support units in the EOSC helpdesk • Ticket redirection: the EOSC Helpdesk redirects the entry request for the specific service to a mailing list or 2nd-

			level ticketing system managed by the providers. • Full integration: tickets could seamlessly propagate to the dedicated helpdesk after submission in the EOSC Helpdesk.
Monitoring	• Monitoring	EOSC monitoring	It monitors services by emulating typical user scenarios which allows them to infer the quality of service the actual user gets. As a result, it offers near real-time status updates which allow both end-users and site admins to have an overview of the service offered at any given time. EOSC monitoring allows to quickly identify and correlate problems before they affect end-users and ultimately the productivity of the services, the infrastructure and finally the organisation.
Messaging	• Messaging	EOSC messaging	A real-time messaging service that enables asynchronous communication between the different components of the EOSC Core Back-Office.

However, each of these functional components is a part of the overall EOSC platform. All these elements have been properly integrated and are working in a coordinated manner to provide a homogeneous and coherent experience to users of the EOSC platform.

The diagram in Figure 4.1 presents the overall functional EOSC Platform architecture highlighting the connections and interfaces between each component. Details about the interactions between Core back-office components are presented later in this document.

4.2 EOSC Resource Catalogue

The EOSC Resource Catalogue serves as the central platform of the EOSC Future ecosystem, providing functionalities for resource registration, maintenance, administration, and sharing from diverse providers. The Catalogue serves as a reference point for all EOSC Future components, facilitating the searchability and accessibility of the resources for both researchers and end-users. Specifically, as shown in Figure 4.2, it contains metadata about services, data sources, service catalogues, training resources, and research products, along with semantic relationships between them, highlighting the nature of the connection between services and products (e.g., hosted By, generated By, etc.). The types of EOSC entities and resources are defined by the EOSC Resource Description Framework Interoperability Framework[1] (WP3), based on the Provider & Resource Description Template, and Research Products, aligned with the OpenAIRE Guidelines.

[1] https://wiki.eoscfuture.eu/display/PUBLIC/EOSC+Portal+Profiles

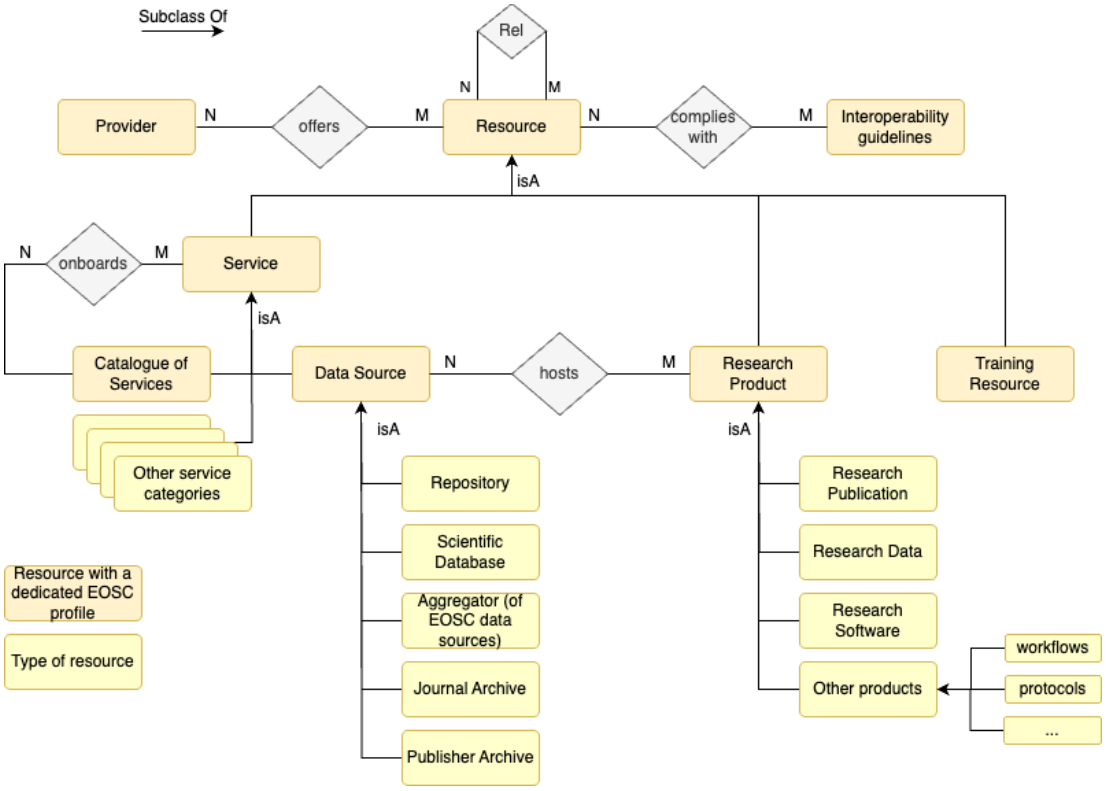

Figure 4.2: The EOSC Resource Catalogue Data Model

The EOSC Resource Catalogue is a confluence of the resources onboarded by **EOSC Providers** in the **EOSC Service Catalogue**, the **EOSC Research Product Catalogue**, via dedicated **EOSC Provider Portals**. Different kinds of resources require different processes of onboarding, based on specific Rules of Participation, which in turn require curators with different skills and backgrounds. In particular the EOSC Research Product Catalogue contains metadata of EOSC research products hosted into EOSC Data Sources and is stored as part of the EOSC Research Graph (aka, OpenAIRE Graph).

As a combination of such components, the EOSC Resource Catalogue should provide functionality for:

- Providers' registration to the EOSC to become eligible to onboard resources;
- Providers' onboarding of their services/research products;
- EOSC Portal Onboarding Team (EPOT) members' management of the onboarding process, managing the catalogue of providers and services and auditing the validity of the catalogue entries;
- Adding entire regional or thematic catalogues to the EOSC ecosystem by providers of catalogues;
- APIs for searching, browsing, and navigating the EOSC Resources and Providers' meta-information.

The following sections describe more in detail the EOSC Resource Catalogue components in support of resource onboarding and resource discovery, navigation, and access.

4.2.1 EOSC Service Catalogue

The onboarding of EOSC service resources requires a registration and validation procedure via the EOSC Service Providers Portal and the acquisition of service profiles into the EOSC Service catalogue component.

The EOSC Service Provider Dashboard: the portal enables the front-end functionality for the registration of EOSC providers, organisations that publish their resources via the EOSC catalogue. It offers capabilities to onboard and manage EOSC resources. It also offers the provider dashboard, where representatives from provider organisations have a detailed view of their offerings in the EOSC Portal as well as various usage statistics on their resources. Finally, it offers members of the EOSC Portal Onboarding Team (EPOT) the functionality to manage EOSC portal catalogue entries, i.e. manage the onboarding process of providers that apply to list their resources in the Portal, audit the onboarded resources, etc.

The **EOSC Service Registry:** the component offers the underlying storage functionality and the interoperability tools for the programmatic access, registration, manage (CRUD) of providers, services and catalogues. It also offers the necessary API functionality for the interoperability of service catalogues from individual providers or aggregators (e.g. thematic or regional catalogues) with the EOSC portal. Modifications to the entities maintained in the Service Catalogue are synchronised with the EOSC Research Graph, which aggregates the entire set of entities and their relationships in the EOSC Portal.

The list of functions is listed in Table 4-2. All these functions together allow the EOSC Resource Catalogue to satisfy the user requirements and to implement the related items of the technical roadmap.

Table 4-2: EOSC provider portal and service registry functions: onboarding and management

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
API and UI to onboard/manage EOSC providers	Providers can register to EOSC to become eligible for the onboarding of resources	Building an EOSC listing of eligible providers to ensure controlled onboarding of services	EOSC Service Registry EOSC Provider Portal
API and UI to onboard EOSC services	EOSC providers can onboard their services into the EOSC service catalogue: the action can be performed via APIs (service catalogues synchronisation framework) or via UIs by provider representatives.	Populating the EOSC service catalogue and EOSC-RG to deliver an up-to-date map of EOSC resources	EOSC Service Registry
API and UI to onboard EOSC research products	EOSC providers can onboard their research products into the EOSC Catalogue: the action can be performed by providers of data source services, and takes place via data sources' APIs	Populating the EOSC-RG to deliver an up-to-date map of EOSC resources	OpenAIRE research graph OpenAIRE PROVIDE
API and UI to onboard training resources	EOSC providers can onboard their training resources into the EOSC service catalogue: the action can be performed via APIs (service catalogues synchronisation framework) or via UIs by provider representatives.	Populating the EOSC service catalogue and EOSC-RG to deliver an up-to-date map of EOSC resources	EOSC Service Registry EOSC Provider Portal
API and UI to link interoperability guidelines to resources	EOSC providers can link their resources with interoperability guidelines from the EOSC Interoperability Registry through the EOSC service catalogue: the action can be performed via APIs (service catalogues synchronisation framework) or via UIs by provider representatives.	Building a relation between EOSC onboarded resources and interoperability guidelines	EOSC Interoperability Registry EOSC Service Registry EOSC Provider Portal
Minting PIDs for services	Service providers registering a service can optionally request the minting of a PID for the service	Provision of PIDs for EOSC services	EOSC Service Registry
UI for EOSC Portal onboarding team (EPOT) members to audit onboarding of services and providers	EOSC Portal onboarding team (EPOT) members can manage the onboarding process (approve or reject an application), manage the catalogue of providers and services and audit the validity of the catalogue entries.	Improving the quality of the catalogue entries and applying the rules of participation in new applications	EOSC Provider Portal
UI for providers to view and manage the services in the EOSC Portal	Service providers can view the list of services registered in the EOSC portal and perform a variety of actions such as activate,	Enables the delivery of an up-to-date map of EOSC resources	EOSC Provider Portal

	deactivate, view usage statistics, etc.		
APIs for the onboarding of service profiles from local service catalogues (from RIs, clusters, countries, etc.)	Catalogue providers can synchronise their catalogues (or a subset) with the EOSC Service Catalogue publishing in one shot multiple services into EOSC	Enables the publication into EOSC of multiple services registered in third-party catalogues	EOSC Service Registry EOSC Provider Portal

4.2.2 The EOSC Research Product Catalogue

The EOSC Research Product Catalogue supports the onboarding of research products from EOSC Data sources. Concretely, research product profiles are metadata records describing the bibliographic, attribution, and provenance information regarding publications, data, software and other products hosted (i.e. deposited) into an EOSC data source.

EOSC Research Product Provider Dashboard: the portal is powered by the OpenAIRE PROVIDE service (http://provide.openaire.eu). Via the EOSC Service Provider Dashboard, EOSC providers can register their data source as a service resource in the EOSC Service Registry and opt to onboard the research products in it. To this aim, the Dashboard requires the user to configure the parameters necessary to harvest the metadata records from the data source (e.,g. OAI-PMH, FTP, protocol) and then the tools to validate the records against the EOSC IF Guidelines for Research Products[2]. If the Data Source passes the validation, then EOSC data curators can set up the metadata harvesting workflows that will automatically schedule regular harvestings to keep the EOSC Research Product Catalogue up to date with the Data Source content.

EOSC Research Graph (aka Knowledge Graph): the component is powered by the OpenAIRE Graph[3] and offers capabilities to manage and access a knowledge graph, where (i) nodes represent EOSC services, data sources, products, as well as authors, organisations, funders, and grants, and (ii) relationships represent the semantic associations between such entities. Nodes of the graph contain metadata information describing the entities and, for EOSC services and products, a relationship to the EOSC Interoperability Framework guidelines the resources are compliant with. The EOSC Research Graph integrates usage statistics from the EOSC Accounting for Research Products and semantic links from EOSC Services to research publications.

The EOSC Research Graph functionalities for resource discovery/access are listed in **Error! Reference source not found.**. The component is built via the integration and extension of the EOSC Service Registry, the OpenAIRE research graph, and the OpenAIRE PROVIDE.

Table 4-3: EOSC research graph functions: resources discovery and access

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
APIs to search, browse, and navigate the EOSC Research Graph	The REST API supports search/browse functionalities for the EOSC data model entities based on their attributes. The API requires a security token.	Realisation of added-value services that reuse content from the EOSC Research Graph; for example, third party resource catalogues (RIs, clusters) or commercial services	APIs swagger component
Generation of EOSC Research Graph datasets (dumps) in Zenodo.org	Datasets (dumps) of the EOSC Resource Catalogue content are made available for download in Zenodo.org	Consumers can access and process offline the whole content of the EOSC Research Graph	Files download from Zenodo.org

[2] https://wiki.eoscfuture.eu/display/PUBLIC/E.+v4.00+EOSC+Research+Product+Profile
[3] http://graph.openaire.eu

4.2.3 EOSC Interoperability Framework Guidelines Registry

The Interoperability Framework Registry is a database containing Interoperability Guideline profiles. By associating services and products to guidelines, providers can indicate compatibility and composability features with other EOSC resources or EOSC Core Components. Guidelines can be onboarded into the registry or be assigned to service resources via the EOSC Service Providers Dashboard. An API allows for CRUD operations on IF DB records and manages relations between IF Guideline entries and EOSC Service Catalogue entries. The EOSC Interoperability Framework Guideline Registry is tightly integrated with EOSC Service Catalogue:

- The EOSC Service Providers Portal component offers the UI for the onboarding of guidelines, by acquiring metadata information in respect of the IF guideline profile[4];
- The EOSC Service Registry provides an API gateway for onboarding and managing guidelines, and for linking them to service resources.

Table 4-4: EOSC Interoperability Framework Registry functions

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
API and UI to onboard interoperability guidelines	EOSC providers can onboard their interoperability guidelines into the EOSC Interoperability Registry through the EOSC service catalogue: the action can be performed via APIs (service catalogues synchronisation framework) or via UIs by provider representatives.	Populating the EOSC Interoperability Framework Registry with Interoperability Guidelines	EOSC Service Registry EOSC Provider Portal EOSC Interoperability Framework Registry

4.2.4 The Integrated Search Index

The EOSC Resource Catalogue contains only EOSC onboarded resources, hence EOSC services, EOSC products into EOSC Data Sources, EOSC IF guidelines, and EOSC training resources. As shown in Figure 4.3 the catalogue is materialised in the Integrated Search Index (for more information about this component, refer to D5.1b.), which is a Solr full-text index that collects service profiles from the EOSC Service Registry, product profiles hosted by EOSC Data Sources from the EOSC Knowledge Graph, EOSC IF guideline profiles from the EOSC IF registry, and training resources from the EOSC Training catalogue.

[4] https://wiki.eoscfuture.eu/display/PUBLIC/EOSC+Interoperability+Guideline+Profile+-+Data+Model

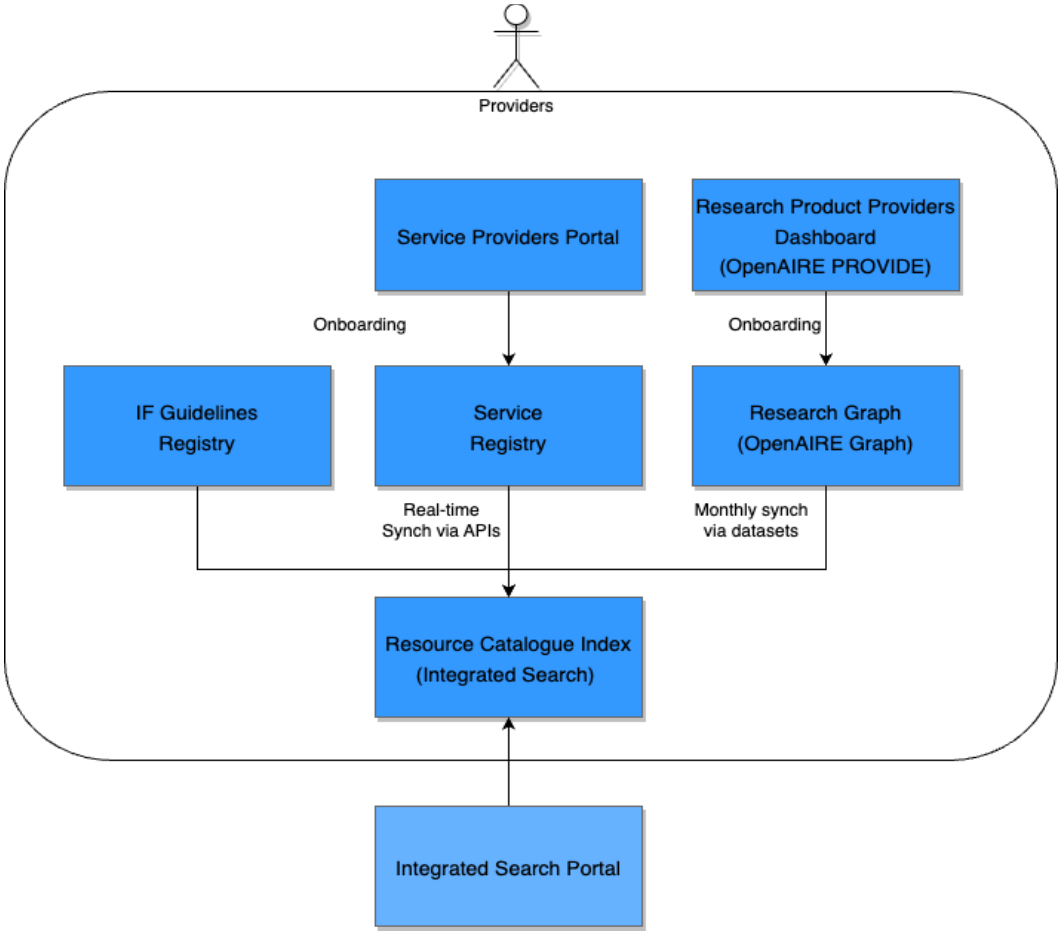

Figure 4.3: High-level Architecture of the EOSC Resource Catalogue components and Integrated Search Index

4.3 EOSC Service Order Management & Composition

One of the formative aspects of the EOSC Portal vision was to facilitate the collaboration between EOSC end-users and EOSC resource providers to stimulate the uptake of EOSC resources. By providing different levels of support for various EOSC resource access methods, the EOSC Marketplace addresses the inherent diversity of the EOSC ecosystem. Among other functionalities supporting resource accessibility, the EOSC Marketplace provides end-user features to order resources, monitor user requests and communicate with resource providers. For the providers, in turn, the EOSC Marketplace offers various interoperability patterns to integrate the order management process in alignment with a vision of a federated system of systems. The implemented interoperability patterns are aimed at individual providers and provider communities adopting the guidelines to integrate their own resource provisioning mechanisms. EOSC's service order management process also engages the EOSC Portal operations team, which plays a key role in the EOSC Portal CRM (Customer Relationship Management), end-user support and guidance through the composability of resources in the EOSC ecosystem.

4.3.1 Service Order Management

As a connection point between resources and order management systems (OMSes) for resource providers, the EOSC Marketplace facilitates the ordering process. It is also an entry point for users looking to advance their research projects by using EOSC resources. They can follow the entire path from resource discovery to order fulfilment in a single portal, resulting in a coherent experience from their side.

On the other hand, providers and communities are presented with several flexible options to integrate their ordering process. Firstly, they can specify offerings for their resources. Providers can configure their offerings using the offering API or an ergonomic UI. Both are flexible enough to support various offering use cases. Secondly, they can handle orders placed in the system in several interoperable ways, either by using the existing

SOMBO system (service order management back office), or by using the marketplace ordering API. The latter allows them to integrate existing order management systems (OMSes) while providing out-of-the-box support for Jira-based solutions as a reference implementation for the integration.

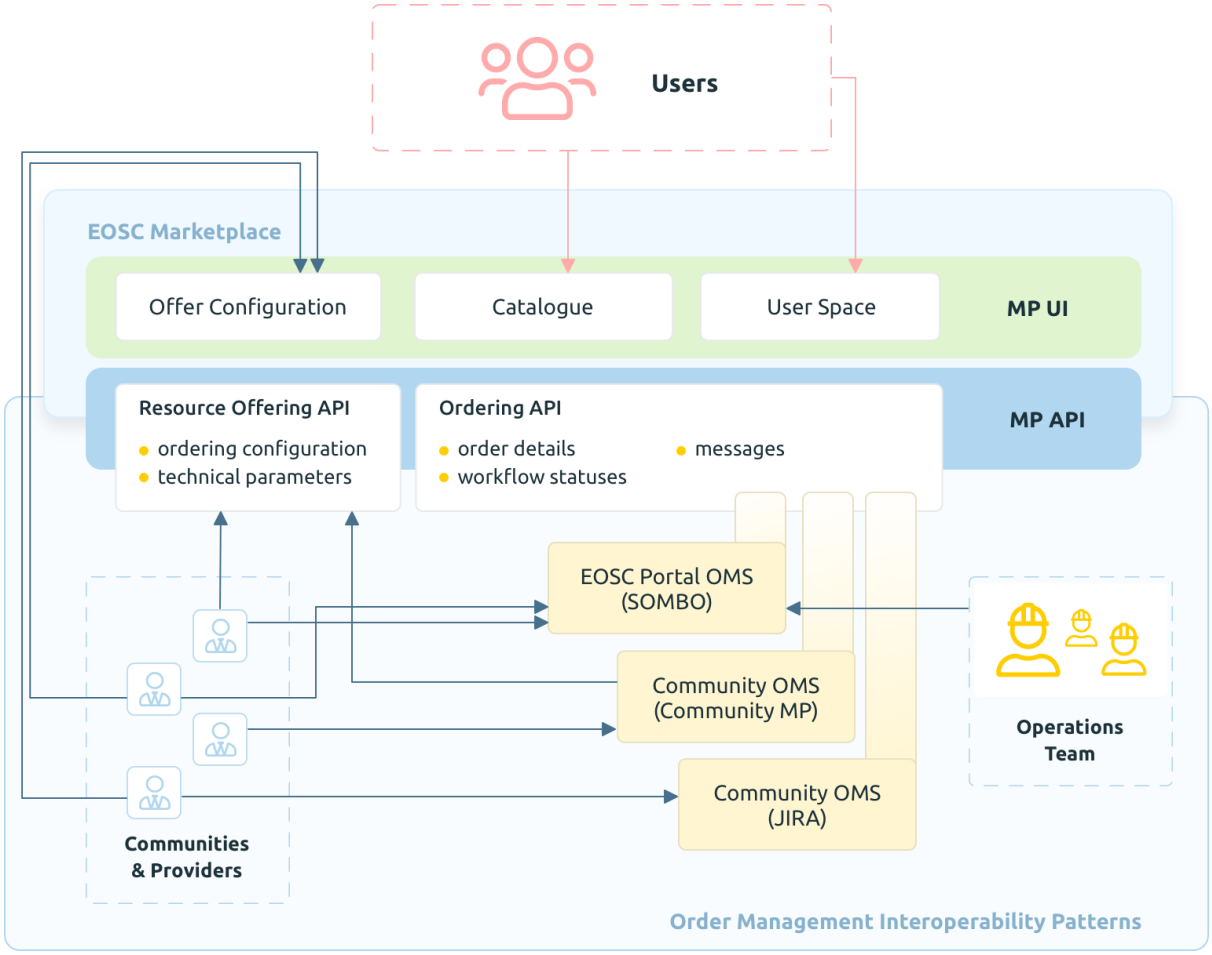

Figure 4.4: Order Management Interoperability Patterns

The EOSC Marketplace is the central part of the order management, which facilitates the order processing and provides various ways to interface with it. End-users can use the main component available under https://marketplace.eosc-portal.eu to access both the catalogue (discovery and access) and the user-space (order management and support). The Marketplace provides offering API and ordering API, for the use of providers, communities and the EOSC Portal operations team, It also provides several methods of order handling.

OMSes are a separate group of components that integrate closely with the Ordering API, but which are free to have additional functionalities for their users. An example of such a component is the EOSC Portal OMS, i.e. SOMBO. On one hand, it has its own UI catering to its users (mostly providers in addition to the operations team). On the other hand, it connects to the EOSC Marketplace through the ordering API.

Figure 4.4 presents elements of the OMSes oriented towards the EOSC Marketplace. However, these elements may be part of a more complex local ecosystem of services. An example could be a marketplace service established by a community of providers, which also integrates with the offering API, thus freeing their providers from having to use the EOSC Marketplace directly.

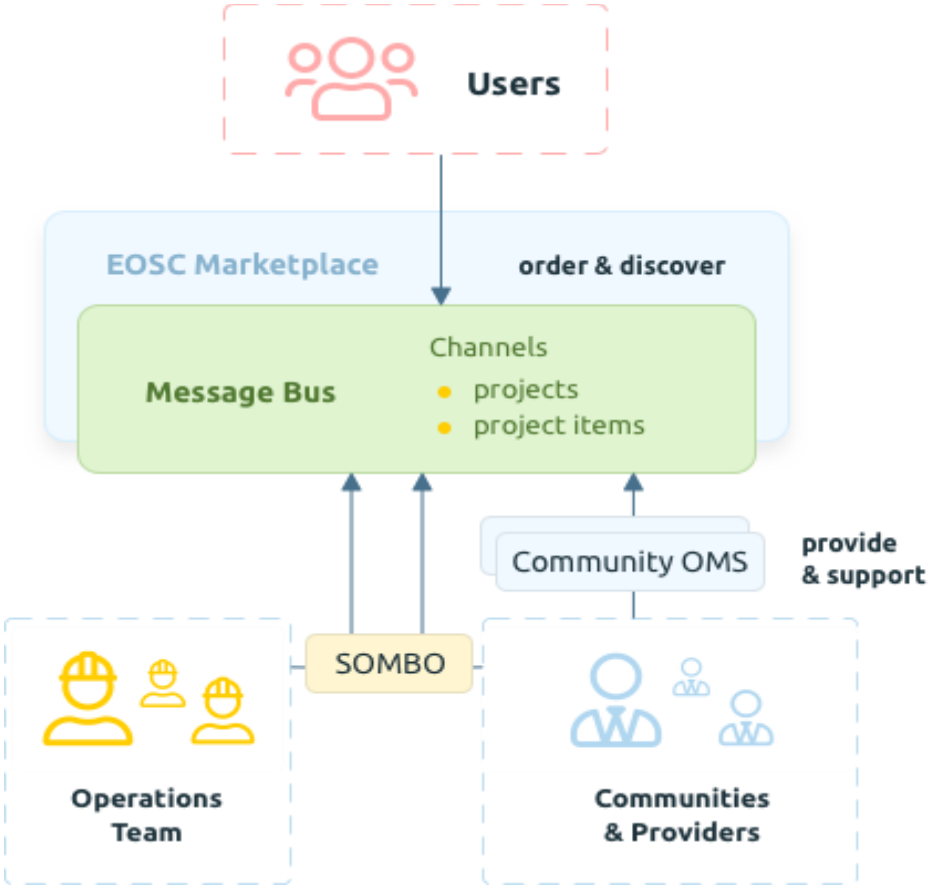

Figure 4.5: Communication between the EOSC Marketplace, the users, the communities and providers, and the EOSC Portal operations team

Specifically, the functionalities provided by the ordering API form a message bus that allows a tri-directional communication (as far as roles are related) between the users, the communities, and providers, and the EOSC Portal operations team. By using a standardised approach, all of them can communicate using their systems of choice.

Table 4-5: Access and compose resources: service order management

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Define resource offers (Offering API or UI)	Define variants of your service: for different tiers, different target users, different SLA etc. Provide information about access type and offer description. Marketplace UI and Marketplace offering API are available for the providers to use.	Allows providers to properly advertise resource variants. Allows end-users to select a service variant suitable for their use case.	EOSC Marketplace
Define resource order related technical parameters (offering API or UI)	Define parameters that need to be provided by the end-user during the order specification. Marketplace UI and Marketplace offering API are available for the providers to use.	Allows providers to collect orders from a user. Allows end-users to specify technical parameters of the ordered resources.	EOSC Marketplace
Select order management system for a resource (offering API or UI)	Define the way the resource orders are processed. Providers may use SOMBO service to handle resource orders or integrate their own order management systems. The OMS may be selected with	Allows providers to choose the preferred order processing method.	EOSC Marketplace

	the use of Marketplace UI or the Marketplace offering API		
Manage resource orders (ordering API or SOMBO UI)	Handle resource orders. Get information about the order specification (technical parameters, project metadata - user typology, affiliation, scientific domain, other requests in the user project). Provision the resources, notifying users about order status. Provide resource access information. Make use of the SOMBO UI or integrate your own order management system.	Allows providers to handle resource orders	EOSC Marketplace SOMBO (service order management back office) Marketplace API client Marketplace Jira adapter
Message user (Ordering API or SOMBO UI)	Send messages to the user requesting a resource or willing to compose resources (project-level and resource order/project item-level communication). Adopt the SOMBO UI or integrate your own order management system.	Allows providers/operations team to communicate with the users	EOSC Marketplace SOMBO (service order management back office) EOSC Marketplace API client EOSC Marketplace Jira adapter
Message resource provider (ordering API or SOMBO UI)	Message another provider provisioning resources for the same user project. Adopt the SOMBO UI or integrate your own order management system.	Allows providers/operations team to consult with other providers during resource provisioning	EOSC Marketplace SOMBO (Service order management back office) EOSC Marketplace API client EOSC Marketplace Jira adapter
Register order management system	Register an order management system that may be used to process orders for a provider or a community of providers. Get credentials to integrate other systems with the use of the Marketplace Ordering API.	Allows providers to integrate existing order management systems with EOSC Marketplace	EOSC Marketplace
Order a resource	Request access to/ provisioning of a resource. Specify technical parameters for the order. Select a project to collect resources for a specific use case.	Allows end-users to request resources	EOSC Marketplace
Monitor order status	Get information about the order delivery progress.	Allows end-users monitor order progress	EOSC Marketplace
Message service provider	Provide additional information for the provider. Describe your use case. Ask for help in using or accessing the resource.	Allows end users to contact providers	EOSC Marketplace
Message EOSC Portal operations team	Describe your use case. Ask for help in contact with the provider. Ask for help in resource composition.	Allows end-users to get support from the Operations Team	EOSC Marketplace
Rate requested resource	Provide ratings for the ordered resources.	Allows providers to get information about the user satisfaction. Allows to get information about resource ratings provided by other users	EOSC Marketplace

4.3.2 Composition

Resource composition is a widespread functionality based on the high-level aim to combine different resources to support a single use case of a user. Ideally, it would include the possibility to compose workflows based on

the combined use of a multitude of available services, utilise infrastructural services to prepare domain-specific service offers, match data coming from various datasets to perform multidisciplinary research, with boundless possibilities to make use of the harmonised set of resources, regardless of their type and interoperability barriers. In practice, many challenges may be addressed in terms of the technical interoperability and integration of services, their maturity level, access policy alignment etc. The evolution of functionalities supporting resource composition should be perceived as a series of steps supporting the increase of interoperability with respect to the technical limitations and adequate attention paid to the specific use cases, implemented as appropriate solutions consistently bridging the gap between the current state-of-play of the EOSC environment and the target vision of the ubiquitous alignment in various resource procurement strategies. Although the understanding of the gaps that may be successfully bridged thanks to the outcomes of the EOSC Future project is still maturing and successively growing with the progress of requirement analysis, there are some initial steps to be taken, identified in the initial phase of the project. A basic assumption in the process of identifying specific resource composition challenges is that the delivery of functionalities for at least two actors in the resource provider and end-user interplay is needed. One subset of the functional requirements is focused on those providers who need to offer their resources in a way that will allow for resource composition. Conversely, end-users require functionalities that allow them to compose resources based on their needs. Both perspectives need to be addressed in the design of any foreseen solution.

EOSC Marketplace user projects allow the end-users to collect various resources that support their use cases. Each of the resource providers taking part in the ordering process is notified and asked to deliver their resource based on the user specification. Each of the providers is able to determine the context (the use case) of the resource provisioning based on the information provided in the project metadata. They are also informed about the other resources that have been requested with the use of the ordering mechanism to support a specific user project/use case. Providers are allowed to communicate with each other with the EOSC Marketplace acting as a broker in the resource delivery. The implemented mechanism also allows the EOSC Portal operations team to coordinate the delivery of multiple resources supporting a single use case, by contacting the respective providers as well as the end-user that requested access. The very same messaging mechanism can be used as a main channel for end-users mainly looking for support in the resource composition. This channel also acts as a part of the EOSC Portal CRM (customer relationship management) in order management.

Providers are offered with SOMBO service (service order management back office) as a ready-to-use solution to handle resource orders and communicate in the scope of the ordering process. They may also integrate their own resource provisioning mechanisms with the use of the Marketplace REST API. Reference API client implementation is provided as a Python library and the reference adapter is made available, allowing to integrate Atlassian Jira as a resource orders management system.

Resource composition was an extensive subject that was researched, analysed, and elaborated upon during the EOSC Future project. It was found that for researchers to be effective and efficient, they require a consistent environment. While researchers possess knowledge of their research topic, they often lack the necessary tools and expert knowledge about potential options to carry out their projects effectively.

Recognizing this need, the EOSC offers a diverse range of research resources including data, software, publications, and datasets. Our team of experts not only assist researchers in finding the required solutions but also go a step further by creating Bundles. These bundles combine two or more complimentary offers, resulting in faster actions and streamlined processes.

By providing technical support tailored to the specific needs of researchers, along with complementary resources, we enhance the efficiency of their work. This allows researchers to focus more on their core tasks, confident that they have the necessary support to carry out their projects effectively. See the overview in the picture below:

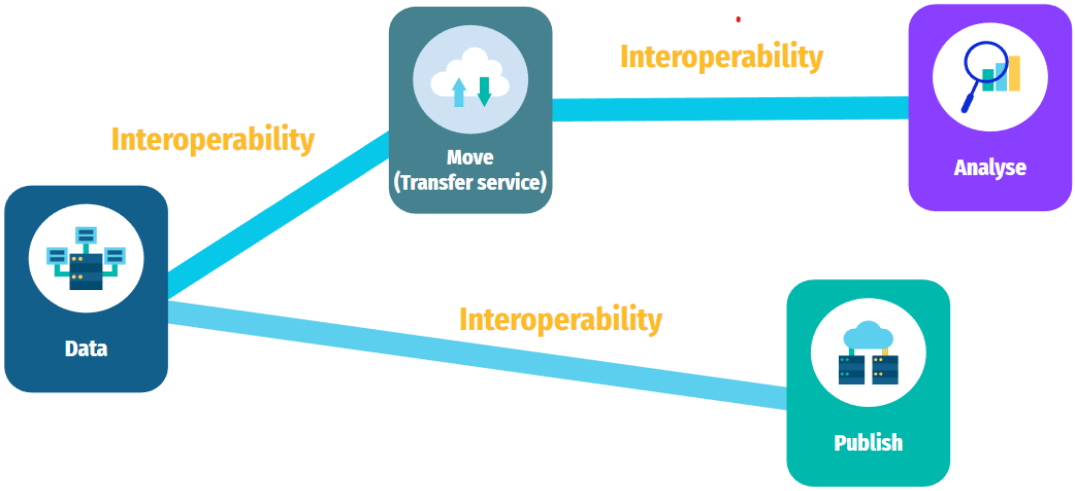

Figure 4.6: EOSC Resources Composition

A detailed analysis was conducted to explore different possibilities of connections that refer to researchers' activities. As a result of the conducted research, we obtained the following researchers' actions:

1. Processing:
    a. Capabilities needed: transfer, compute (possibly external) processing tool, storage (possibly external)
    b. What matters for this goal:
        i. For storage:
            1. Size of the dataset
            2. Time of retention
2. Publishing the datasets:
    a. Capabilities needed: publishing service
    b. What matters for this goal:
        i. Size of the dataset
        ii. DOI
        iii. free/not free
        iv. Domain
        v. Security (anonymise personal data/licence)
3. Data management
4. Anonymisation
5. Move/Storage (big) data
    a. Capabilities needed: transfer, storage
6. Use efficient computing resources
    a. Capabilities needed: computing, storage (possibly external)
7. Discover/re-use data
8. Make repository findable in Open Science
    a. Capabilities needed: data sources
9. Produce data

Regarding the bundles, research was conducted to determine user needs, based on which a comprehensive and extensively consulted dedicated schema for bundles was specified (see D5.2b). Later, a specialised form for providers was prepared based on the schema, enabling them to easily create various combinations of service offers. By exploring the needs of researchers, we have identified the necessity of expanding bundles to include resources of other types, such as training and some research products. Research in this area is still ongoing and will be addressed in future stages of functionality development.

Table 4-6: Access and compose resources: service order management

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Compose resources in user projects	User projects are created by the end-users. Project metadata describe the use case for resources in the project.	Allows end-users to order different resources to support a single use case.	EOSC Marketplace
Order resource bundles	Users are able to order compound resource offers (bundles). The composition of resources is delivered to them based on a stepwise specification of the respective resources needed to deliver the compound order.	Allows the end-users to order composition of resources.	EOSC Marketplace
Handle resource orders for a user project	The providers are informed about the items in the project and can communicate with other providers to deliver support in composing resources for a single use case. Providers can use SOMBO (service order management back office) to handle resource orders or integrate them through API to link their own resource provisioning systems and automate the resource delivery.	Allows resource providers to communicate with each other to deliver resources for a specific use case.	EOSC Marketplace SOMBO (service order management back office) EOSC Marketplace API client EOSC Marketplace Jira adapter
Configure resource bundles	Providers are able to configure compound resource offers (bundles) to support resource delivery based on a set of resources that are composed to deliver the offered value.	Allows resource providers to specify resources delivered as a bundle (composition of resources).	EOSC Marketplace

4.4 EOSC and Open Science Statistics

The EOSC Portal to be developed in WP5 will offer access to a number of different EOSC statistics:

- **Open science statistics,** i.e. queries and data analysis of the EOSC research graph. Statistics include research impact or trends, defined in terms of scientific outcomes, for countries, institutions, research infrastructures (e.g. 'number of open access datasets for Croatia');
- **EOSC statistics,** i.e. dashboards with statistics of usage of the EOSC Portal and service orders;
- **EOSC service accounting,** i.e. indicators about the usage, by EOSC users, of EOSC Core services (the EOSC portal itself) and EOSC Exchange services, in order to transparently report service cost to the EC;
- **EOSC research product usage statistics,** i.e. indicators collected from EOSC data source services related to downloads and views of research products: publications, datasets, or software.

4.4.1 Open Science Statistics

The Open Science Statistics presents a collection of indicators and visualisations that help interested stakeholders (policy-makers and research administrators among others) better understand the open science landscape in Europe across countries and, in the near future, subject areas. The platform assists the monitoring, and consequently the improvement, of open science policy uptake across different dimensions of interest, revealing weak spots and hidden potential. Based on the OpenAIRE research graph, following open science principles and an evidence-based approach, the indicators can be used to provide timely and reliable insights into the evolution of open science in Europe and assist in promoting good practices.

Table 4-7: EOSC and Open Science statistics: Open Science Statistics

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Overview of Open Science across Europe	A set of indicators that present the overview of Open Science across Europe, such as the number of open access publications/datasets and the number of repositories and journals.	Access to indicators and statistics about Open Science across Europe	OpenAIRE Open Science Observatory
Detailed view of Open Science across Europe	Set of more detailed indicators across Europe, such as the number of open access publications by type, country, data source, and organisation, or the number of publications with a PID or a license.	Access to detailed indicators and statistics about Open Science across Europe	OpenAIRE Open Science Observatory
Detailed view of open science for individual countries	Set of more detailed indicators of Open Science for a specific country. This page is similar to the detailed view of Open Science across Europe with the difference that it displays those indicators for a specific country.	Access to detailed indicators and statistics about Open Science for a specific country	OpenAIRE Open Science Observatory

4.4.2 EOSC Statistics

The EOSC Statistics is provided through a Metrics portal that aggregates data from EOSC Portal Google Analytics, the EOSC resource catalogue and the service order management component. These figures are consolidated and provided through different tables and reports. The final aim of this portal is to automatically create reports for the European Commission.

Table 4-8: EOSC and Open Science statistics: Metrics Portal

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Service order metrics	Collect and consolidate data from the service order management component and provide it through tables and figures	Statistics about service order requests and the evolution over time	EOSC-hub Metrics Portal
Service provider metrics	Collect and consolidate data from EOSC resource catalogue and provide it through tables, figures, history.	Statistics about service providers and the evolution in time	EOSC-hub Metrics Portal
EC report	Collect and consolidate data from EOSC Portal Google Analytics, EOSC resource catalogue and the service order Management component. This part is used to provide reports for the European Commission.	Statistics about usage (access, pages, views) of the EOSC Portal and the EOSC Marketplace	EOSC-hub Metrics Portal

4.4.3 EOSC Accounting

The EOSC resource accounting service is able to aggregate (push and pull) usage indicators for different types of EOSC resources and services. It will support the accounting of virtual access resources.

4.4.3.1 Service Accounting

The EOSC Accounting service collects, stores, aggregates, and displays usage information of service resources. This usage data is collected from the resource centres of the EOSC infrastructure, operating services in the EOSC Exchange according to the EOSC accounting framework to be developed in WP3.

Accounting information is gathered from distributed sensors into a central accounting repository, where it is processed to generate summaries that are made available through an accounting portal. Depending on the use case the data may go via intermediate repositories that collate accounting data for particular regions, infrastructures or communities. The overall architecture is depicted in the following figure:

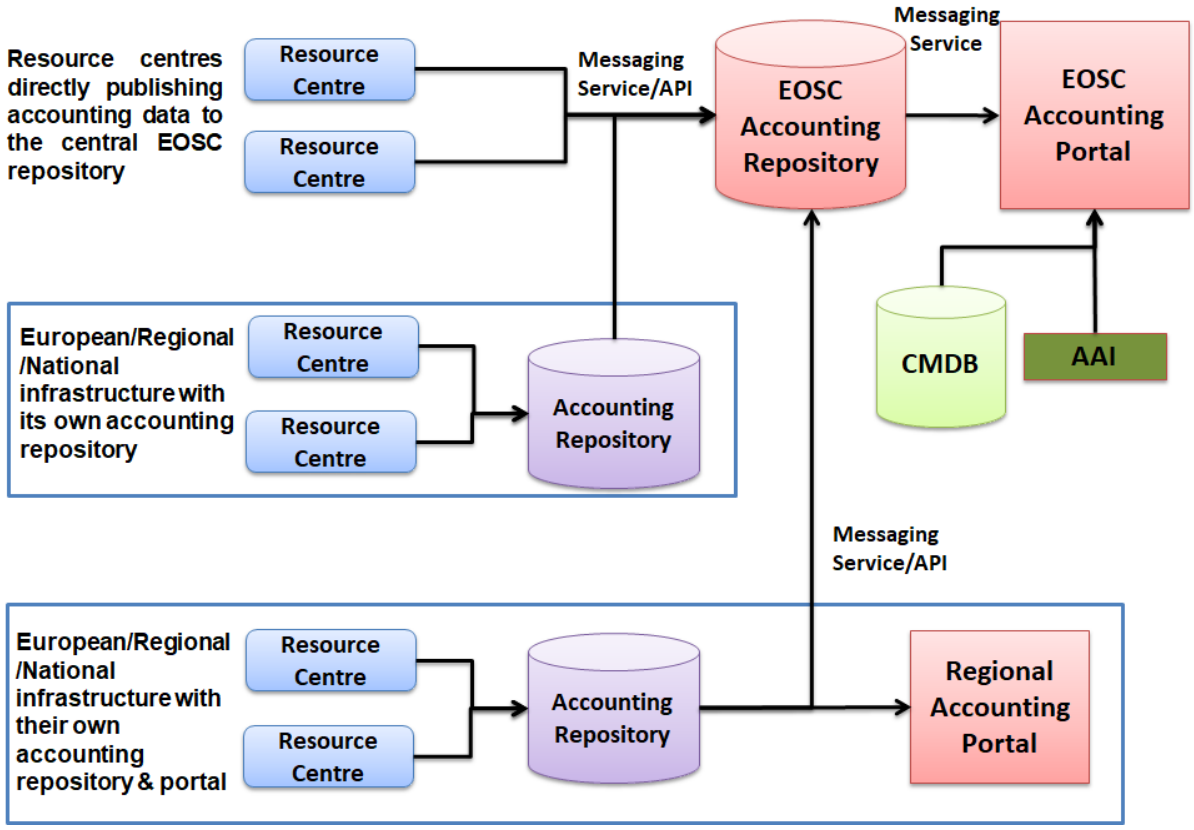

Figure 4.7: High-level architecture of the EOSC Accounting

The accounting repository has a database back end and needs to ensure the exchange of accounting information with peer e-infrastructures. The accounting portal receives and stores accounting information/data from the EOSC Exchange services that participate in the EOSC accounting framework (WP3) and architecture. For example, EOSC services providing computing and storage may report resource consumption by users and user groups (e.g. Virtual Organisation/VO). The accounting repository may then generate aggregated summaries and show views via a web portal. By grouping resource centres in a country on specific time intervals, a customised view can be generated and displayed. The databases are organised into resource record databases (e.g. CPU, VM, storage, etc.), a user record database, and a topology database.

Table 4-9: EOSC and Open Science statistics: Service Accounting

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Aggregated views of usage.	Service usage across centres is aggregated on a selection of metrics into combined views that provide useful breakdowns of usage.	Users can see their resource usage wherever that usage occurred.	EOSC Accounting for Services, EOSC Accounting for Research Products, EOSC Messaging
Views that allow usage to be checked against allocation.	Allocations are imported and displayed against the service used that has been collected previously.	Users and their community leaders can see how much of their allocation they are using.	EOSC Accounting for Services, EOSC Accounting for Research Products, EOSC Messaging
Provider-centric views of resource usage by users.	Alternative provider-centric views can be generated based on the same aggregated data once it has been processed.	Resource providers can see how their resources are being used.	EOSC Accounting for Services, EOSC Accounting for Research Products, EOSC Messaging

Views that allow comparisons to be made between resource providers within and between regions and communities.	Data can be viewed at different levels of the topological hierarchy.	Managers of national research and e-infrastructures can see how their group of resource providers are performing.	EOSC Accounting for Services, EOSC Accounting for Research Products, EOSC Messaging

4.4.3.2 Research Publications, Data and Software Accounting

The OpenAIRE Usage Counts Service gathers usage activity for research items like publications, research data and software, as well as consolidated usage statistics reports respectively, from OpenAIRE's distributed network of data providers (repositories, e-journals, CRIS) by utilising open standards and protocols and exploiting reliable, consolidated, and comparable usage metrics like counts of item downloads and metadata views conforming to the COUNTER Code of Practice.

The Usage Counts Service tracks anonymised usage events, i.e. metadata views or downloads, for research items using either plug-ins and patches for repository platforms like EPrints or DSpace, or by using a generic tracking tool for all other infrastructures. Usage information is transferred offline, to OpenAIRE's databases for further processing using the COUNTER Code of Practice and statistical analysis. Usage statistics are deployed via OpenAIRE's Portal, OpenAIRE's Repository Dashboard or via an API endpoint, which is based on SUSHI-Lite protocol.

The Service generates regular dumps of the usage stats for integration into the EOSC Knowledge Graph, to support discovery by statistics and their display.

Table 4-10: EOSC and Open Science statistics: Research publications, data and software

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Constant tracking of views and downloads of open science content providers	Platform dedicated software or generic script is used to track usage events in an anonymised and real time manner.	Statistics on usage activity	OpenAIRE UsageCounts
Collection of aggregated usage statistics reports	Usage statistics are collected from sources like national aggregators, without the need to install tracking software at each node (e.g repositories). Dedicated software that exploits the SUSHI-Lite protocol is used to collect the reports.	Access to aggregated usage statistics reports	OpenAIRE UsageCounts
Provision of standardised usage statistics reports compliant with the COUNTER CoP	An open-source API, based on SUSHI-Lite protocol and the COUNTER code of practice, is used to publish the reports.	Access to impact indicators for research products, which allows evaluation of research from an early-stage Access to evidence-based analytical metrics of views and downloads, aggregated from all over the world. Enables comparability with statistics from other data sources	OpenAIRE UsageCounts
Usage statistics integration with the Knowledge Graph	The service generates a dump of the collection of aggregated data	The EOSC Knowledge Graph uses the dump to integrate usage statistics at the PID level	OpenAIRE UsageCounts

4.5 Helpdesk

The helpdesk is an essential service in any IT infrastructure to provide efficient support for its services. It is used as an entry point for users and should implement incident and request management and provide an efficient communication channel between customers, users and providers of the IT resources and services.

The EOSC Helpdesk works as a unified ticketing system by managing the requests related to different Core services operated by different providers. The EOSC helpdesk effectively supports customers in communication with EOSC. It provides a personal dashboard, history and current status of submitted requests. It enables an effective first level of support, which is connected with multiple support units for single EOSC Core service. The EOSC Helpdesk can be also integrated with third-party helpdesks, so that tickets could seamlessly propagate to the dedicated helpdesk after submission in the EOSC Helpdesk if required.

Current EOSC Helpdesk design and its implementation enables the integration of community helpdesks and the delivery of the helpdesk as a service to multiple communities simultaneously. It allows for community-specific configurations. Each community can have its own dedicated portal which acts as a single point of interactions for community users, branding, and customization options.
Figure 4.8 shows ticket workflow in the distributed environment of the EOSC Helpdesk, which integrates or implements community support channels, different submission methods and notification schemes. To summarise the complex workflows shown in the Figure 4.8 it's important to mention that EOSC Helpdesk implements effective ticket management for EOSC Core or community services and enables structured information exchange and collaborative work between different support channels if required. The ticket submitted to central EOSC Level 1 Support can be easily propagated to any integrated community channel and vice versa.

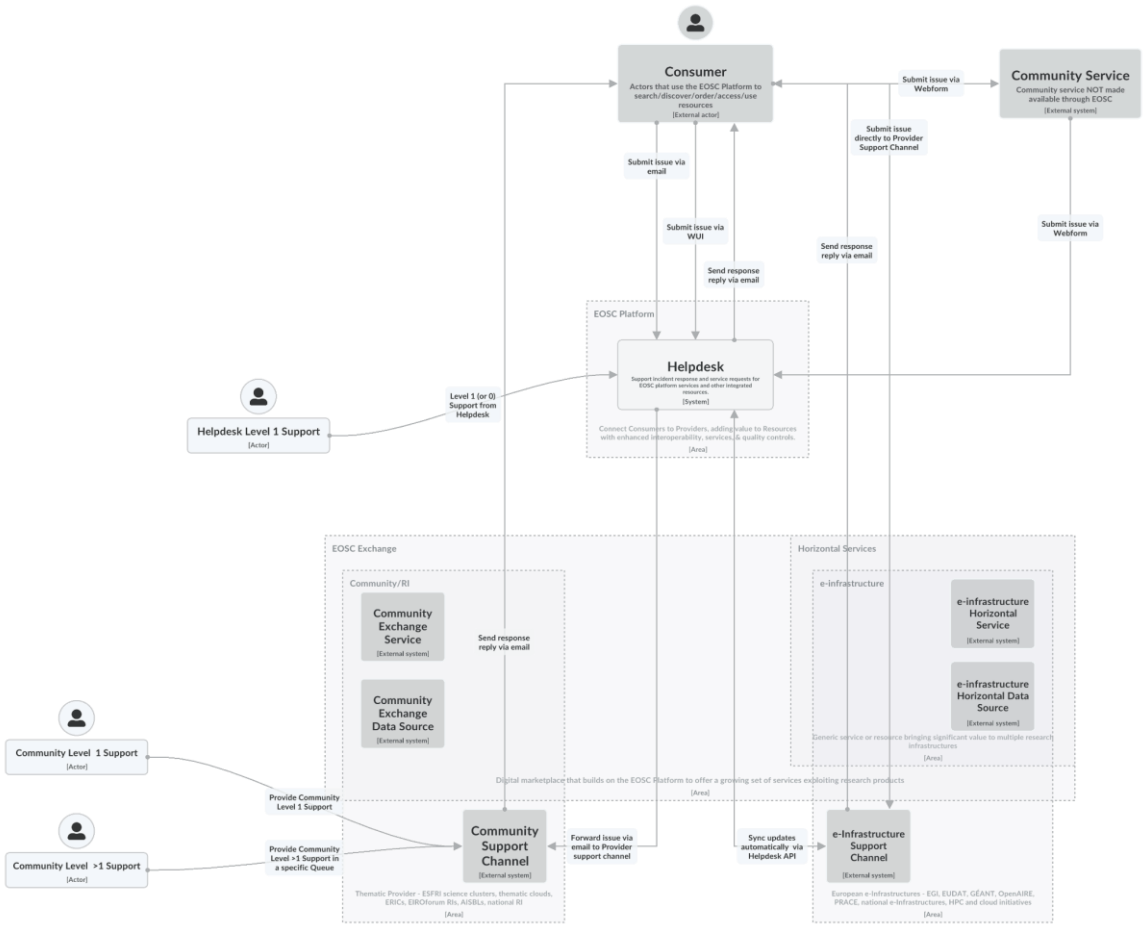

Figure 4.8: Functional architecture and ticket workflow of the EOSC Helpdesk

The EOSC helpdesk is provided as a service for EOSC service providers, communities and users and supports three main usage scenarios:

- **Direct Usage:** Use the EOSC helpdesk directly as the ticketing system. This scenario is to be implemented when a provider does not have its own helpdesk and would like to use EOSC Helpdesk to manage the user requests, addressed to the community services. In this case, the community obtains a portal, a support unit or a set of support units for its own disposal.
- **Ticket Redirection:** Use the EOSC helpdesk only as a contact point to redirect the entry request for the specific service to a mailing list or 2nd level ticketing system. In this case, the EOSC Helpdesk central service would simply redirect by e-mail or via API the incoming tickets to the external system directly managed by the provider.
- **Full Integration:** Bi-directional integration of the community ticketing system with the EOSC helpdesk, which means the full synchronisation of the content between the two systems. In this scenario, the community tickets can be managed in any of the two helpdesks. For example, the ticket which initially has been opened in the EOSC Helpdesk is propagated to community helpdesk and all changes done for this ticket in community helpdesk are visible also in EOSC Helpdesk.

Table 4-11 provides the description of the main functions currently implemented in the EOSC Helpdesk.

Table 4-11: EOSC Helpdesk

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Creation of a ticket using helpdesk interface	The ability for the requester to create, view or modify ticket using helpdesk dashboard	Users could create and manage tickets centrally in EOSC Helpdesk.	Helpdesk Back Office
Creation of a ticket via webform or mail	The ability to create ticket without AAI login by filling a webform or sending an e-mail	Allows users to send requests without accessing or login to the helpdesk system directly from the webpage of dedicated service in use e.g. Marketplace	Helpdesk Back Office
Search filters	The ability to search any ticket or group of tickets according to the filter or set of the filters using the search interface	Users could search, sort their tickets based on multiple search criteria and obtain a better overview of their tickets.	Helpdesk Back Office
Notification	The function allows to notify the ticket submitter, owner or anyone subscribed to the ticket about the ticket modification.	Keeps users informed about any change or final solution of the request.	Helpdesk Back Office
Creation of a new support unit	Any EOSC provider could manage support units for any entity like service, service group, team of developers etc.	EOSC providers could use helpdesk as service to manage support for services for their users	Helpdesk Back Office
API Interface	An API interface implemented via RESTfull API which allows communication between helpdesk and other services	EOSC providers could integrated their helpdesk systems or other services with EOSC Helpdesk	Helpdesk Back Office
Helpdesk Delivery as a Service	Standalone community Domain, branding, customization options, community group and agents management	Unified and decentralised access to community support of the service	Helpdesk Multiportal module

4.6 EOSC Monitoring

The EOSC service availability and reliability monitoring (EOSC Monitoring) is a key service needed to gain insights into an infrastructure, the applications, services, and even into processes/behaviours. The EOSC Monitoring has been implemented in EOSC-hub through ARGO and it continued its evolution and operation in

EOSC - Future. It monitors services by emulating typical user scenarios, which allows them to infer the quality of service the actual user gets. It mimics the actual end-user behaviour without requiring special privileges or special configurations from the service provider side. As a result, ARGO offers near real-time status updates which allow both end-users and site admins to have an overview of the service offered at any given point in time. The major objective of the monitoring system is to quickly identify and correlate problems before they affect end-users and ultimately the productivity of the services, the infrastructure and, finally, the organisation.

The high-level architecture of the EOSC Monitoring is shown in the following figure. The service collects status (metrics) results from one or more monitoring engine(s) and delivers daily and/or monthly availability (A) and reliability (R) results of distributed services. Both status results and A/R metrics are presented through a Web UI, with the ability for a user to drill down from the availability of a site to individual test results that contributed to the computed figure.

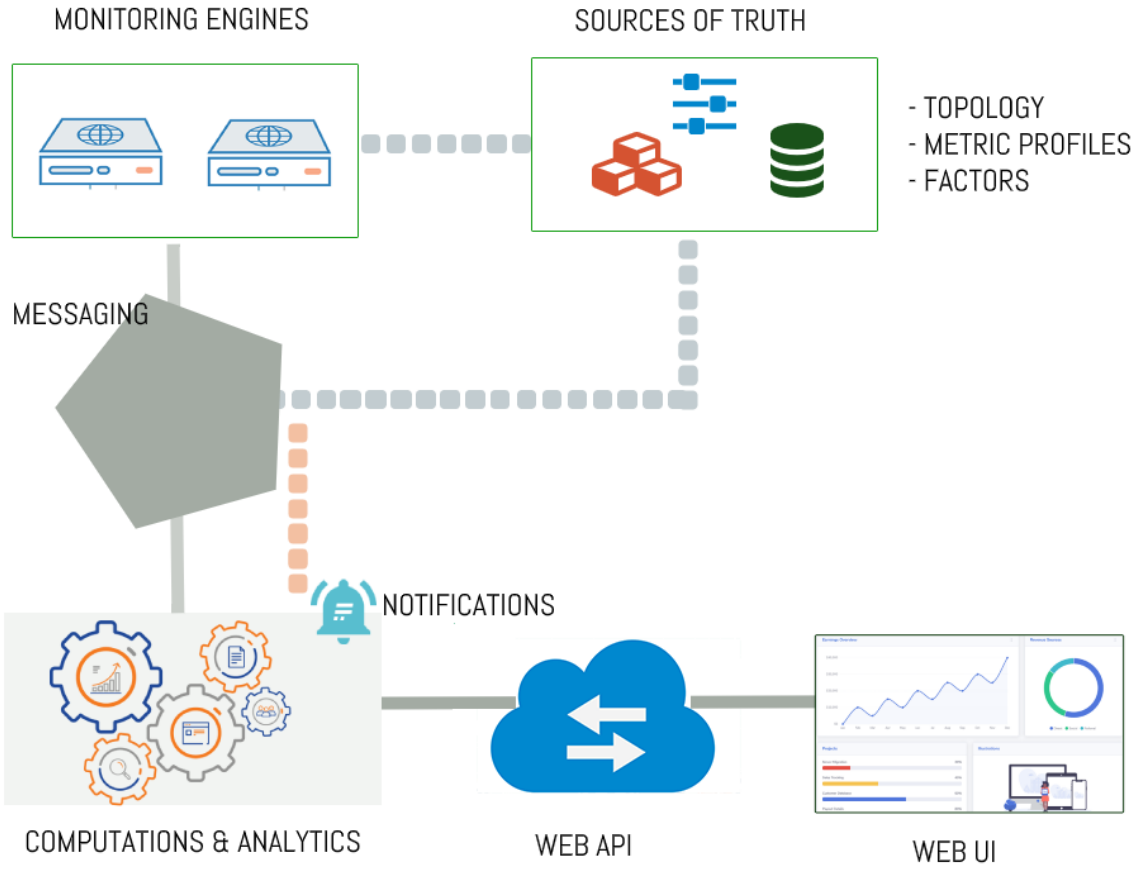

Figure 4.9: High-level architecture of a monitoring service

Table 4-12: EOSC Monitoring

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Get topology from the EOSC CMDB sources	Ability to automatically ingest topology items and their organisational hierarchies from EOSC CMDB sources	Easily ingest topology information from EOSC CMDB sources	EOSC Monitoring for Core Services
Get topology from the EOSC service registry	Ability to automatically ingest topology items and their organisation hierarchies from service registry	Easily ingest topology information from service registry	EOSC Service Registry, EOSC Monitoring for Exchange Services

Manage Reports	Ability to define reports that produce availability and reliability results based on specific profiles and options. Users with administrative rights can manage reports. Other users may only view the results generated by the report.	Using the same superset of monitored data as origin, users can create different reports based on different criteria (subsets of topology, organisation of hierarchies, thresholds etc) that produce different sets of results and views	EOSC Monitoring
Get a/r results	Ability to get, for each report, the daily/monthly availability and reliability results produced for monitored services on different levels of hierarchy (service endpoints, services, groups of services).	EOSC users can easily observe the availability and the reliability of services (daily, monthly) they are interested in using.	EOSC Monitoring
Get status results	Ability to get, for each report, the daily/monthly status timeline results produced for monitored services on different levels of hierarchy (service endpoints, services, groups of services).	EOSC users can easily observe the state of services (near real-time) they are interested in using.	EOSC Monitoring
Get trends	Ability to get results produced by analytics that signify various trends such as top items in critical state, or top flapping items (that change states constantly).	EOSC users can get valuable insights on emerging patterns and trends among the services they are interested in.	EOSC Monitoring
Issue a recomputation request	Ability to issue a recomputation request for a specific past date in order to exclude items from the results due to configuration issues between checked items and monitoring service	Ensures consistent compute data even after disputes from service owners	EOSC Monitoring
Manage metric profiles	Ability to manage metric profiles (per report) that define which services and checks are relevant and should be taken into account when producing the status and A/R results. The user with administrative rights can easily create reports that keep services and metric checks that the user deems relevant	Ability to get tailored and focused availability/reliability and status reports containing relevant services and metrics	EOSC Monitoring
Manage aggregation profiles	Ability to manage aggregation profiles per report which instruct specifically in which ways the various monitored items are combined in order to produce results for top level groupings. The users with administrative rights can easily reorganise the way the service endpoints and the service themselves are combined on higher groups (such as group of services) either in a distributed or a highly available way.	Ability to get tailored and focused availability/reliability and status reports with specific organisation of items in higher groups.	EOSC Monitoring
Manage threshold profiles	Ability to manage threshold profiles. A Threshold profile contains a list of threshold rules. Threshold rules refer to low level	Ability to get tailored and focused availability/reliability and status reports with	EOSC Monitoring

	monitoring numeric values that accompany metric data in combination with threshold limits applied on those values that can deem the status of the monitoring result as 'warning', or 'critical' etc. Users with administrative rights can easily define custom thresholds and override defaults from probes.	finetuned produced states based on thresholds.	
Quick View Issues	Ability to quickly view a list with the most problematic monitored items (services, endpoints etc..)	Quickly view the most problematic items of the topology that require attention	EOSC Monitoring
View topology of monitored items	Ability to view (with rich filtering support) topology items such as service endpoints and hierarchical groupings of them such as sites and service groups	Easily view topology information and how it changes from day to day	EOSC Monitoring
View service results	Ability to view availability and reliability and status timeline results scoped to a specific service or service group.	Users such as service owners can access the API and directly view results scoped only for the services they own	EOSC Monitoring
Notifications	Ability to get notified through email about failing or recovering services with rich information on what went wrong	Service owners can quickly get notified about failing components.	EOSC Monitoring

4.6.1 EOSC Monitoring for Core services

EOSC Core Monitoring is the service that provides availability and reliability (A/R) and status reports for all the services that are part of EOSC Core. It gets its topology from the EOSC CMDB sources and probes each service accordingly so as to calculate its A/R scores and status timelines.

The functions of EOSC Core monitoring are described in the previous section (4.6 EOSC Monitoring)

4.6.2 EOSC Monitoring for Exchange services

EOSC Exchange Monitoring is the service that provides availability and reliability (A/R) and status reports for all services onboarded to EOSC. It gets its topology from the EOSC service registry and probes each service accordingly in order to calculate its A/R scores and status timelines. EOSC-Exchange monitoring should also be capable of consuming results from external monitoring services from existing e-infrastructures and other external sources.

This service should also be capable of exporting results via an API to all service owners and other external systems/portals.

The functions of EOSC Exchange monitoring are described in the previous section (4.6 EOSC Monitoring)

4.7 EOSC Messaging

The Messaging service is a real-time messaging service that allows the user to send and receive messages between independent applications. It is implemented as a publish/subscribe service. Instead of focusing on a single messaging service specification for handling the logic of publishing /subscribing to the broker network, the service focuses on creating nodes of publishers and subscribers as a service. In the publish/subscribe paradigm, publishers are users/systems that can send messages to named-channels called topics. Subscribers are users/systems that create subscriptions to specific topics and receive messages.

EOSC Core is a distributed system of systems. A Messaging service is therefore required to integrate these individual systems into a coherent EOSC Core. Messaging is used as a transport layer for the integration between different core services. The main services that are using the messaging are:

- **Accounting**: it adopted the messaging as a transport layer for collecting accounting data from the various service providers. The accounting information is gathered from different collectors into a central accounting repository, where it is processed to generate statistical summaries that are available through the accounting portal.
- **Monitoring service**: it uses the Messaging service to send the messages from the monitoring engine to its components, and or to send the monitoring data to external services.

Adoption of the Messaging service from other EOSC Core services will be assessed during the project lifetime.

Table 4-13: EOSC Messaging

Functions	Description (how the function is built and what it provides)	Added value for EOSC users	Components
Publishing Messages to the Messaging Service	The service provides an API that can be used to publish messages.	Provide a transport layer & cache mechanism between services	EOSC Messaging
Consuming messages from the Messaging Service	The service provides an API that can be used to consume messages.	Provide a transport layer & Cache mechanism between services	EOSC Messaging
Provide publishing and consumption metrics	Messaging provides various metrics on publishing and consumption actions such as number of messages, size of data, transport rates etc.	Easy to view statistics about publishing and consumption activities	EOSC Monitoring, EOSC Messaging
Ability to replay published messages	Messaging provides mechanisms for consumers to replay old messages and re-consume them again.	Ability to reprocess old messages on demand based on reference points using specific message ids or publishing dates	EOSC Messaging
Push messages automatically to verified remote endpoints	Messaging provides the ability to use push subscriptions attached to topics so as to automatically forward published messages to verified remote endpoints	Ability to use messaging to easily create reactive systems that publish messages automatically to remote endpoints and thus avoiding polling mechanisms on the consumer side	EOSC Messaging
Sites publish messages to accounting project in the messaging Service	The sites accounting data follow a predefined format and are published in a predefined topic name in the messaging service.	Enable easy publishing of data so the accounting may calculate the consumption of the services of the site.	EOSC Accounting for Services, EOSC Messaging
Accounting consumes messages related to accounting data from Sites.	The accounting service is consuming the messages from the messaging service.	Easily consume data via messaging so the accounting may calculate the consumption of the services of the Site.	EOSC Accounting for Services, EOSC Messaging
Monitoring engines publish monitoring data to Monitoring-analytics engine	Monitoring engines use messaging to publish monitoring data to be gathered from monitoring analytics engine	Easily publish data from different monitoring sources to a final destination for analytics and report generation	EOSC Monitoring, EOSC Messaging
Monitoring-analytics engine consumes monitoring data	Monitoring analytics engine uses messaging to consume (gather) monitoring data published from monitoring engines	Easily gather data from different sources to a central destination to perform analytics and generate reports	EOSC Monitoring, EOSC Messaging

Republish monitoring data for specific sites.	Republisher script that consumes and publishes (distributes) messages to specific topics (e.g. sites).	Can easily have access to the monitoring data.	EOSC Monitoring, EOSC Messaging

5 Technical Architecture

This section describes the technical architecture of the EOSC Core Back-Office. For each functional component identified in the previous sections, the selected background components are introduced with a short description, a list of the main features and other relevant information (e.g. link to documentation). In addition, for each of these background components, a technical roadmap was defined. It identifies the technical work that has been conducted (1) to enhance the components and fill the gaps to satisfy the requirements, and (2) to define and develop the interfaces and API needed to connect the component to the other components with the aim to create a homogeneous and coherent EOSC Core Back-Office.

5.1 EOSC Resource Catalogue

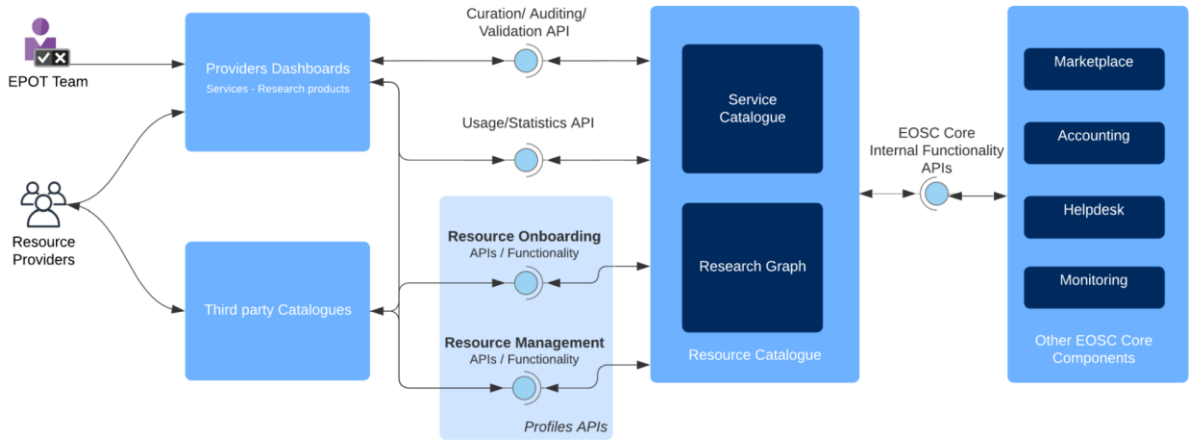

Figure 5.1: EOSC Resource Catalogue high-level architecture. Arrows identify API interfaces described below.

The EOSC Resource Catalogue consists of five components, the EOSC Service Providers Portal, the EOSC Service Registry, the EOSC Research Product Providers Dashboard, the EOSC Research Graph, and the Integrated Search Index (details available in D5.1b), as described in the functional architecture of the service, as depicted in Figure 5.1. The components are delivered by combining and integrating the efforts of EOSC Enhance and OpenAIRE-Advance projects in providing respectively the first release of the EOSC Service Registry/Provider Portal and the OpenAIRE Graph, together with the EOSC Enhance Provider Portal and the OpenAIRE PROVIDE services in support of the onboarding procedures for services and research products. The goal is to continue the operation of the existing services, integrating their data and extending their APIs and UIs where necessary, to offer to EOSC users a single access point to an EOSC Resource Catalogue.

The following sections describe how the OpenAIRE and EOSC Enhance components have been adapted and combined to deliver the EOSC Resource Catalogue and provide the technical features of the existing components: EOSC Service Registry (Table 5-1), EOSC Provider Portal (Table 5-2), EOSC Research Graph (Table 5-3), and EOSC Research Product Dashboard (Table 5-4).

5.1.1 EOSC Research Graph.

5.1.1.1 Data model and Content

The EOSC Research Graph is the result of integrating the EOSC Service Registry as a data source of the OpenAIRE Graph. Specifically:

- The OpenAIRE Graph is a metadata knowledge graph keeping an up-to-date map of the scientific/scholarly communication record. It contains information about research products (publications, datasets, software, and other products), the related data sources (repositories,

databases, publishers, etc. where the research product is made available), funders (EC, European countries, and international funders), the related projects, authors (ORCID), and organisations (ROR, GRID, OrgReg, ISNI, PIC, etc.).
- The EOSC Service Registry maintains a list of service and provider profiles, as published by EOSC providers and curated by EOSC staff (EPOT Team).

OpenAIRE Graph's integration

- The OpenAIRE Graph data model has been extended to include EOSC Service entities, EOSC IF guidelines, and usage statistics for research products (EOSC Accounting for Research Products).
- The EOSC Service Registry has been included as a data source of the Graph, and workflows for metadata collection (harvesting) and integration in the Graph have been developed. EOSC Service profiles are deduplicated against the various data source profiles collected from OpenDOAR, FAIRSHaring, and re3data registries of data sources, so as to ensure a disambiguated Graph.

EOSC Service Registry's integration

- The EOSC Service Registry pushes its content to the OpenAIRE Graph, to keep it in sync with additions, removals, and updates. EOSC services appear as nodes in the OpenAIRE Graph, as well as their relationships to organisations that are EOSC providers.
- The EOSC Service Registry has updated its data model to include: (i) the data source subclass, (ii) the service catalogue subclass, (iii) the training resource subclass, and (iv) the EOSC IF guidelines.
- The EOSC Service Registry now supports "extension" functionality for providers to take advantage of seamless integrations with other EOSC Core components like Monitoring and Helpdesk for their resources. Moreover, all resources are automatically monitored for availability/reliability and online status after onboarding approval.

5.1.1.2 APIs and data dumps

OpenAIRE Research Graph's integration

- The OpenAIRE Graph adapted its APIs to offer search, browse, and navigation functionalities (i.e. link resolution) that take into account services and EOSC Interoperability Framework Guidelines.
- The OpenAIRE Graph delivers monthly dataset dumps of the EOSC Research Product Catalogue content (subset of the OpenAIRE Graph related with EOSC data sources and associated products) in order to support the population of the Integrated Search index required by the EOSC portal in in WP5.

EOSC Service Registry's integration

- The EOSC Enhance Service Registry has updated its data model to ensure the milestones regarding 'access to EOSC portals/catalogue', 'access to INFRAEOSC-07 services', and 'access to horizontal services' can be demonstrated.
- The EOSC Enhance Service Registry pushes its content to the OpenAIRE Graph, to make sure EOSC services will appear and be properly linked to the related EOSC products.

5.1.2 EOSC Service and Research Product Provider Portals

5.1.2.1 Data model and actual content

EOSC Provider Portal's integration and extension roadmap

- The EOSC Provider database will have to adapt its data model to include PIDs for organisations (more than one); in the case of EOSC providers that do NOT have a PID, specific policies will have to be defined (e.g. 'obligation to register to one of the authorities endorsed by the EOSC').

OpenAIRE PROVIDE's integration and extension roadmap

- OpenAIRE PROVIDE[5] delivers UIs for the registration, validation, and management of EOSC data source profiles; the functions are intended to check the compliance to the OpenAIRE Guidelines of the metadata records of the products exposed (made harvestable) by a given data source.
- The Service makes sure that registered and validated data sources willing to be part of the Graph can be eligible to become EOSC services. In such cases, and prior request of the data source manager, OpenAIRE's PROVIDE will register the data source as a service in the EOSC Service Registry. To this extent, OpenAIRE's PROVIDE must become a trusted catalogue for the EOSC, via formal agreements, operational plans, etc.

5.1.2.2 User Interfaces

EOSC Enhance **Provider Portal's integration**

The EOSC Enhance Portal Provider has been extended to manage data source and service catalogue profiles, training resources, and interoperability guidelines. For data sources, the following workflows have been implemented.

From Provider's Portal to Service Registry and PROVIDE: data sources services are registered via the Service Registry.

- When a provider registers a data source, the Portal accesses OpenAIRE's PROVIDE APIs to access the list of data sources available out there and pick the service profile of the one of interest (today OpenAIRE refers to re3data.org, OpenDOAR, and FAIRSharing). The Provider's Portal then proceeds to the registration of the data source to the Service Registry, by tracking the OpenAIRE ID.
- The Service Registry UIs has been updated to allow the provider of data source service to open the relative pages in the OpenAIRE's PROVIDE and proceed with the registration/validation of the research products for inclusion in the EOSC-RG (via the OpenAIRE Research Graph).

From PROVIDE to Service Registry: data source services registered via PROVIDE

- PROVIDE UIs will request the data source managers, who registered and successfully validated data sources for the OpenAIRE Graph, to register the data source services to the EOSC Service Registry. In the case of positive response, PROVIDE will register the data source profile to the EOSC Service Registry on behalf of the data source manager. OpenAIRE's PROVIDE de facto becomes a trusted in-sync catalogue for the EOSC (for the set of validated data sources willing to appear in the EOSC-Exchange).

Table 5-1: EOSC Service Registry technical features

Component Name	EOSC Service Registry
Short description	The Service Registry is the storage component offering the necessary programmatic interfaces for the addition, modification, and access to information regarding providers, resources and user activity collected in the EOSC portal. It contains: • Information for resources and providers entered by providers during the onboarding process based on the EOSC profiles schema. • Information collected by the interaction of providers with the portal and specifically with the Providers Component for the management and customisation of their

[5] http://provide.openaire.eu

	offering. Such information includes updates and changes applied to a resource offered by a provider. • Information collected by the interaction of the end users with the Portal and specifically with the End User Component, including statistics for the end user visitation, orders, personalisation features such as user ratings, favourites, user profiles. • Registry administration, which offers functionality to validate/approve/curate the contents of the EOSC Service Registry. More specifically, it offers functionality for the EOSC Portal Onboarding team to manage incoming requests for the validation of a provider's or a resource registration, as well as perform a series of quality controls in the contents of the registry. • REST APIs (for enabling synchronous communication). All registry content is available in the form of open REST APIs in https://api.eosc-portal.eu/openapi for both XML and JSON representation schemas. In general, all get methods are open, whereas POST and PUT require an API key. • Apache Active MQ[6] (for enabling asynchronous communication). An SSL JMS messaging service enables the subscription of clients to registry-generating events, regarding CRUD operations on providers and resources. • ResourceSync[7] (for enabling harvesting). The registry offers a resource sync API, which can be harvested by external systems- harvesters.
Main features	• EOSC profiles management (vocabularies, schema, etc); • Onboarding of Providers and Resources to EOSC via APIs; • Management of Providers and Resources records to EOSC via APIs; • Validation mechanism via the APIs; • EOSC portal Id assigned to resources and providers; • Versioning of providers\resources; • Messaging (JMS) of CRUD events for 3rd party consumers; • Authorisation of provider users, and EPOT team.
URLs	• EOSC Portal API: https://providers.eosc-portal.eu/openapi • EOSC API Sandbox https://sandbox.providers.eosc-portal.eu/openapi • EOSC API Beta https://beta.providers.eosc-portal.eu/openapi • EOSC API base URL: https://api.eosc-portal.eu/
Documentation	• UI Documentation: https://eosc-portal.eu/providers-documentation/eosc-provider-portal-basic-guide • API: https://providers.eosc-portal.eu/developers • GitHub: https://github.com/madgeek-arc/resource-catalogue
API types and use-cases	**EOSC API base URL:** https://api.eosc-portal.eu/{method} API Calls and examples: **Provider Controller (API):** • **put /provider**: updates the provider assigned the given id with the given provider, keeping a version of revisions. • **get /provider/all:** filter a list of Providers based on a set of filters or get a list of all providers in the catalogue. • **get /provider/services/{id}**: Get a list of services offered by a provider. • **get /provider/{id}:**Returns the provider with the given id. **Service API:**

[6] http://activemq.apache.org/
[7] http://www.openarchives.org/rs/toc

	• **post /resource:** creates a new Resource. • **put /resource:** updates the resource assigned the given id with the given resource, keeping a version of revisions. • **post /resource/validate:** validates the resource without actually changing the repository. • **get /resource/{id}:** get the most current version of a specific Resource, providing the resource id. • **get /resource/all:** filter a list of resources based on a set of filters or get a list of all resources in the Catalogue. • **get /resource/by/{field}:** get all resources in the catalogue organised by an attribute, e.g. get Resources organised in categories. • **get /resource/byID/{ids}:** get a list of resources based on a set of ids. • **get /resource/{id}/{version}:** get the specified version of a resource, providing the resource id and version. **Catalogue API:** • **put /catalogue: Updates a specific Catalogue** • **get /catalogue/all: Get a list of all Catalogues in the Portal.** • **post /catalogue/{catalogueId}/datasource: Creates a new Datasource for the specific Catalogue.** • **put /catalogue/{catalogueId}/datasource: Updates the Datasource of the specific Catalogue.** • **delete /catalogue/{catalogueId}/datasource/{id}: Deletes the Datasource of the specific Catalogue with the given id.** • **get /catalogue/{catalogueId}/datasource/{resourceId}: Returns the Datasource of the specific Catalogue with the given id.** • **post /catalogue/{catalogueId}/provider: Creates a new Provider for the specific Catalogue.** • **put /catalogue/{catalogueId}/provider: Updates the Provider of the specific Catalogue** • **get /catalogue/{catalogueId}/provider/all: Filter a list of Providers based on a set of filters or get a list of all Providers in the Catalogue.** • **delete /catalogue/{catalogueId}/provider/{id}: Deletes the Provider of the specific Catalogue with the given id.** • **get /catalogue/{catalogueId}/provider/{providerId}: Returns the Provider of the specific Catalogue with the given id.** • **post /catalogue/{catalogueId}/resource: Creates a new Service for the specific Catalogue.** • **put /catalogue/{catalogueId}/resource: Updates the Service of the specific Catalogue.** • **delete /catalogue/{catalogueId}/resource/{id}: Deletes the Service of the specific Catalogue with the given id.** • **get /catalogue/{catalogueId}/resource/{resourceId}: Returns the Service of the specific Catalogue with the given id.** • **post /catalogue/{catalogueId}/trainingResource: Creates a new Training Resource for the specific Catalogue.** • **put /catalogue/{catalogueId}/trainingResource: Updates the Training Resource of the specific Catalogue.** • **delete /catalogue/{catalogueId}/trainingResource/{id}: Deletes the Training Resource of the specific Catalogue with the given id.** • **get /catalogue/{catalogueId}/trainingResource/{resourceId}: Returns the Training Resource of the specific Catalogue with the given id.** • **get /catalogue/{catalogueId}/{providerId}/datasource/all: Get all the Datasources of a specific Provider of a specific Catalogue** • **get /catalogue/{catalogueId}/{providerId}/resource/all: Get all the Services of a specific Provider of a specific Catalogue** • **get /catalogue/{catalogueId}/{providerId}/trainingResource/all: Get all the Training Resources of a specific Provider of a specific Catalogue** • **get /catalogue/{id}: Returns the Catalogue with the given id.**

	Training Resource API: • **post /trainingResource: Creates a new TrainingResource.** • **put /trainingResource: Updates the TrainingResource assigned the given id with the given TrainingResource, keeping a version of revisions.** • **get /trainingResource/all: Filter a list of Training Resources based on a set of filters or get a list of all Training Resources in the Catalogue.** • **get /trainingResource/by/{field}: Get all Training Resources in the catalogue organized by an attribute, e.g. get Training Resources organized in categories.** • **post /trainingResource/validate: Validates the Training Resource without actually changing the repository.** • **get /trainingResource/{id}: Get the most current version of a specific Training Resource, providing the Resource id.** **EOSC Profiles Controlled Vocabulary API:** • **get /vocabulary/byType:** get all vocabulary entries organised per type. • **get /vocabulary/byType/{type}:** get a list of vocabulary entries for the given specific type. • **get /vocabulary/{id}:** get a vocabulary entry by id. • **get /vocabulary/countries/EU:**returns a list of EU countries. • **get /vocabulary/countries/WW:** Returns a list of WW countries.
EOSC Future components expected to interact with this component	• EOSC Marketplace: consumes registry (resource, providers) entries. • EOSC Marketplace: provides user events. • EOSC order Management: provides order events for resources of a provider. • EOSC IF registry: provides connections between registry resources and interoperability guidelines. • Provider Portal: provider onboarding, resource management, and EPOT operations over the service registry. • Marketplace: offering monitoring data via API. • EOSC Exchange systems: CRUD operations via API for updating \ syncing the registry.

Table 5-2: EOSC Service Provider Portal technical features

Component Name	EOSC Service Providers Portal
Short description	**The Service Provider Portal** offers functionality to users representing a Provider organisation, who wish to onboard their organisation and onboard resources in the EOSC Portal registry, to manage and customise the way offerings are presented to end users and finally to gain insights on a multitude of usage statistics, user-generated events and statistics collected in the Portal. Offered functionality include: • *Onboarding services*, which implements the EOSC portal onboarding process, i.e., the registration of a new provider and the registration of Resources managed by a provider. The onboarding services will target authenticated users who will be able to onboard either via a Web-based step-wise process or programmatically by using the EOSC registry's APIs. • *Resource management service*: It offers the functionality for users to view the list of resources assigned to their organisation and manage all characteristics of their offerings. Resource management will also enable users to 'activate'/'deactivate' a resource in the EOSC Portal, to assign it to categories or other classification schemes (e.g., scientific domain, TRL, etc.), to manage the different versions of a resource and add new users who will be responsible for managing the offerings of a provider. • *Providers Dashboard*: it serves as the UI entry point of a provider's users. The dashboard offers an overview to the users for the list of providers, it represents the list of resources and their properties including a history of changes applied to a resource. The dashboard will also give access to a rich set of statistics, which are

	collected by the EOSC portal and will be associated with resources and providers onboarded in the EOSC Registry. These include usage statistics, such as views and visits for a resource, aggregated views and visits for all resources offered by a provider, search-related statistics which are associated with a provider, orders placed for a resource, ratings, recommendations offered to users related to a resource and finally favourites that users add in the EOSC portal. • *Statistics* over the content of the registry, organised by resources and providers, such as number of resources per scientific discipline, providers per country, etc.
Main features	For providers: • Onboarding; • Management of resources; • Live usage statistics from the EOSC Portal; • Email notifications; • Interaction with EPOT team. For EPOT team members: • Onboarding management; • Auditing and registry management; • Email digest and interaction with providers. For other users (funders EOSC profiles management (vocabularies, schema, etc): • Statistics.
URLs	• EOSC Provider entry point https://providers.eosc-portal.eu/ • EOSC Provider Sandbox https://sandbox.providers.eosc-portal.eu/ • EOSC Provider Beta https://beta.providers.eosc-portal.eu/ • EOSC portal API: https://providers.eosc-portal.eu/openapi • EOSC API base URL: https://api.eosc-portal.eu/
Documentation	• UI Documentation: https://eosc-portal.eu/providers-documentation/eosc-provider-portal-basic-guide • API: https://providers.eosc-portal.eu/developers • GitHub: https://github.com/madgeek-arc/resource-catalogue
EOSC Future components expected to interact with this component	• *EOSC Service Registry APIs:* CRUD operations to Service registry • *EOSC Marketplace:* consumes registry (resource, providers) entries • *EOSC Marketplace*: provides user events • *EOSC Order Management*: provides order events for resources of a provider • *EOSC IF registry:* provides connections between registry resources and interoperability guidelines • *Marketplace*: offering monitoring data via API • *Third party EOSC Catalogues*: CRUD operations via API for updating/syncing the registry

Table 5-3: OpenAIRE Graph technical features

Component Name	**OpenAIRE Research Graph**
Short description	The Graph is a public, open access, collection of metadata and semantic links (~1bn) between research-related entities collected from data sources or inferred from PDFs (14m+). Entities are *research products*, namely publications (144m+), datasets (58m+), software (330K+), and other research products (6.5m+), organisations (250 000), *funders* (~25), *funding streams, projects* (3M+), , and *data sources* (~125k).
Main features	*Content acquisition* • The Graph content is constantly (scheduled jobs) aggregated from the data sources. • Data sources are of different kinds: publishers, data repos, databases (PDB, ENA, etc.), institutional repos, thematic repos (ArXiv, EuropePMC, Repec, etc), preprint servers, software repos, entity registries (ORCID, ROR, Crossref, DataCite, GRID.ac, OpenDOAR, re3data, PDBs, etc). • The Graph index is published monthly, being the result of harmonising, deduplicating, and enriching the harvested metadata by FTM. • Selected data sources, while still being regularly aggregated, can be kept in-sync real-time with the index.

	Access to content • The Graph is open and accessible via APIs and data dumps in Zenodo. • The Graph is accessible via discovery portals for communities and generic users.
URLs	• Graph informative site: http://graph.openaire.eu • Graph discovery website: http://explore.openaire.eu (all users), http://connect.openaire.eu (scientific communities) • Graph APIs: https://develop.openaire.eu • Graph dataset dumps: https://develop.openaire.eu/graph-dumps.html
Documentation	• https://develop.openaire.eu
API types and use-cases	• Resolution of records *by id/PID*: website widgets (e.g. project websites to collect related publications), European Commission Participant Portal to recommend project publications to project coordinators. • Query on APIs to collect list of records: DSpace and ePrints platforms to collect list of projects from deposition UIs. • Download of data dumps: Elsevier, Springer, researchers, download the data to integrate with their services or perform data analysis.
EOSC Future components expected to interact with this component	*Graph as a consumer: data sources* • *EOSC Service registry*: services to become entities of the graph (extension of the current 'data source' type); regular harvesting, but also in-sync with Graph index (updates to the registry are sent to the index). • EOSC IF database: regular harvesting of EOSC IFs as research products (identify proper resource product type). • Data Usage Statistics (downloads and views of research products): explore the possibility to include stats as part of the information indexed with the data, so as to enable discovery based on stats. • AI components: embedding as part of the graph index the information to ensure AI discovery is supported (this may require the realisation of extra indexing structures). Components consuming the Graph • Marketplace: enabling discovery of EOSC resources via APIs. • EOSC Portal (consumers/providers): enabling discovery of EOSC resources via APIs. • AI components: support data provision to enable AI-supported discovery.

Table 5-4: OpenAIRE PROVIDE technical features

Component Name	OpenAIRE PROVIDE
Short description	The OpenAIRE PROVIDE is a one-stop shop web service where content providers (repository, data archive, journal, aggregator, CRIS system) interact with OpenAIRE. It provides the front-end access to many of OpenAIRE's backend services: register - validate data sources; enrich the metadata or the content of the data source; assess - subscribe to the OpenAIRE Usage Statistics service.
Main features	*Validation & registration* The OpenAIRE Validator service allows users to test their repository's compatibility with the OpenAIRE Guidelines. The set of supported guidelines can be easily modified and adapted to new needs and use cases. https://www.openaire.eu/validator-registration-guide. *Enrichments* Content enrichment is a powerful tool that enables Content providers to enrich research artefacts with additional metadata through the OpenAIRE Notification Broker Service. https://www.openaire.eu/content-enrichment-guide. *Usage statistics* OpenAIRE's Usage Statistic service uses the Matomo open-source analytics platform (matomo.org) to track usage activity. It collects and analyses usage data from the network of OpenAIRE data providers and exploits usage metrics like downloads and metadata views. https://www.openaire.eu/guides-usage-statistics.
URLs	https://provide.openaire.eu
Documentation	https://provide.openaire.eu/about
API types and use-cases	No APIs offered to end users

EOSC Future components expected to interact with this component	*Usage Stats Service* Provides the usage counts to display in the Provide portal. Research Graph Provides statistics about the graph and also consumes the registered products.

5.2 EOSC Service Order Management & Composition

The main prerequisite for setting up ordering for a specific resource is the completion of resource onboarding in the EOSC Portal. The onboarding process, supporting user interfaces[8] and APIs that can be used to onboard resources, are covered by other interoperability guidelines and other providers' documentation[9]. Each resource can be configured to allow for direct ordering in EOSC Portal via the EOSC Marketplace. Two ways to manage the ordering configuration are available for the use of resource providers. Ordering may be configured either via the EOSC Marketplace user interface for authenticated provider administrators or via the Marketplace offering API. Both methods allow for creating resource offers, configuring technical order parameters and selecting order management systems that will be used to handle resource orders. In order to process the resource orders, a provider can choose the desired method from a selection of tools following the interoperability guidelines for order management. The *Service Order Management Back Office* (SOMBO) is available for all EOSC resource providers and allows them to instantly manage resource orders. Other interoperability patterns to support other provider-specific or provider community-specific order processing methods exist though. The next paragraphs describe the process of order management integration and interoperability patterns that can be selected to integrate other order management systems.

The configuration of resource ordering allows to instantly manage resource orders with the use of the service management back office. Nevertheless, the ordering process is much more flexible and offers integration methods allowing to meet specific needs of resource providers. The process of preparing integration for other order management systems is independent of resource ordering configuration and may happen in parallel. The first step of integration is to choose the desired integration method from a selection of the available interoperability patterns (see the functional specification). The main entity that allows to set up the integration with the EOSC Marketplace to make other systems interoperate with the ordering process, is an instance of the Order Management System (OMS) registration. An OMS registration is an EOSC Marketplace entity used for authentication and authorisation in the Marketplace Ordering API - documentation available from https://marketplace.eosc-portal.eu/api_docs?subsection=integration_methods. Once the registration is complete, one of the described interoperability patterns may be utilised for order processing by the integrating OMS. The newly registered OMS will also be available for certain providers (or provider communities, depending on the OMS type), and the Marketplace will allow to configure it as the resource OMS while setting up the resource ordering.

The flexible EOSC Marketplace ordering API allows for different levels of integration by third parties, fitting their needs and expectations. Three integration levels, as sketched in Figure 5.2, are presented below and cover most of the identified use-cases.

[8] https://providers.eosc-portal.eu/

[9] https://eosc-portal.eu/providers-documentation

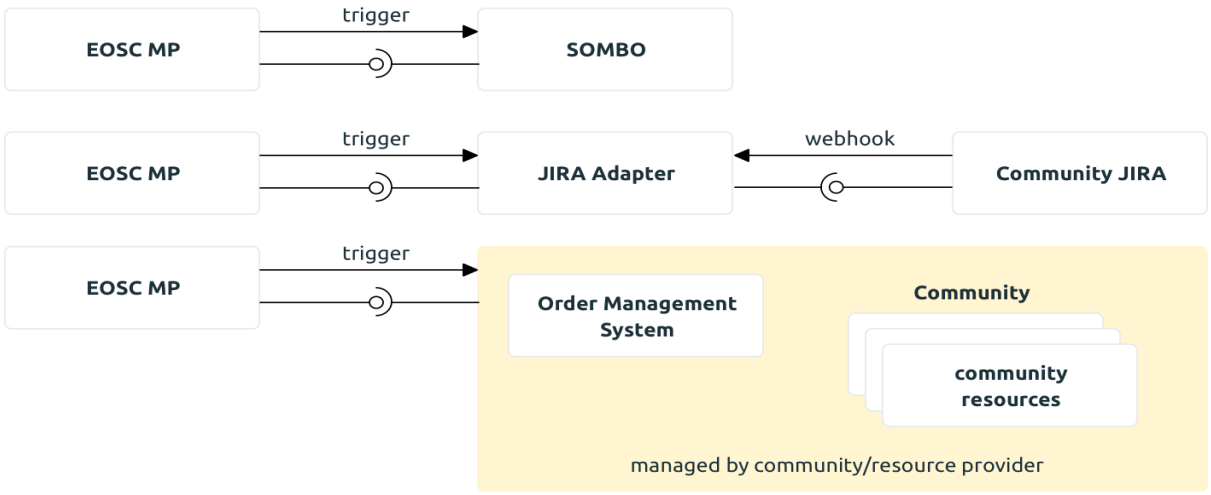

Figure 5.2: EOSC Service Order Management and Composition

As mentioned earlier, the EOSC Marketplace offers flexible ways to integrate ordering to improve interoperability. It has an API that is general enough to handle various provider use cases. The integration interface consists of a REST API and an optional asynchronous update trigger endpoint.

There are several ways for providers to integrate their order management processes. One is to use SOMBO to handle user orders. This doesn't incur any implementation cost.

A provider can also use a reference Jira adapter[10] to integrate their Jira instance. This implementation can also be used as a reference to facilitate implementing in-house integrations, such as in the last case. A community integrates their own existing OMS with the EOSC Marketplace ordering process there.

To ensure interoperability, it is paramount that all involved systems are compatible representations of projects, project items (mostly but not exclusively orders) and messages that are general enough to meet their needs. The EOSC Marketplace strives to make as few assumptions about the exchanged data as possible, so as to allow differing use cases.

All the mentioned integration patterns are elaborated in the following subsections.

5.2.1 Service Order Management Back-Office (SOMBO)

The Service Order Management Back Office (SOMBO) is a tool designed to track all the orders received by the Marketplace and to propose different actions on these orders. The workflows and connections with other components are shown in Figure 5.3.

[10] https://github.com/cyfronet-fid/oms-adapter-jira/

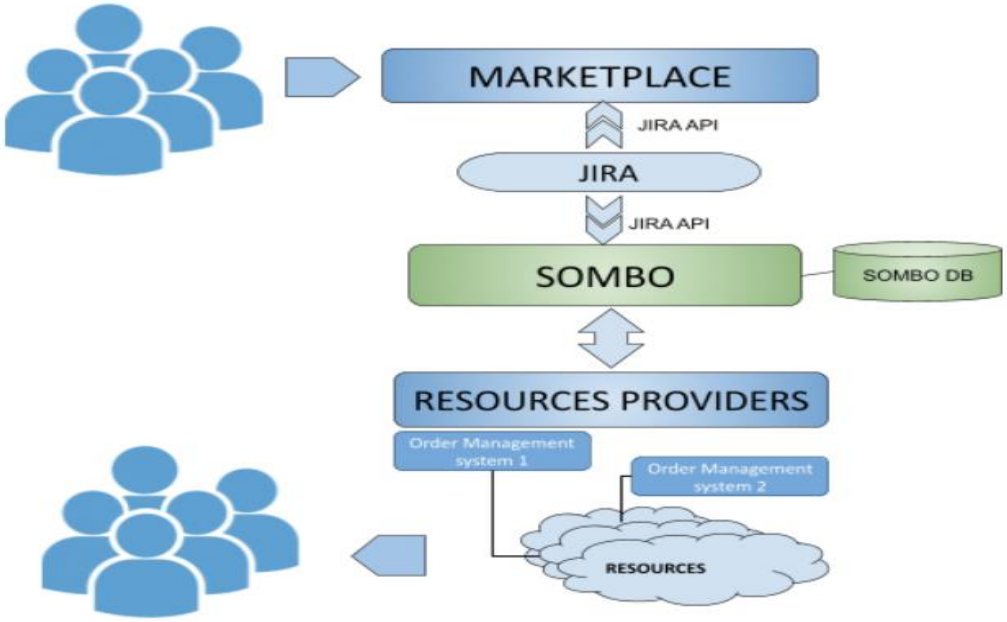

Figure 5.3: SOMBO high-level architecture

For a new order, the tool allows shifters to:

- Analyse the orders;
- Approve/reject the orders;
- Request more information from the customer;
- Exchange information between third parties;
- Start a negotiation process;
- Automatically generate an SLA/OLA from given templates;
- Coordinate (semi-) automatic procedures to enable a customer to access a service.

5.2.2 Order Management process adapters for existing Order Management Systems / reference Jira adapter

Using an OMS adapter is an interoperability pattern utilising a simple service as a mediator in communication between EOSC Marketplace and an existing, possibly closed-source, system (e.g. ticketing system) to manage resource orders. Reference implementation supports integration with the Jira system, but any integrator can develop its own compatibility layer with its current ticketing system.

EOSC Marketplace asynchronously informs the adapter about any new activity using an HTTP trigger. Every other communication attempt is handled proactively by the OMS adapter. Usually, a ticket management system with which integration occurs informs an OMS adapter to notify the Marketplace about changes, then the adapter calls the Marketplace's API. In case a change occurs in the Marketplace (new Project / Project Item is created, Message is posted) the Marketplace triggers OMS adapter (every registered OMS adapter has a corresponding trigger configuration) and the adapter performs any necessary calls to the integrating system (ticketing system) to enact that change.

It is in the scope of the OMS adapter to store mappings between Marketplace's ids and ticketing system ids.

The reference implementation based on the JIRA system[11] contains a Marketplace API client which is responsible for communication with the Marketplace ordering API. It also contains a thin translation layer which maps project Item parameters to fields in Jira. The core philosophy of the system is that it should not, under any circumstances, lose information about ordering changes. To fulfil this goal, all actions are scheduled in the underlying message queue (Celery). This way they can be retried if network errors occur, or in case they fail. They can also be reviewed manually by an operator.

[11] https://github.com/cyfronet-fid/oms-adapter-jira

5.2.3 Marketplace API for Order Management Systems

The most sophisticated integration pattern is to implement direct communication with the EOSC Marketplace. In order to support use cases specific to a certain provider or community of providers, the Marketplace ordering API can be used to manage resource orders. This solution is suitable for setting up interoperability between EOSC Marketplace and other existing marketplaces, gateways, infrastructures and services that need the highest flexibility and deepest integration with EOSC Portal. This integration pattern allows providers to create tailor-made resource order-handling solutions and map sophisticated resource provisioning workflows to EOSC Portal-based interaction with its end-users.

As before, the reference implementation should help with integration[12]. It utilises a more general EOSC Marketplace API client, that is the preferred way to integrate[13]. The technical features of the Marketplace components are described in Table 5-5.

Table 5-5: EOSC Marketplace technical features

Component Name	EOSC Marketplace

Main features	• Possibility to offer resources through the EOSC channel
	• Dedicated panels and provider supporting functionalities
	• Functionalities supporting the connection of existing catalogues or order management systems
	• Increased (demonstrable) impact of resources delivered (e.g. order statistics) –> additional funding
	• Access to EOSC resources
	  ○ Discovery
	  ○ Information
	  ○ Access mechanism
	• EOSC support and expertise
	  ○ User requests
	  ○ EOSC experts support
	  ○ EOSC providers support
	• Dedicated functions and panels
	  ○ User profile
	  ○ Marketplace projects
	• Possible access to statistics connected with the interest around EOSC resources
	• Additional source of information about the exploitation of the EOSC resources
	• Lower the marketing and transaction costs considerably compared to targeting individual research institutes or researchers.
	• Exposure to EOSC offerings that might enrich enterprise services

URLs	• Service URL: https://marketplace.eosc-portal.eu • Source code repository: https://github.com/cyfronet-fid/marketplace • API documentation: https://marketplace.eosc-portal.eu/api_docs • API documentation (OpenAPI standard): https://marketplace.eosc-portal.eu/api_docs/swagger/ • API endpoint: https://marketplace.eosc-portal.eu/api • Reference OMS adapter for JIRA system: https://github.com/cyfronet-fid/oms-adapter-jira • Python API client: https://github.com/cyfronet-fid/marketplace-api-client
Documentation	• https://eosc-portal.eu/using-the-portal • https://www.youtube.com/watch?v=T2G70yZ52Kc

API types and use-cases	• Offering API
	  ○ Supporting providers' capabilities to publish information about the services'/resources' offers via a dedicated API. No need to use user interface for the sake of showing the consumers the service/resource offers
	  ○ Allows to create/update/delete the offers
	  ○ Allows to manage the technical parameters of te offers which are essential for the order management process
	• Ordering API

[12] https://github.com/cyfronet-fid/oms-adapter-jira
[13] https://github.com/cyfronet-fid/marketplace-api-client

	○ API allowing to retrieve information about the customer orders ○ Order status handling ○ Channel for the user support (user/provider/OMS team messages exchange) ○ Passing user details (credentials, SLAs etc) ○ Passing MP project details (information about the scientific use case the customer is bringing)
EOSC Future components expected to interact with this component	EOSC Recommender System EOSC Research Graph EOSC Service Registry

5.3 EOSC Statistics and Open Science Statistics

EOSC statistics will be exposed via the EOSC portal to show Open Science trends and indicators, metrics about portal usage, accounting and usage of services and research products. This section describes the technical feature of such components, described respectively in Table 5-6, Table 5-7, Table 5-8 and Table 5-9 .

5.3.1 Open Science Monitor

Open Science monitor back-end and UIs will be integrated into the EOSC portal by interoperating with the OpenAIRE Open Science observatory services, whose technical features a described in Table 5-6.

Table 5-6: OpenAIRE Open Science Observatory technical features

Component Name	**Open Science Monitor**
Short description	The Open Science Observatory presents a collection of indicators and visualisations that help interested stakeholders (policy makers and research administrators, among others) better understand the Open Science landscape in Europe across countries and (coming soon) subject areas. The platform assists the monitoring, and consequently the improvement, of open science policy uptake across different dimensions of interest, revealing weak spots and hidden potential. Based on the OpenAIRE research graph, following open science principles and an evidence-based approach, the indicators can be used to provide timely and reliable insights on the evolution of open science in Europe and assist in promoting good practices.
Main features	End users: • Provide indicators and visualisations of the Open Science landscape in Europe
URLs	https://osobservatory.openaire.eu
Documentation	https://osobservatory.openaire.eu/methodology
API types and use-cases	*Not in place*
EOSC Future components expected to interact with this component	As a consumer: • OpenAIRE Graph, to get data for all offered indicators

5.3.2 EOSC Metrics Portal

The EOSC Metrics portal is a portal aggregating data from Google Analytics and the Jira Helpdesk system. These figures are consolidated and provided through different tables and reports. Its technical features are described in Table 5-7.

Table 5-7: EOSC-Hub Metrics Portal technical features

Component Name	**EOSC-Hub Metrics Portal**
Short description	The Metrics portal is a portal aggregating data from Google Analytics and the Jira Helpdesk system. These figures are consolidated and provided through different tables and reports.
Main features	• Statistics about Service Orders • Statistics about Service Providers • Statistics about usage (access, pages, views) of the EOSC Portal and the EOSC Marketplace
URLs	https://opsportal.eosc-portal.eu/metricsEOSC
Documentation	https://gitlab.in2p3.fr/opsportal/eosc-opsportal/-/wikis/EOSC-Metrics
API types and use-cases	Not in place

EOSC Future components expected to interact with this component	All statistics will be provided through API calls

5.3.3 EOSC Service Accounting

5.3.3.1 Service Accounting

The EOSC Accounting Service is a platform designed to efficiently collect, aggregate, and exchange metrics across various infrastructures, providers, and projects. The system provides a REST API, which accepts input from diverse resources, stores it in a database, and aggregates the incoming data. It also offers an intuitive user interface that allows clients to interact with the platform and access accounting data for specific time periods. All API resources are only accessible to authenticated clients, ensuring secure access to sensitive data.

5.3.4 Fundamentals - Elements of the Service

The Accounting Service is a system that provides a framework for organising and managing accounting data for a specific project, provider or installation. It involves various roles such as Project Admin, Provider Admin, and Installation Admin, each with their specific set of responsibilities.

One of the key elements of the Accounting Service is Metrics, which are quantitative measures used to assess and track the performance or usage of a service. A Metric Definition is a way of representing and describing the type of metric.

In the Accounting Service, Metric and Unit Type are essential components that allow for the collection and tracking of various types of Metrics. These types are included as part of the Metric Definition and provide greater flexibility and specificity in the Metrics that the Accounting System can collect. A Metric Type defines how physical quantities are collected over time, while a Unit Type expresses and measures physical quantities used in various infrastructures, service providers, and projects. Together, these elements enable users to gather and analyse data at different levels of granularity and with different units of measurement.

The Accounting System allows the client to create a hierarchical structure consisting of three levels: Project, Provider, and Installation. The Project is the main resource of the system, and clients can create it by registering their unique European Project ID. Providers are organisations that offer at least one installation to a specific project, and they can be registered through the EOSC Resource Catalogue or created directly through the Accounting Service. Installations are specific instances or parts of a resource/service that are allocated to a specific project by a Provider. An Installation can be associated with a Resource, which can be retrieved from the EOSC Resource Catalogue. The Accounting System communicates with the Resource Catalogue to retrieve the available Resources and stores specific information about a Resource.

This hierarchical structure allows clients to organise and manage accounting data for their project and enables them to easily view and analyse metrics at different levels of granularity.

Table 5-8: EOSC Accounting for Services technical features

Component Name	**EOSC Accounting for Services**
Short description	The EOSC Accounting for Services is a platform designed to efficiently collect, aggregate, and exchange metrics across various infrastructures, providers, and projects.
Main features	The key functionalities offered by the EOSC Accounting Service are: • Efficient collection, aggregation, and exchange of metrics. • REST API that accepts input from diverse resources. • Database storage and aggregation of incoming data. • Intuitive user interface for accessing accounting data. • Secure access to sensitive data through authenticated clients.
URLs	https://accounting.eosc-portal.eu
Documentation	https://argoeu.github.io/argo-accounting/
API types and use-cases	https://argoeu.github.io/argo-accounting/openapi/explore
EOSC Future components expected to interact with this component	AAI, Service Registry, Messaging

5.3.4.1 EOSC Usage Statistics for research publications, data and software

Figure 5.4 shows the high-level architecture of the EOSC Data Usage Statistics services, powered by OpenAIRE UsageCounts services. UsageCounts collects usage data from EOSC data sources, such as institutional/thematic repositories, journals, data repositories, aggregators, discovery portals, etc. Then, it aggregates them, delivering standardised activity reports about research usage and uptake.

Two approaches are exploited for the collection of usage data, named PUSH and PULL, both depicted in Figure 5.4. Push is the default workflow offered by the UsageCounts service in OpenAIRE and allows server sidetracking of events. Open Access repositories embedded tracking code in the form of DSpace plugins, EPrints patches, or generic tracking software that exploit Matomo's analytics platform HTTP API. Usage Activity is tracked and logged at Matomo platform in real time. Information is transferred offline, using Matomo's API, to OpenAIRE's DBs for further processing using the COUNTER Code of Practice and statistical analysis. Statistics are subsequently deployed via OpenAIRE's Portal, OpenAIRE's Repository Dashboard or Sushi-Lite API endpoint.

A different approach for the UsageCounts service, named Pull and also depicted in Figure 5.4, allows data providers or usage statistics aggregation services (e.g. IRUS-UK) to offer a bulk download method for the usage data. In particular, the Pull approach supports the gathering of consolidated statistics reports using other protocols such as SUSHI-Lite. These statistics are also stored into OpenAIRE's database for statistical analysis and are deployed via OpenAIRE's Portal, OpenAIRE's Repository Dashboard, or Sushi-Lite API.

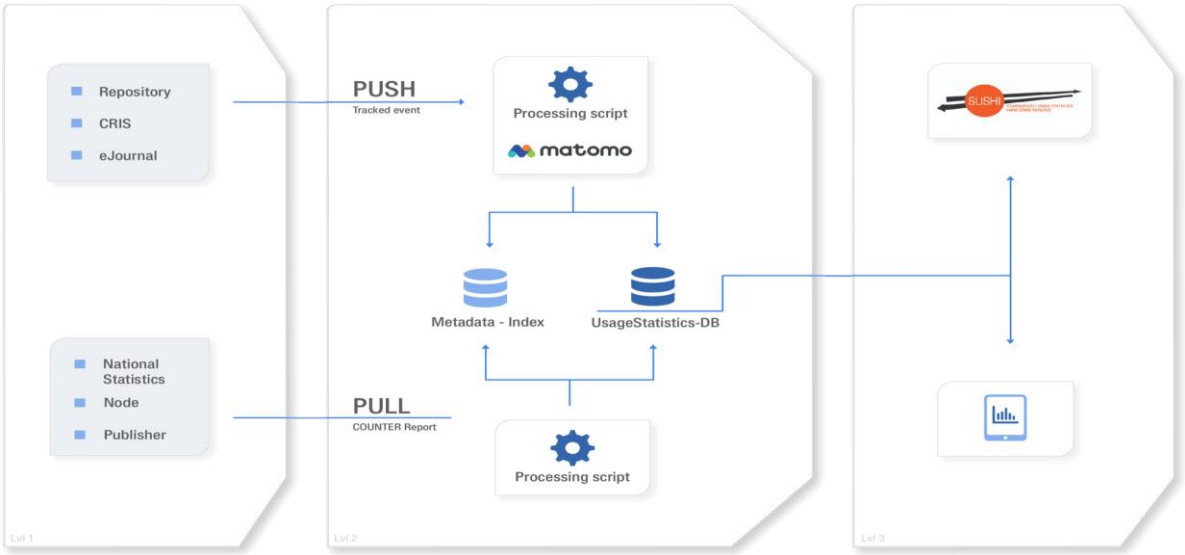

Figure 5.4: Data Usage Statistics for the EOSC: UsageCounts high-level architecture

Table 5-9: OpenAIRE UsageCounts technical features

Component Name	OpenAIRE UsageCounts
Short description	The OpenAIRE UsageCounts service is the usage statistics service of OpenAIRE Research Graph. UsageCounts gathers raw usage activity and consolidated usage statistics reports, respectively, for OpenAIRE Research Graph products and from the network of OpenAIRE content providers (repositories, e-journals, CRIS, national aggregators, etc.). This is realised by utilising open standards and protocols and exploiting reliable, consolidated and comparable usage metrics like counts of products downloads and metadata views conformant to COUNTER Code of Practice.
Main features	Content acquisition PUSH Approach: • Server side tracking: Plugins (Dspace), patches (Eprints), Generic Matomo Tracker script (for all platforms) using Matomo's Web Analytics HTTP API. • Usage activity is tracked and logged in real time.

	• Information is transferred offline to OpenAIRE's databases for aggregation with metadata information and statistical analysis. PULL Approach: • Gathering of consolidated statistics reports from aggregation services, such as IRUS-UK, using protocols such as SUSHI-Lite. • Information is transferred offline to OpenAIRE's DBs for aggregation with metadata information and statistical analysis. *Access to Content* • Usage statistics are deployed via OpenAIRE's Portals or a Sushi-Lite API.
URLs	• UsageCounts informative website: http://usagecounts.openaire.eu • UsageCounts registration website: http://provide.openaire.eu • UsageCounts discovery website: http://explore.openaire.eu • UsageCounts SushiLite APIs: https://usagecounts.openaire.eu/resources#apis
Documentation	• https://openaire.github.io/usage-statistics-guidelines/
API types and use-cases	*COUNTER CoP R4 Reports* • **AR-1 - Article Report 1**[14]**.** This report enables you to view the number of successful article download requests by month and repository. • **IR-1 - Item Report 1**[15]**.** This report enables you to view the number of successful item download requests by month and repository identifier. • **JR-1 - Journal Report 1**[16]**.** This report enables you to view the number of successful full-text article requests by month and journal. • **RR-1 - Repository Report 1**[17]**.** This report enables you to view the number of successful item downloads for all repositories participating in the usage statistics service. • **BR-1 - Book Report 1**[18]. This report enables you to view the number of successful book title requests by month and title • **BR-2 - Book Report 2**[19]**.** This report enables you to download the number of successful book section requests by month and title.
EOSC Future components expected to interact with this component	UsageCounts as a consumer: data sources • EOSC Service registry: services to become entities of the graph (extension of the current 'data source' type); • EOSC IF database: EOSC IFs as research products; • Monitor & Dashboards: downloads and views of research products. Components consuming the UsageCounts: • Marketplace: enabling discovery of EOSC resources usage via APIs; • EOSC portal (consumers/providers): enabling discovery of EOSC resources usage via APIs.

5.4 EOSC Helpdesk

After a thorough analysis of multiple helpdesk technologies (see D4.2a and D4.2b) the new technology Zammad was chosen and deployed in the EOSC Future project. This technology has several benefits compared to the previously used technology for the EOSC Helpdesk which are:

- Open source product;
- Modern and responsive interface for customers and agents;
- Support of REST API;
- Knowledge base associated with helpdesk with full-text search, rich text editor;
- Chat functionality for live support.

[14] https://usagecounts.openaire.eu/sushilite/AR1
[15] https://usagecounts.openaire.eu/sushilite/IR1
[16] https://usagecounts.openaire.eu/sushilite/JR1
[17] https://usagecounts.openaire.eu/sushilite/RR1
[18] https://usagecounts.openaire.eu/sushilite/BR1
[19] https://usagecounts.openaire.eu/sushilite/BR2

Figure 5.5 shows the high-level technical architecture of the Helpdesk based on the new technology. In addition, the bulk of multiple integrations with other systems and community helpdesk have been added.

The main functional units of the EOSC Helpdesk are:

- Multiple submission channels: Helpdesk main Portal, Webforms, E-mail channels.
- Helpdesk Back Office: the core component of the EOSC Helpdesk which implements all major helpdesk functions like ticket management, group management, user management, smart search, analytics and reporting etc.
- Integration modules: enable integration with multiple services and community helpdesks.
- Helpdesk Community Portals: enable delivery of the Helpdesk as a Service for communities.

The EOSC helpdesk has been integrated with EGI GGUS and multiple community helpdesks during the course of the project. Any webpage of the EOSC service could include a submission webform for request as currently implemented on EOSC Portal webpage and Marketplace. Upon community request, the helpdesk can be provided as a service for any EOSC community with a dedicated helpdesk portal with multiple customizations.

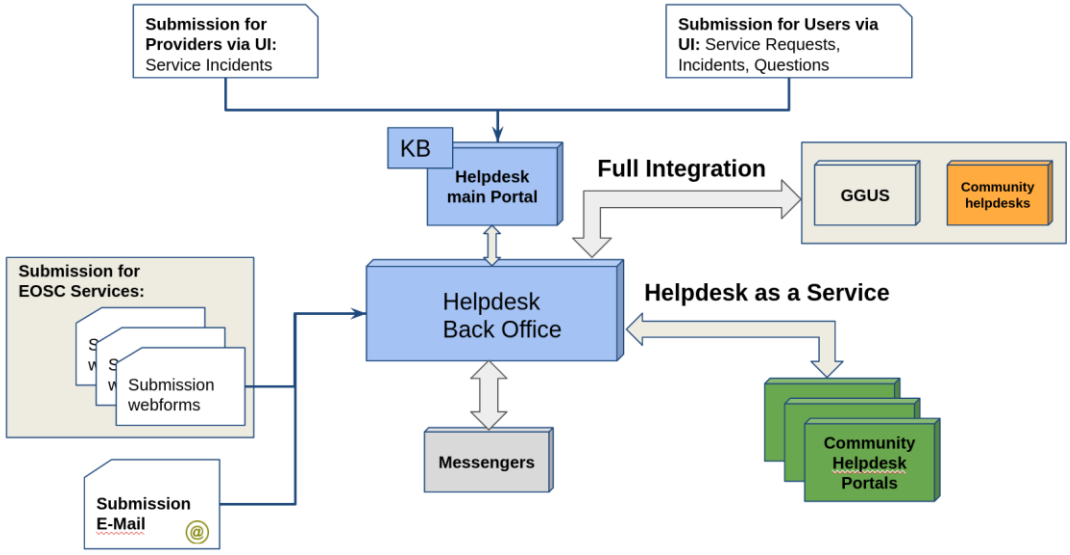

Figure 5.5: High-level architecture and integrations of the EOSC Helpdesk based on the new technology.

Short technical descriptions of the component (the current or new adopted tool) are provided in Table 5-10.

Table 5-10: EOSC-Hub Helpdesk technical features

Component Name	EOSC Helpdesk
Short description	The EOSC helpdesk is a single point of contact for all EOSC customers for help and support. It is provided for EOSC users to request and obtain support for any problem or issue, and for the EOSC communities as a service to implement user support and incident management.
Main features	The features of the EOSC Helpdesk can be grouped into two target groups: users and helpdesk internal agents or teams. Main features offered to the user are: • Creation of a ticket for any of the EOSC services (Core and Exchange); • Display all the tickets created by the owner; • Find a previously created ticket; • Notify the user of answers and changes to the tickets; • Access integrated with the EOSC Portal AAI system; • Helpdesk knowledge base. Features offered to the helpdesk teams are: • Notification when a new ticket is created; • Classification of the tickets;

	• Escalation of the tickets; • Creation of a new support unit with the assignation of an administrator role to specific users; • Management of incident or disruption of Hub services; • Interface for communicating with other service providers ticketing systems; • First level support for EOSC integrated services as a service; • Submission of the tickets via email to any support unit; • Chat functionality with users; • Helpdesk knowledge base.
URLs	https://helpdesk.eosc-portal.eu/
Documentation	(helpdesk based on GGUS technology) https://ggus.eu/?mode=docu https://wiki.egi.eu/wiki/GGUS
API types and use-cases	https://wiki.egi.eu/wiki/GGUS:SOAP_Interface_FAQ
EOSC Future components expected to interact with this component	EOSC CMDB EOSC Monitoring Web portals of EOSC services Community Helpdesks

5.5 EOSC Monitoring

EOSC Monitoring is based on the ARGO Monitoring Service[20] provides a flexible and scalable framework for monitoring the status, availability and reliability of a wide range of services provided by infrastructures with medium to high complexity. ARGO generates reports using customer-defined profiles (e.g. for SLA management, operations, etc.). During report generation, ARGO takes into account custom factors such as the importance of a specific service endpoint and scheduled or unscheduled downtimes. The foundations of the ARGO Monitoring Service are:

- Sources of truth - registries containing information about what should be monitored and how the monitoring should be performed. (Supports predefined format in csv, JSON & XML).
- Configuration management database is a registry which contains information about the topology of the infrastructure - entities such as sites, service endpoints, entity organisation (groups, hierarchies) and contact information of users responsible for operations. (Supports predefined format in csv, json & xml).

Management teams can monitor the availability and reliability of the services from a high-level view down to individual system metrics and monitor the conformance of multiple SLAs. The dashboard design enables easy access and visualisation of data for end-users. APIs are also supported so as to allow third parties to gather monitoring data from the system.

The main components of ARGO are:

- **Connectors:** Through the use of custom connectors, the monitoring service can connect to multiple external Configuration Management Databases and Service Catalogues.
- **POEM**: POEM represents a central dashboard (UI) for managing most of the resources used in the Monitoring engine. It follows the principle towards One-Stop-Shop functionality and using web API as a centralised source of truth. It is the central management interface for a provider to prepare all the resources for the monitoring.
- **Analytics Engine:** A powerful and scalable analytics engine built on top of Apache Flink and HDFS. The analytics Engine is responsible for the aggregation of the status results and the computation of availability and reliability scores of composite services using customer-defined algorithms. It supports stream processing in real-time as one of the key features. Monitoring results flow through the Messaging Service, to the streaming layer (in parallel to the HDFS). The streaming layer is used in order to push raw metric results to the metric result store and to compute status results and push them to the status store in real-time.

[20] https://argoeu.github.io/argo-monitoring/

- **Web API**: The Web API provides the serving layer. It consists of a high-performance and scalable data store and a multi-tenant REST HTTP API, which is used for retrieving the status, availability and reliability reports and the actual raw metric results.
- **Web UI:** The default web UI is based on the Lavoisier Data Aggregation Framework.
- **Notification Service:** The service responsible for sending real-time status events to Service Owners in case of service or service component failure or recovery.

The detailed technical features are listed in Table 5-11.

Table 5-11: EOSC Monitoring technical features

Component Name	EOSC Monitoring - ARGO service availability and reliability monitoring
Short description	EOSC Monitoring is based on ARGO which provides monitoring of services, visualisation of their status, dashboard interfacing, notification system and generation of availability and reliability reports. The dashboard design enables easy access and visualisation of data for end-users. Third parties can gather monitoring data from the system through a complete API. Central deployment of the ARGO monitoring engine can serve a large infrastructure reducing the maintenance costs.
Main features	The key features of ARGO Monitoring Service are: • Multiple availability and reliability reports per tenant; • Multiple tenants; • High availability deployment (upon request); • Loosely coupled: support APIs in the full stack so that components are independent in their development cycles; • Support for topology configurations, metrics and profiles to add flexibility and ease of customisation; • Real-time notifications; • Customer-defined thresholds; • Aggregation of different monitoring sources; • Identification of Service trends.
URLs	• EOSC Exchange monitoring: https://monitoring.eosc-portal.eu • EOSC Core monitoring: https://monitoring-core.eosc-portal.eu • EOSC Core Status https://status.eosc-portal.eu • API: https://api.argo.grnet.gr
Documentation	• UI documentation: https://monitoring.eosc-portal.eu/eosc/documentation • http://argoeu.github.io/overview/ • Web API: https://argoeu.github.io/argo-web-api/ • POEM: https://argoeu.github.io/poem/v1/ • Web API: https://argoeu.github.io/api/v2/
API types and use-cases	API Calls and examples: **Results API:** • Availability & reliability results: https://argoeu.github.io/argo-web-api/docs/ar_results • Availability & reliability status timelines: https://argoeu.github.io/argo-web-api/docs/status_results • Individual metric results: https://argoeu.github.io/argo-web-api/docs/metric_results • Issues explorer: https://argoeu.github.io/argo-web-api/docs/issues • Trends https://argoeu.github.io/argo-web-api/docs/trends • Recomputation requests: https://argoeu.github.io/argo-web-api/docs/recomputations **Topology API:** • Topology statistics: https://argoeu.github.io/argo-web-api/docs/topology_stats • Topology of endpoints:

	https://argoeu.github.io/argo-web-api/docs/topology_endpoints • Topology of higher-level groups: https://argoeu.github.io/argo-web-api/docs/topology_groups **Profiles & Reports API:** • Operations profiles: https://argoeu.github.io/argo-web-api/docs/topology_stats • Metrics profiles: https://argoeu.github.io/argo-web-api/docs/topology_endpoints • Aggregation profiles: https://argoeu.github.io/argo-web-api/docs/topology_groups • Threshold profiles: https://argoeu.github.io/argo-web-api/docs/topology_groups • Reports: https://argoeu.github.io/argo-web-api/docs/reports • Weights: https://argoeu.github.io/argo-web-api/docs/weights • Downtimes: https://argoeu.github.io/argo-web-api/docs/downtimes
EOSC Future components expected to interact with this component	• EOSC Service Registry: will act as topology source for Argo • EOSC IF database: EOSC IFs as monitoring data • Dashboards: offering monitoring data via API • Marketplace: offering monitoring data via API • EOSC portal (consumers/providers): will act as topology source for Argo

5.5.1 EOSC Monitoring for Core Services

EOSC Core is the set of services required in order to provide a viable EOSC. EOSC-Core assembles all the essential elements to operate and provide the means to discover, share, access and reuse data and services. These elements address key technical, cultural and policy decisions of EOSC and must be maintained over the long term. The goal of EOSC Core monitoring is to monitor all these services that are part of the EOSC Core so that they are reliable to fulfil their purpose as part of it. The services of the EOSC Core are monitored via the ARGO Monitoring Service. The EOSC Core Monitoring gets its topology from http://gocdb.eosc-portal.eu/ and monitors each component of EOSC Core services accordingly. The results are presented at https://monitoring-core.eosc-portal.eu & https://status.eosc-portal.eu and are available (with authorisation) from https://api.argo.grnet.gr. In order for a service to be monitored by EOSC Core Monitoring, one needs to follow a predefined procedure (register service, associate metric etc.).

5.5.2 EOSC Monitoring for Exchange Services

EOSC-Exchange is a set of services built on the EOSC Core to ensure a rich set of services (common and thematic) exploiting FAIR data and encouraging its reuse are available to publicly funded researchers. The goal of EOSC Exchange Monitoring is to check the availability and reliability of services to assist researchers seeking for a service.

The services onboarded to EOSC-Exchange are monitored via the ARGO Monitoring service.

The services are described and grouped (topology of the EOSC-Exchange) in the EOSC Portal, and the EOSC-Exchange Monitoring performs generic checks to verify the availability and reliability of each endpoint listed. The topology is currently sourced from the Marketplace component. It will be retrieved from the Service Registry as soon as the API is available. The services are currently monitored using only generic tests.

In parallel, the EOSC Exchange Monitoring Service currently integrates data from external monitoring sources (such as those operated by INFRAEOSC-07 projects like DICE and soon EGI-ACE). In one of the next releases, we expect that EOSC Exchange monitoring would be capable of performing more dedicated metrics via the Semi-automated Self Registration of Metrics/ Probes workflow that will be offered to the providers.

5.6 EOSC Messaging

The EOSC Messaging service is a real-time messaging service that allows the user to send and receive messages between independent applications. It is implemented as a publish/subscribe service. Instead of focusing on a

single messaging service, specifications for handling the logic of publishing/subscribing to the broker network the service focuses on creating nodes of publishers and subscribers as a service. In the publish/subscribe paradigm, publishers are users/systems that can send messages to named channels called topics. Subscribers are users/systems that create subscriptions to specific topics and receive messages.

Messaging components. The main components of the service are:

a) HaProxy: is the service load balancer.
b) Kafka: is a distributed publish-subscribe messaging system that is designed to be fast, scalable, and durable.
c) Zookeeper: is a centralised service for maintaining configuration information, naming, providing distributed synchronisation.
d) HTTP API: The messaging api.
e) Metadata store cluster: a distributed metadata store based on mongodB.

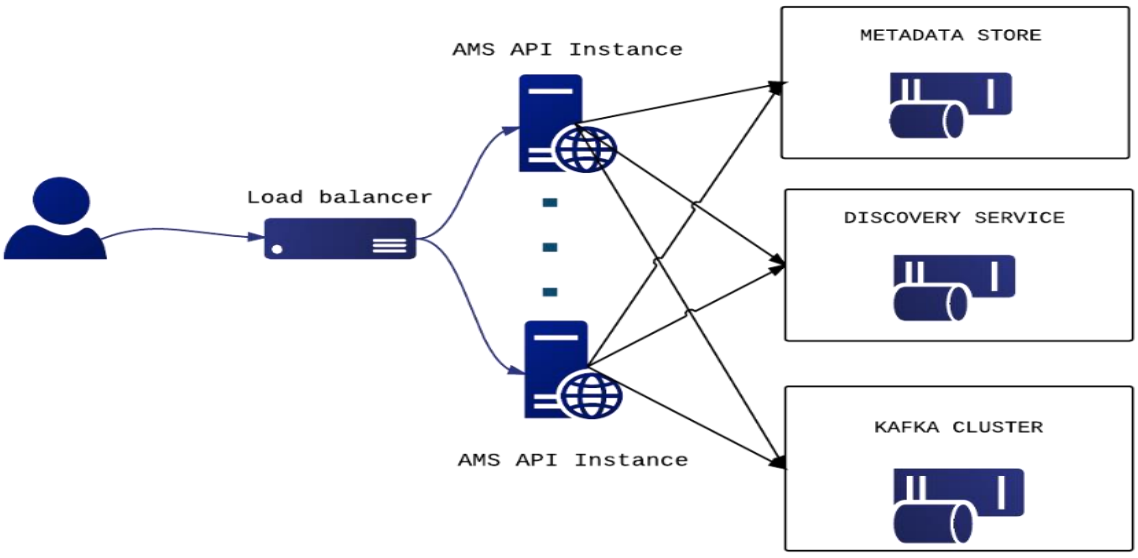

Figure 5.6: ARGO Messaging Service High-Level Architecture

As shown in Figure 5.6, a default deployment of messaging service comprises a haproxy server, which acts as a load balancer for the 3 AMS servers running in the backend. Table 5-12 shows the technical features of the EOSC Messaging Service.

Table 5-12:EOSC Messaging Service technical features

Component Name	EOSC Messaging Service
Short description	The Messaging service (AMS) is a publish/subscribe service, which implements the Google PubSub protocol. Instead of focusing on a single messaging API specification for handling the logic of publishing/subscribing to the broker network the API focuses on creating nodes of Publishers and Subscribers as a Service. It provides an HTTP API that enables Users/Systems to implement message-oriented service using the publish/subscribe model over plain HTTP.
Main features	**Features** • **Ease of use**: It supports an HTTP API and a python library to easily integrate with the AMS. • **Push delivery**: AMS instantly pushes asynchronous event notifications when messages are published to the message topic. Subscribers are notified when a message is available. • **Replay messages**: replay messages that have been acknowledged by searching for a timestamp. • **Schema support**: on-demand mechanism that enables a) the definition of the expected payload schema, b) the definition of the expected set of attributes and

	values and c) the validation for each message if the requirements are met and immediately notify the client. • **Replicate messages on multiple topics**: republisher script that consumes and publishes messages for specific topics (ex. SITES). **Architectural aspect** • **Durability**: provide very high durability, and at-least-once delivery, by storing copies of the same message on multiple servers. • **Scalability**: It can handle increases in load without noticeable degradation of latency or availability. • **Latency**: A high-performance service that can serve more than 1 billion messages per year. • **Availability**: it deals with different types of issues, gracefully failing over in a way that is unnoticeable to end-users. Failures can occur in hardware, in software, and due to load. In addition, the Messaging Service has some components like: • **Argo-ams-library**: a simple library to interact with the ARGO Messaging Service. • **Argo-AuthN**: Argo-authn is a new authentication service. This service provides the ability to different services to use alternative authentication mechanisms without having to store additional user info or implement new functionalities. The AUTH service holds various information about a service's users, hosts, API s, etc. and leverages them to provide its functionality. • **AMS metrics**: Metrics about the service and the usage.
URLs	https://msg.argo.grnet.gr http://argoeu.github.io/argo-messaging/
Documentation	Presentation: ARGO-Messaging-Service-GRNET Documentation: http://argoeu.github.io/argo-messaging/ Swagger API: https://api-doc.argo.grnet.gr/argo-messaging/

API types and use-cases	• /projects/{PROJECT}: handle projects
	• Manage users
	  ○ User registrations - handle user registrations
	  ○ Users - handle user accounts and tokens
	• **Topics**: topics are resources that can hold messages. Publishers (users/systems) can create topics on demand and name them (usually with names that make sense and express the class of messages delivered in the topic). A topic name must be scoped to a project.
	  ○ Ideas as a Publisher
	  ○ Manage Topics
	  ○ [PUT] Create new topic
	  ○ [DELETE] Delete topic
	  ○ [GET] Get a topic
	  ○ [GET] List Topics
	  ○ [POST] Publish message/s to a topic
	  ○ [POST] Modify ACL of a given topic
	  ○ [GET] List ACL of a given topic
	  ○ [GET] Topic Metrics
	• **Subscriptions**: A named resource representing the stream of messages from a single, specific topic, to be delivered to the subscribing application. A subscription name must be scoped to a project.
	  ○ Manage subscriptions
	  ○ [PUT] Create subscriptions
	  ○ [GET]: List all subscriptions under a specific topic
	  ○ [GET]: List all subscriptions
	  ○ [GET]: Get a subscription's list of authorised users

	○ [POST]: Modify ACL of a given subscription ○ [DELETE]: Delete subscriptions ○ [POST]: Modify ACK deadline ○ [POST]: Pull messages from a subscription (consume) ○ [POST]: Sending an ACK ○ [POST]: Modify offsets ○ [GET]: Get offset by timestamp ○ [GET]: Subscription metrics ○ [GET]: Get offsets ○ Push-enabled subscriptions ○ [POST]Request to create push-enabled subscription ○ [POST]: Manage subscriptions - verify ownership of a push endpoint ○ [POST]: Modify push configuration • **Schemas:** Schemas is a resource that works with topics under a specific project by validating the published messages. ○ [GET]: Retrieve a schema ○ [GET]: Retrieve all available schemas ○ [POST]: Create new schema ○ [PUT]: Update schema ○ [DELETE]: Delete schema ○ [PUT]: Validate message • **Metrics** ○ AMS metrics: AMS operational metrics ○ Project metrics: related metrics for the specific project: e.g. the number of topics, subscriptions) ○ Topic metrics: related metrics for the specific topic: e.g. for the number of published messages **Use Cases** **Accounting**: Use of the messaging-as-a-transport layer for collecting accounting data from the sites. The accounting information is gathered from different collectors into a central accounting repository, where it is processed to generate statistical summaries that are available through the Accounting Portal. The software used for transferring accounting records (SSM) is using the Messaging System. The SSM publishes messages to predefined topics. It uses the AMS-library to connect to AMS and collect the accounting data.
EOSC Future components expected to interact with this component	• **EOSC Accounting** • **EOSC Monitoring**

5.7 EOSC Interoperability Framework Guidelines Registry

The EOSC Interoperability Framework Guidelines Registry is a repository service containing all Interoperability Guideline entries used for compatibility and composability between EOSC onboarded Resources and for compatibility to EOSC Core Components Guidelines. Implementation allows onboarding on Interoperability Guidelines and linking of Guidelines to Resources, through Providers Portal UI. There's also an API to allow CRUD operations on IF DB records and manage relations between IF Guideline entries and EOSC Service Catalogue entries.

This registry is tightly integrated to the EOSC Service Registry and EOSC Providers Portal in terms of procedures because guidelines onboarding is essentially identical with the resource onboarding procedures, although the data model/profile for the interoperability guidelines is of totally different structure and scope.

Providers seeking to onboard interoperability guidelines should be already onboarded providers in the EOSC Service Catalogue. After initial onboarding of the guideline, an evaluation procedure for the guideline takes place and if the guideline is approved, it's available for resources to declare compliance to it.

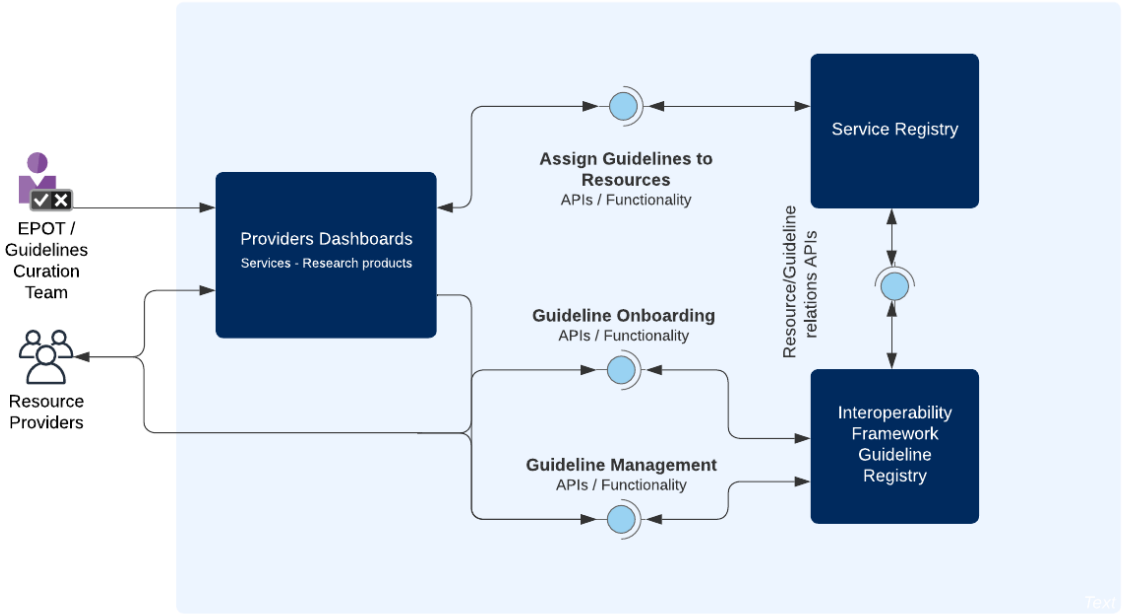

Figure 5.7: Interoperability *Framework Guidelines Registry Messaging Service High-Level Architecture*

EOSC Interoperability Framework Guidelines Registry main features are (grouped by use-case users):

For providers:

- Onboarding of Guidelines that can be used by others to interoperate with the Provider's resources;
- Management of already onboarded Interoperability Guidelines;
- Link a Resource with an onboarded Interoperability Guideline using a UI integration in the Providers' Portal;
- REST API supporting all operations above;
- Support the ability to define Configuration Templates for creating "configurations" of Interoperability Guidelines for Resources (to be released).

For EPOT team members

- Initial onboarding approval/rejection of Interoperability Guidelines;
- Auditing and curation of Interoperability Guidelines entries.

For other EOSC Core Components:

- Vice: Provide listings of Interoperability Guidelines that are adhered to by a Resource;
- Versa: Provide listings of Resources that have stated that are compatible to a specific Interoperability Guideline.

Table 5-13 shows the technical features of the EOSC Interoperability Framework Guidelines Registry.

Table 5-13: EOSC Interoperability Guidelines Registry technical features

Component Name	**EOSC Interoperability Framework Guidelines Registry**
Short description	The Service Registry is the storage component offering the necessary programmatic interfaces for the addition, modification, and access to information regarding providers, resources and user activity collected in EOSC portal. It contains: • Information for resources and providers entered by providers during the onboarding process based on the EOSC profiles schema. • Information collected by the interaction of providers with the portal and specifically with the Providers Component for the management and customisation of their offering. Such information includes updates and changes applied to a resource offered by a provider.

	• Information collected by the interaction of the end users with the Portal and specifically with the End User Component, including statistics for the end user visitation, orders, personalisation features such as user ratings, favourites, user profiles. • Registry administration, which offers functionality to validate/approve/curate the contents of the EOSC Service Registry. More specifically, it offers functionality for the EOSC Portal Onboarding team to manage incoming requests for the validation of a provider's or a resource registration, as well as perform a series of quality controls in the contents of the registry. • REST APIs (for enabling synchronous communication). All registry content is available in the form of open REST APIs in https://api.eosc-portal.eu/openapi for both XML and JSON representation schemas. In general, all get methods are open, whereas POST and PUT require an API key. • Apache Active MQ[21] (for enabling asynchronous communication). An SSL JMS messaging service enables the subscription of clients to registry-generating events, regarding CRUD operations on providers and resources. • ResourceSync[22] (for enabling harvesting). The registry offers a resource sync API, which can be harvested by external systems- harvesters.
Main features	• EOSC Interoperability Framework Guidelines Onboarding and Management • Providers can link Resources to Interoperability Guidelines to facilitate compatibility and/or composability with other Resources.
URLs	• EOSC Portal API: https://providers.eosc-portal.eu/openapi • EOSC API Sandbox https://sandbox.providers.eosc-portal.eu/openapi • EOSC API Beta https://beta.providers.eosc-portal.eu/openapi • EOSC API base URL: https://api.eosc-portal.eu/
Documentation	• UI Documentation: https://eosc-portal.eu/providers-documentation/eosc-provider-portal-basic-guide • API: https://providers.eosc-portal.eu/developers • GitHub: https://github.com/madgeek-arc/resource-catalogue
API types and use-cases	**EOSC API base URL:** https://api.eosc-portal.eu/{method} API Calls and examples: **Interoperability Guidelines API: Operations for Interoperability Guidelines** • **post /interoperabilityRecord: Creates a new Interoperability Record.** • **put /interoperabilityRecord: Updates the InteroperabilityRecord with the given id.** • **get /interoperabilityRecord/all: Get all Interoperability Records** • **get /interoperabilityRecord/bundle/all: Get all Interoperability Record Bundles** • **get /interoperabilityRecord/relatedResources/{id}: Returns the Related Resources of a specific Interoperability Record given its id.** • **post /interoperabilityRecord/validate: Validates the Interoperability Record without actually changing the repository.** • **get /interoperabilityRecord/{id}: Returns the Interoperability Record with the given id.** **Interoperability Guidelines and Resources API: Operations for Resources linked to Interoperability Guidelines** • **post /resourceInteroperabilityRecord: Creates a new ResourceInteroperabilityRecord.** • **put /resourceInteroperabilityRecord: Updates the ResourceInteroperabilityRecord with the given id.** • **get /resourceInteroperabilityRecord/all: Filter a list of ResourceInteroperabilityRecords based on a set of filters or get a list of all**

[21] http://activemq.apache.org/
[22] http://www.openarchives.org/rs/toc

	ResourceInteroperabilityRecords in the Catalogue. • **get /resourceInteroperabilityRecord/byResource/{resourceId}: Returns the ResourceInteroperabilityRecord of the given Resource of the given Catalogue.** • **get /resourceInteroperabilityRecord/{id}: Returns the ResourceInteroperabilityRecord with the given id.**
EOSC Future components expected to interact with this component	• EOSC Marketplace: consumes registry (interoperability guidelines) entries. • EOSC Service Registry: consumes and populates registry (interoperability guidelines) entries. • Provider Portal: guideline onboarding, resource management, and EPOT operations over the service registry. • Marketplace: offering monitoring data via API. • EOSC Exchange systems: CRUD operations via API for updating \ syncing the registry.

6 Conclusion and Next Steps

This document presented the EOSC Core Back-Office final functional and technical specifications.

The functional architecture presented here was derived from requirements collected during the project lifetime and from the high-level technical roadmap. The identified Enabling Functional Components in the functional architecture were mapped to a series of background components developed in past EOSC projects (EOSC Enhance, EOSC-hub, OpenAIRE Advance). WP4 adopted, extended, and integrated all these components to release the EOSC Core Back-Office that is now in production.

The technical architecture was designed to connect and integrate all these background components into a homogeneous and coherent platform. Technical roadmaps were defined for each component. These roadmaps have driven the WP4 development activities.

The EOSC Core Back-Office architecture has been integrated with the overall EOSC Platform architecture developed by the project. All its components are now operational and, all together, are able to support the majority of the use cases identified during the project lifetime and by the project technical roadmap.

This final specification should be considered as a fundamental document to perform the handover towards the initiatives that will deliver and enhance the EOSC Platform after the end of the EOSC Future project (e.g. the EOSC Procurement and the project that will be awarded in the call HORIZON-INFRA-2023-EOSC-01-04).

7 References

[1] EOSC-hub D10.5 - Requirements and gap analysis report [online] Available at: < https://www.eosc-hub.eu/deliverable/d105-requirements-and-gap-analysis-report>
[2] EOSC-hub D10.6 - Requirements and gap analysis report v2 [online] Available at: < https://www.eosc-hub.eu/deliverable/d106-requirements-and-gap-analysis-report-v2>
[3] OpenAIRE-Connect wiki [online] Available at https://support.d4science.org/projects/openaire-connect-wiki/wiki>
[4] EOSC Enhance D1.5 - EOSC Portal requirements (release 1) [online] Available at: https://eosc-portal.eu/eosc-enhance-deliverables>
[5] EOSC Enhance D1.6 - EOSC Portal requirements (release 2) [online] Available at: https://eosc-portal.eu/eosc-enhance-deliverables>