

# D13.2 First report on EOSC Execution Framework Architecture and capabilities

30/09/2025

## Abstract

This document provides a description of the first release of the EOSC Execution Framework, which includes the EOSC Interoperability Framework (IF) registry and the EOSC Deployment Service. These components are integrated with the User Space to facilitate the composability and dynamic deployment of resources onboarded and accessible from the EOSC Resource Catalogue.



Funded by  
the European Union

EOSC Beyond receives funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101131875

Disclaimer: Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

## Document Description

D13.2 First report on EOSC Execution Framework Architecture and capabilities			
<b>WP13</b>			
<b>Due date</b>	30/09/2025	<b>Actual delivery date:</b>	30/09/2025
<b>Nature of document</b>	Report	<b>Dissemination level</b>	Public
<b>Document Status</b>	Under EC Review	<b>Version</b>	1.0
<b>Lead Partner</b>	OpenAIRE		
<b>Authors</b>	Alessia Bardi (CNR, OpenAIRE), Carlos Brandt (EGI), Miguel Caballer (UPV) Thanassis Mantes (ARC), Gérman Moltó (UPV)		
<b>Reviewers</b>	Dusan Vudragovic (IPB), Anastas Mishev (UKIM)		
<b>Approved by</b>	TCB		
<b>Document link</b>	<a href="https://documents.egi.eu/document/4128">https://documents.egi.eu/document/4128</a>		
<b>DOI</b>	<a href="https://zenodo.org/records/17234674">https://zenodo.org/records/17234674</a>		
<b>Keyword</b>	EOSC Execution Framework, EOSC Interoperability Framework, EOSC Deployment Service, Composability-based applications		

## Revision History

Issue	Date	Description	Author/Reviewer
V 0.1	01/07/2025	Draft version	Alessia Bardi (CNR, OpenAIRE)
V 0.2	14/07/2025	Contributions from Task leaders	Miguel Caballer (UPV), Gérman Moltó (UPV), Thanassis Mantes (ARC), Carlos Brandt (EGI)
V 0.3	03/09/2025	Status wrt the technical roadmap (D10.2)	Alessia Bardi (CNR, OpenAIRE)
V 0.4	15/09/2025	Draft ready for review	Alessia Bardi (CNR, OpenAIRE)
V 0.5	23/09/2025	Addressed reviewers' comments	Alessia Bardi (CNR, OpenAIRE), Thanassis Mantes (ARC)
<b>V 1.0</b>	<b>30/09/2025</b>	<b>Final</b>	

Copyright and licence info

This material by Parties of the EOSC Beyond Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

## Table of content

Document Description	2
Revision History	2
Table of content	3
Figures	4
Tables	4
Executive Summary	5
1. Introduction	6
1.1. Purpose of the document	6
1.2. Scope of the document	6
1.3. Structure of the document	6
2. Architecture of the EOSC Execution Framework	8
2.1. Overview	8
2.2. The EOSC Interoperability Framework Registry	12
2.3. EOSC Deployment Service	14
2.4. Composability-based applications	17
2.4.1. Scholarly Communication Metadata Aggregator for EOSC Nodes	18
2.4.2. Data Transfer	19
2.4.3. Data Analysis Environment	21
3. Capabilities of the EOSC Execution Framework	22
3.1. Capabilities of the EOSC Interoperability Registry	22
3.2. Capabilities of the EOSC Deployment Service	23
4. Integration with the User Space	25
5. Conclusions and next steps	29
Acronyms	30

## Figures

Figure 1 - Execution Framework architecture diagram	8
Figure 2 - EOSC-IF upgraded data model	13
Figure 3 - Providers Dashboard, Service Registry and Interoperability Registry architecture diagram	14
Figure 4 - Deployment Service architecture diagram	15
Figure 5 - EOSC Profile for a Deployable Service	16
Figure 6 - Architecture overview of the Scholarly Communication Metadata Aggregator for EOSC Nodes	18
Figure 7 - Data Transfer application workflow: infrastructure with DOI-based dataset in it	19
Figure 8 - EOSC Data Transfer API integrated in EOSC EXPLORE	20
Figure 9 - Data analysis environment set-up workflow	21
Figure 10 - Web-based dashboard of the Infrastructure Manager with the list of available TOSCA templates	24
Figure 11 - Menu of the Provider Dashboard with Deployable Services and part of the form to create and publish a Deployable Service	25
Figure 12 - Dedicated list view for Deployable Services in the Discovery Hub	26
Figure 13 - Dedicated view with details of a Deployable Service	26
Figure 14 - Forms of the Front Office to request the deployment of a Deployable Service	27
Figure 15 - User projects and associated Deployable Service requests	28
Figure 16 - Detailed view of a deployment request	28

## Tables

Table 1- Use cases scenarios and Epics: Table 3-1 of D10.2 annotated with status at M18 (September 2025)	12
Table 2: Acronyms	30

## Executive Summary

This report details the architecture and capabilities of the first release of the EOSC Execution Framework (EF), a pivotal set of components developed within the EOSC Beyond project to enhance the dynamic deployment and composability of services across the EOSC ecosystem.

The first release of the EF aims to streamline how services are offered, discovered, deployed, and integrated, particularly focusing on the dynamic deployment aspect. The EF is fundamentally composed of two enhanced core components: an enriched EOSC Interoperability Framework (IF) Registry and an extended EOSC Deployment Service (DS), which can be seen as an evolution and more generalised version of the EU Node AWM implementing the Node Core Capability "Application Workflow Management " specified in the [EOSC Federation Handbook](#) (section 4.2.1).

Key advancements and capabilities delivered by this first release include:

- Evolution of the IF Registry from a repository of human-readable guidelines to a system supporting machine-composability of resources. It now enables the definition of Configuration Templates for Interoperability Guidelines (IGs), providing structured profiles for services to expose actual API endpoints.
- Extension of the Service Registry to support Configuration Instances and Deployable Services.
- Extension of the Deployment Service to automate the registration of fully qualified and secure domain names for deployed services, and offer automated data staging capabilities to fetch data from multiple open repositories to the dynamically provisioned virtual infrastructure.
- Integration with EOSC User Space: The EF is seamlessly integrated with the EOSC User Space. This integration introduces functionalities for end-users to search, select, and request the deployment of Deployable Services via a user-friendly three-step Offer Ordering Form. Providers can now create and publish their own Deployable Services. Also, Interoperability Guidelines can now be associated with (manually created) Configuration Templates, allowing compliant services to populate Configuration Instances for those templates.
- Three exemplary applications demonstrate and leverage the EF's enhanced capabilities: the Scholarly Communication Metadata Aggregator for EOSC Nodes, the Data Transfer application, and the Setting up Data Analysis Environment application.

Looking ahead, future developments in the Work Package 14 will focus on completing the implementation and expanding capabilities. This includes developing a user-friendly editor for Configuration Templates (up until now, creation of a Configuration Template and association with a Guideline is done manually by the IF Registry admins), extending the EOSC Deployment Service to automatically request TLS certificates and integrate the Data Transfer Service.

## 1. Introduction

The EOSC Execution Framework (EF) is designed to deliver an extensible set of composability-based applications. It comprises two main components: (i) the EOSC Interoperability Framework (IF) Registry, and (ii) the EOSC Deployment Service (DS). The IF Registry associates configurations to Interoperability Guidelines (IGs) to provide machine-readable instructions for instantiating services. The Deployment Service, in turn, is responsible for deploying on-demand services in hybrid Cloud infrastructures. It can be seen as an evolution and more generalised version of the EU Node AWM implementing the Node Core Capability "Application Workflow Management " specified in the [EOSC Federation Handbook](#) (section 4.2.1). Both the IF Registry and the Deployment Service are connected to the Resource Catalogue to describe how services can be accessed and/or deployed by users.

The EF's overarching goal is to facilitate the dynamic deployment and composability of services within the EOSC ecosystem. It enables service providers to offer deployable instances of their onboarded services, while users can configure, deploy, and integrate these instances into custom workflows and applications.

### 1.1. Purpose of the document

This document reports about the architecture and capabilities of the [EOSC EF](#) designed in Task 10.2 and released at M17 (August 2025) of the EOSC Beyond project.

### 1.2. Scope of the document

This document describes the architecture and capabilities of the EOSC EF. The EF is specifically designed to facilitate the dynamic deployment and composability of services within the EOSC ecosystem.

This report focuses on the components implemented during the first phase of the EOSC-Beyond project, particularly addressing the aspect of dynamic deployment of services. The aspect of composability will be addressed by Work Package 14 (WP14) in the second phase of the project.

Drawing from the design and technical roadmap defined in [D10.2](#), this document details the available architectural components, their capabilities, and their coverage of various use case scenarios and epics. Software repository details and more information about the integration of the EF with the User Space are available in [D13.1](#).

### 1.3. Structure of the document

- [Section 2](#): Provides an overview of the architecture of the EOSC Execution Framework.

## D13.2 First report on EOSC Execution Framework Architecture and capabilities

- **Section 3**: Details the capabilities of the EOSC Execution Framework.
- **Section 4**: Describes the integration with the User Space.
- **Section 5**: Presents conclusions and outlines the next steps.

## 2. Architecture of the EOSC Execution Framework

### 2.1. Overview

The EOSC Execution Framework (EF) facilitates the dynamic deployment and composability of services within the EOSC ecosystem. [Figure 1](#) shows the architectural components of the EF and their interactions.

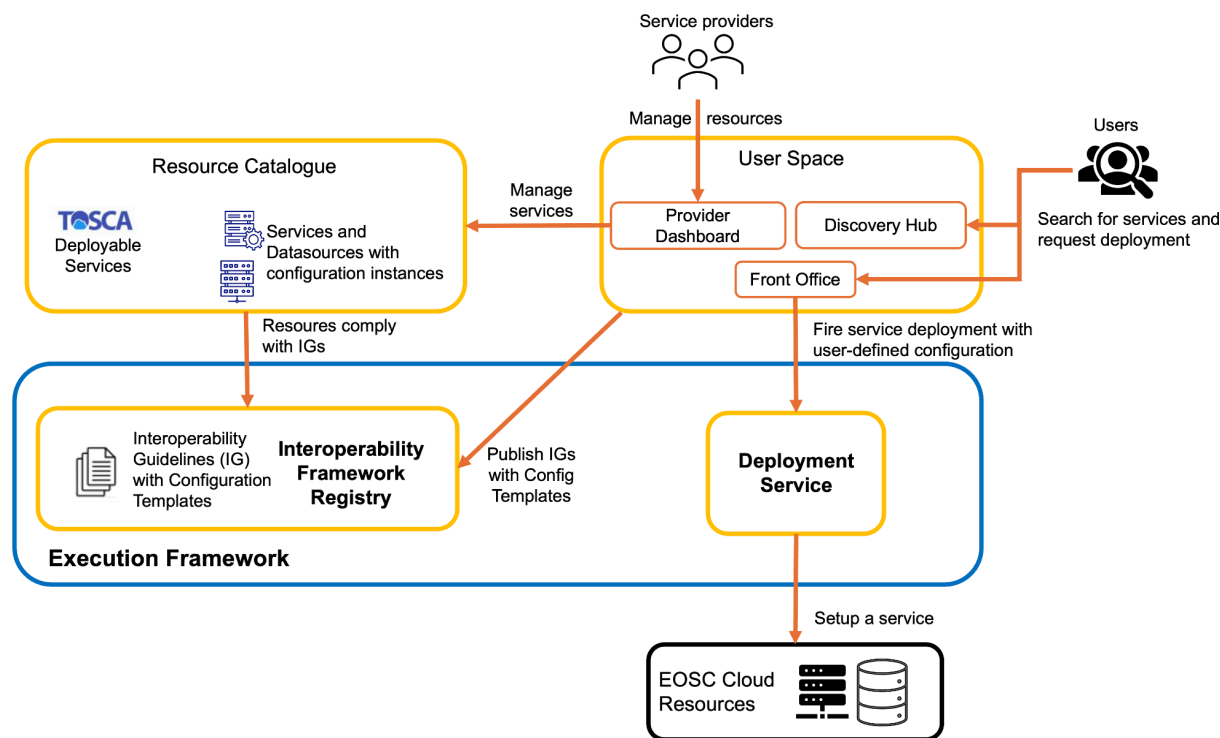


Figure 1 - Execution Framework architecture diagram

The main components of the EF are:

- The **EOSC Interoperability Framework (IF) registry**. A database designed to contain Interoperability Guideline profiles. This registry enables providers to align their services and products with specific guidelines, thereby indicating compatibility and composability features with other EOSC resources or core components. By recording these guidelines, the registry enhances the overall interoperability within the EOSC ecosystem. During M1-M18, the IF registry has been enriched with configuration templates associated with interoperability guidelines.
- The **EOSC Deployment Service (DS)**. Implemented through the Infrastructure Manager (IM), the DS is an open-source service to deploy complex applications on multiple Cloud back-ends (AWS, OpenStack, etc.). It performs virtualised infrastructure provision, automatic software deployment and lifecycle management of these applications. It supports the OASIS TOSCA (Topology and Orchestration Specification for Cloud Applications) standard for application architecture description. During M1-M18, a new guideline for TOSCA Deployable Services has

been created and registered in the IF registry, the DS has been extended to support the automated registration of DNS hostnames and to support automated data staging capabilities.

They interact and integrate with:

- The **Resource Catalogue**, which was extended to:
  - Manage entities of type Deployable Services, intended as TOSCA recipes that can be automatically deployed;
  - Manage Configuration Instances of services and data sources.
- The **User Space**, which was extended with
  - Functionalities for end-users to search, select (via the **Discovery Hub**), and request deployment of Deployable Services (via the **Front Office**);
  - Functionalities for providers to create and publish via the **Provider Dashboard**:
    - Services and data sources compliant with Interoperability Guidelines and, in case the guideline has a configuration template, the relative configuration instance;
    - Deployable Services.

The current architecture supports the workflow envisaged in deliverable [D10.2](#)

1. A service provider onboards (registers) a Deployable Service to the Service Catalogue (in turn to the EOSC Resource Catalogue);
2. The User Space makes the Deployable Service discoverable (in the Discovery Hub) and automatically creates an offer for it by fetching input values from the provided TOSCA recipe (Front Office);
3. An end user finds the Deployable Services in the Discovery Hub and places an order for the underlying offer via the Front Office, specifying dynamic parameters of the TOSCA template and optionally provide DOIs of input datasets that will be automatically downloaded in the deployed infrastructure;
4. The Front Office sends a request for deployment to the Deployment Service which will return the end point of the service environment.

In addition, the technical roadmap defined a set of requirements that the architecture of the EOSC Execution Framework should satisfy (Table 3-1 of [D10.2](#)). Details on the status are given in [Table 1](#), where we specify if a requirement is completely satisfied (DONE), partly satisfied (ONGOING), or to be addressed in the second phase of the EOSC Beyond project by Work Package 14 (PLANNED).

D13.2 First report on EOSC Execution Framework Architecture and capabilities

Requirement	User Story	Technical Implementation	EOSC Components	Epic Reference <sup>1</sup>	Status
<b>Deployable Service Onboarding</b>	A service provider wants to onboard a service and make it deployable for end users.	Providers must be able to register their services in the Resource Catalogue, linking them to Interoperability Guidelines and defining TOSCA-based configuration templates.	EOSC Deployment Service, Service Catalogue, Provider Dashboard	<a href="#">EBTR-270</a>	<b>DONE</b>
<b>Service Deployment via Front Office</b>	An end user needs to deploy an onboarded service dynamically.	The Front Office must allow users to trigger service deployments, automatically retrieving and displaying execution locations once completed.	EOSC Deployment Service, Front Office, Service Catalogue	<a href="#">EBTR-270</a>	<b>DONE</b>
<b>Composing Multiple Services</b>	A researcher wants to combine multiple services into a data-driven workflow.	The Execution Framework must enable the linking and integration of deployable services, ensuring compliance with Interoperability Guidelines.	EOSC Deployment Service, Front Office	<a href="#">EBTR-270</a>	<b>PLANNED</b>

<sup>1</sup> The progress of the tasks is tracked with the help of Jira tickets allocated to the EOSC Beyond project. This Jira environment is accessible only for EOSC Beyond Consortium members, thus the Epic reference in this table and text below referring to the Jira environment is not available for a wider audience.

Requirement	User Story	Technical Implementation	EOSC Components	Epic Reference <sup>1</sup>	Status
<b>Secure and Efficient Data Transfer</b>	A researcher needs to transfer large datasets between repositories.	The Data Transfer Service must provide secure, high-speed transfer capabilities, support AAI-based authentication, and ensure compatibility with multiple data formats.	EOSC Data Transfer Service, EOSC AAI	<a href="#">EBTR-181</a>	<b>ONGOING<sup>2</sup></b>
<b>Automated Data Movement</b>	A researcher wants to automate data transfers as part of a workflow.	Support for scheduled and API-driven transfers, with monitoring and error recovery features.	EOSC Data Transfer Service	<a href="#">EBTR-181</a>	<b>ONGOING<sup>2</sup></b>
<b>Aggregating Metadata from Multiple Sources</b>	A service provider needs to collect metadata from different repositories.	The Metadata Aggregator Service must support OAI-PMH and REST APIs, standardise metadata formats, and allow advanced search/filtering.	Scholarly Metadata Aggregator, Service Catalogue, IF Registry	<a href="#">EBTR-269</a>	<b>ONGOING<sup>3</sup></b>

<sup>2</sup> Completed functionality for epic [EBTR-181](#): (i) DT adapted to latest FTS update and to manage S3 stores; (ii) Integration of DT in EOSC EXPLORE. Work in progress: integration of DT with the Deployment Service

<sup>3</sup> The first version of the Scholarly Metadata Aggregator composability application released at M17 (August 2025) by WP13 queries the EOSC Service Catalogue API and invokes OAI-PMH end points to fetch metadata. The integration of the EOSC Data Transfer API and the possibility for the user to select the target destination is planned in WP14.

Requirement	User Story	Technical Implementation	EOSC Components	Epic Reference <sup>1</sup>	Status
<b>Cross-Disciplinary Metadata Discovery</b>	A researcher wants to search and retrieve scholarly metadata across disciplines.	Implementation of federated search capabilities, integration with persistent identifiers (DOIs, PIDs), and metadata export in multiple formats.	Scholarly Metadata Aggregator, Service Catalogue, IF Registry	<a href="#">EBTR-269</a>	<b>ONGOING<sup>3</sup></b>
<b>Interactive Data Analysis Environment</b>	Researchers need scalable, pre-configured computing environments.	Deployment of Jupyter, RStudio and other solutions, supporting secure data storage and high-performance computing.	EOSC Deployment Service, Service Catalogue, IF Registry, Front Office	<a href="#">EBTR-270</a>	<b>DONE</b>
<b>Collaborative Research and Reproducibility</b>	Research teams need shared workspaces for analysis and result versioning.	Version-controlled workspaces, shared data storage, and DOI/PID-linked outputs for better reproducibility and collaboration.	EOSC Deployment Service, Service Catalogue, IF Registry, Front Office	<a href="#">EBTR-270</a>	<b>ONGOING</b>

Table 1- Use cases scenarios and Epics: Table 3-1 of D10.2 annotated with status at M18 (September 2025)

## 2.2. The EOSC Interoperability Framework Registry

The EOSC IF Registry has now evolved from being a simple collection of human-readable guidelines (which are instructions that EOSC Providers must implement to enable the desired functionality and interoperability between their EOSC services and/or research products) to a collection of guidelines that support machine-composability of resources. These guidelines may reference various existing standards, protocols and processes, in a manner that allows Providers to understand how to design and configure services to be part of the EOSC ecosystem. Although the Interoperability Guidelines are helpful to humans, they do not allow for automated deployment or metadata consistency checks.

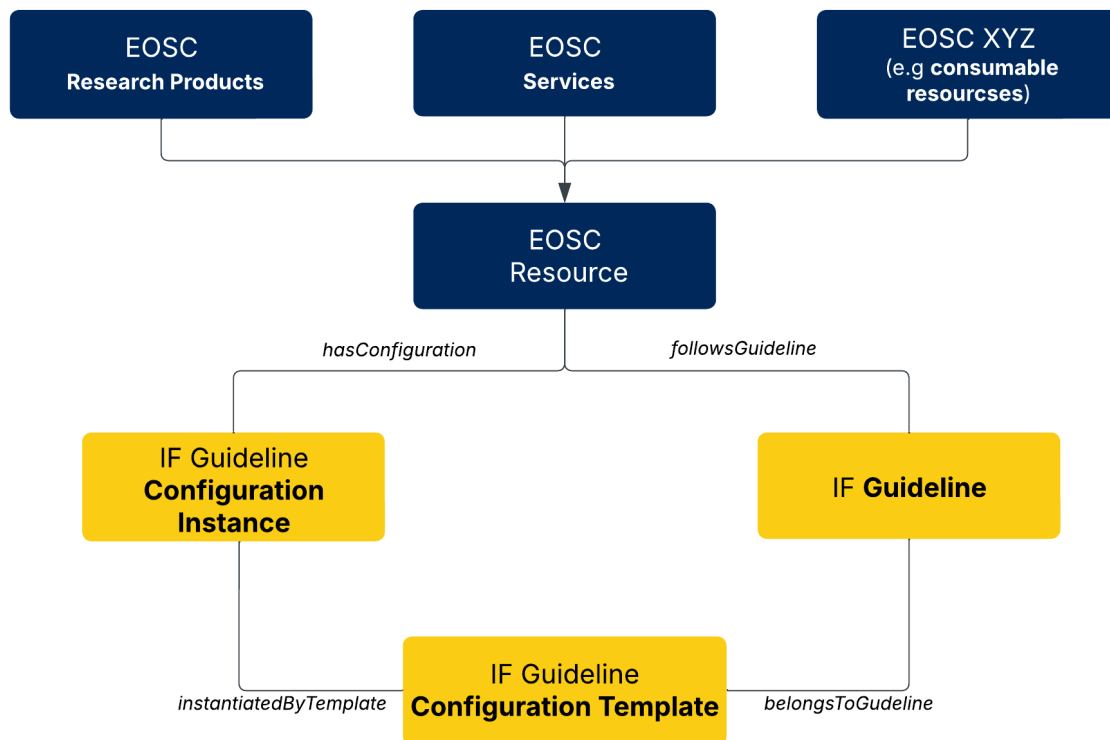


Figure 2 - EOSC-IF upgraded data model

For this, the notion of *Configuration Templates* for Guidelines is now introduced as shown in [Figure 2](#). Configuration Templates are structured metadata profiles that allow providers to detail the actual access parameters of their services according to the guidelines ([EBTR-117](#), [EBTR-118](#)). As such, they are defined together with a Guideline. Guideline providers can onboard Guidelines through the Providers Portal or through an upgrade of the existing API for onboarding and managing Guidelines (definition of Configuration Templates is for now a manual procedure done by Guideline curators).

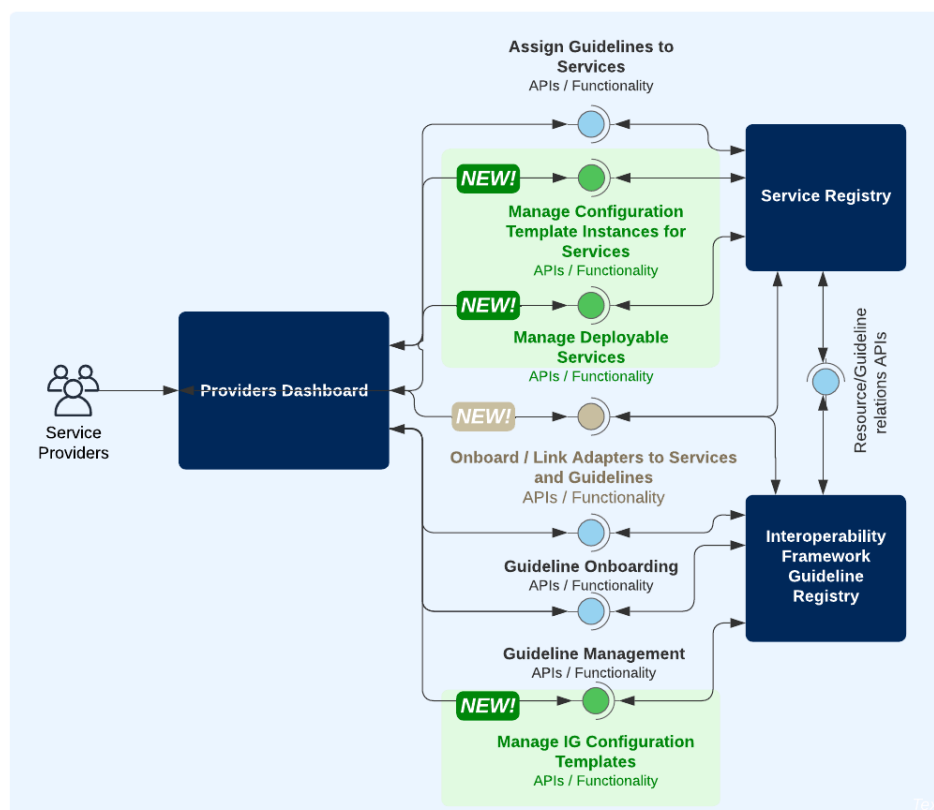


Figure 3 - Providers Dashboard, Service Registry and Interoperability Registry architecture diagram

Based on the above, the EOSC Service Catalogue is modified to accommodate *Service Configurations* (which are actually Configuration Template Instances for a specific Service related to an IF Guideline); The Providers Dashboard is now upgraded to support editing of Configuration Template instances for Services linked to Guidelines through a UI; moreover, these instances need to be persisted in the Service Registry and be part of the data that the Service Catalogue shares with other Core Components (particularly the Front Office) through messaging or through an API ([EBTR-119](#)). Note that the Providers Portal currently does not offer an editor for Configuration Templates to allow the definition of the template's schema: i.e., fields, types and sets of allowed values and the general structure of template metadata. A more structured and UI friendly editor will be implemented in the second half of the project.

Finally, the IF Registry is upgraded to offer the ability to access associated IF Guidelines and related software libraries of an onboarded service (through an API or through EOSC Messaging) to the Front Office UIs ([EBTR-120](#)).

The described new functionalities are highlighted in green in the architecture diagram in [Figure 3](#).

## 2.3. EOSC Deployment Service

The EOSC Deployment Service (DS), based on the [Infrastructure Manager](#), enables the support of automated provisioning and configuration of deployable services and virtualised

computing infrastructure through standard TOSCA templates and DevOps tools (e.g., Ansible) for automated configuration of computational resources.

The service has been extended to provide automated integration and composability of data sources, software tools and computing infrastructures. **Figure 4** shows the general architecture and the integration with the rest of the components of the EOSC Execution Framework and external services. The currently implemented functionalities are highlighted in green, with dashed lines the pending ones, and with black non-dashed lines, previously existing functionalities.

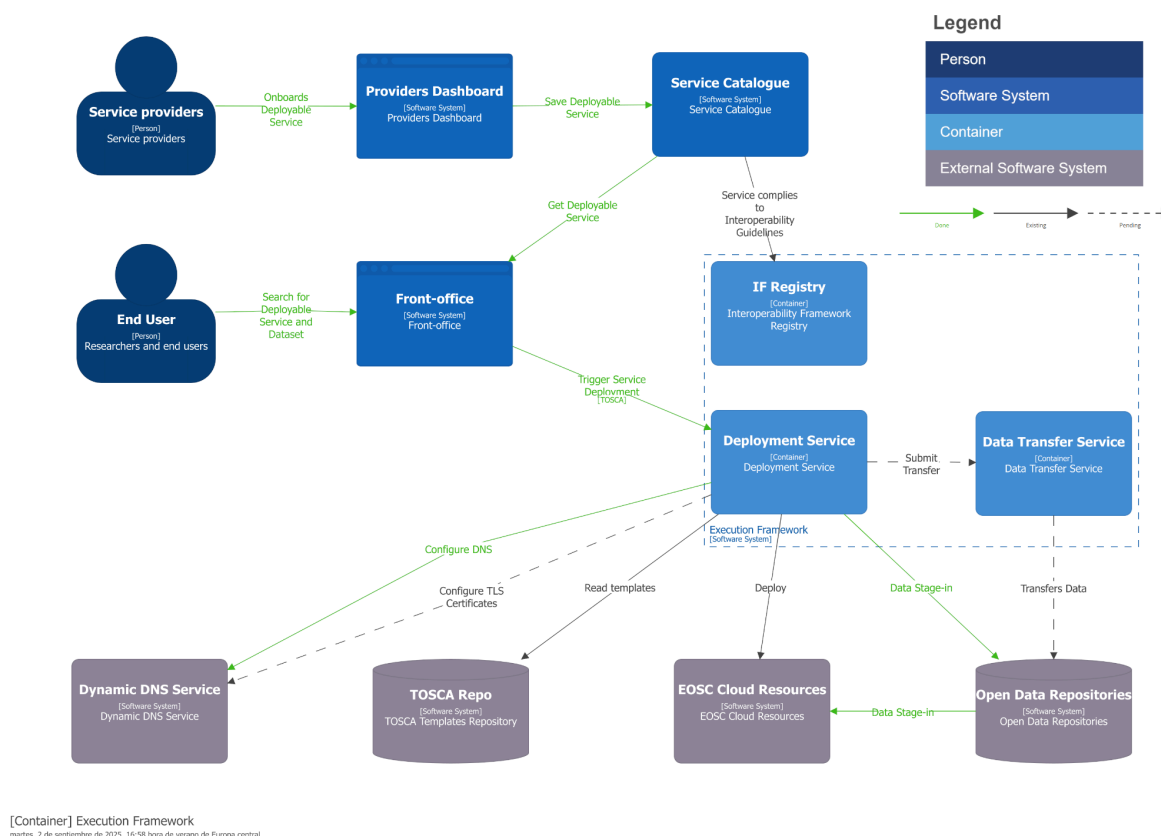


Figure 4 - Deployment Service architecture diagram

To onboard a deployable service, a dedicated template needs to be filled out following the TOSCA specification. To make this process easier, a simplified version has been defined and implemented by the DS that enables any user without TOSCA knowledge to define it.

**Figure 5** summarizes the EOSC Profile defined for a deployable service, which defines the metadata fields required to describe a service that can be automatically deployed via the EOSC Deployment Service. Note that the TOSCA template is referenced in the *url* field to point to an external repository (e.g. GitHub) containing the description of the customized virtualized infrastructure required to deploy the service. As an example, a curated set of TOSCA templates is available at: <https://github.com/grycap/tosca>.

## D13.2 First report on EOSC Execution Framework Architecture and capabilities

	Element name	Description	Type	Value	Multiplicity	Mandatory
Basic	id	unique ID (global)	string	auto	1	Yes
	name	Resource Full Name as assigned by the Provider.	string	<free-value>	1	Yes
	acronym	Acronym of the Resource	string	<free-value>	1	No
	url	The URL where the TOSCA template to deploy this service is stored.	URL	<free-value>	1	Yes
Classification	scientificDomains	The branch of science, scientific discipline that is related to the Resource.	tns:serviceProviderD...	cv	N	No
	tags	Keywords associated to the Resource to simplify search by relevant keywords.	string	<free-value>	1	No
Creator (1..N)	name	Creator name: - person name in the form Surname, Name - organization name (when the creator is an organization)	string	<free-value>	0..1	No
	email	Creator's email	email	<free-value>	0..1	No
	organisation - name					
	organisation - PID	Creator organisation and PID	string	<free-value>	1..N	No
Marketing	description	A high-level description in fairly non-technical terms of a) what the Resource does, functionality it provides and Resources it enables to access.	string	<free-value>	1	No
	tagline	Short catch-phrase for marketing and advertising purposes. It will be usually displayed close to the Resource name and should refer to the main value or purpose of the Resource.	string	<free-value>	1	No
	logo	Link to the logo/visual identity of the Resource. The logo will be visible at the Portal.	URL	<free-value>	1	No
Maturity	version	Version of the Resource.	string	<free-value>	1	Yes
	lastUpdate	Date of the latest update of the Resource.	date	<free-value>	1	No
	license	License (eg, MIT, Apache 2.0, GPL 3.0)	string	cv	1	No

Figure 5 - EOSC Profile for a Deployable Service

A comprehensive interoperability guideline for TOSCA-based deployable services is being finalised, which describes to service owners how to make their services “deployable”. The guideline describes the modifications made in all the services involved in the description and deployment of the deployable templates based on the TOSCA Simple Profile in YAML Version 1.0 OASIS standard. Also, the guideline describes how to onboard a new Deployable Service in the catalogue that can then be deployed, and how to add new deployable templates to extend the current set of existing ones.

A second key enhancement of the DS is supporting the deployment of datasets close to the compute services. To that aim, the TOSCA specification has been extended to describe the location of the datasets, adding the required TOSCA custom node types<sup>4</sup>. Also, the DS has been extended to enable automated data staging capabilities to fetch the data sources from multiple open repositories and make them available to the dynamically provisioned virtual infrastructure. This is achieved by integrating the Datahugger library and extending it with support for additional data sources (e.g. B2Share).

A second step, which is still under development, is to add the support to interact with the Data Transfer Service (DTS) for those cloud providers that host a storage system, which will enable the usage of the DTS to manage the third party data transfer from the open repositories to the cloud storage system. Thus making the data accessible from a location near the computational resources to be deployed. Note that the DTS only supports third-party transfers (e.g. from the data repository to an object storage system). It does not support transferring from the data repository to the file system of the virtual infrastructure deployed via the Deployment Service.

<sup>4</sup> [Custom Types](#) are a standard way in TOSCA to integrate new types in the TOSCA specification without requesting an official extension of the basic normative types.

Additionally, the DS has been extended to interact with the newly developed API of the Dynamic DNS Service to assign DNS entries during deployment. This automation minimises user intervention and provides a secure, ready-to-use infrastructure for data-driven applications. This task will require additional development in the Dynamic DNS Service and the DS to support TLS certificate generation.

We provide further technical details on the developments of the DS carried out until this first report and link the activities performed with the EOSC Beyond Technical Roadmap.

### **Extend TOSCA specification to support the definition of data sources [\[EBTR-130\]](#)**

Two new custom node types have been defined to specify datasets in TOSCA:

- `tosca.nodes.eosc.Dataset`: for direct fetching of data from repositories to VMs
- `tosca.nodes.eosc.dts.Dataset`: to specify the usage of the DTS service to manage the transfer from the repositories to some cloud storage system.

### **The DS must use automated data staging capabilities [\[EBTR-131\]](#)**

The DS has been extended to support fetching the first type of datasets into the deployed set of virtual machines using the [Datahugger](#) library. To demonstrate this functionality a proof of concept was made in collaboration with the NFDI pilot to deploy a Jupyter Notebook with a dataset stored at DESY open [repository](#).

### **The DS should be integrated with the Dynamic DNS service [\[EBTR-132\]](#)**

The Dynamic DNS service API has been extended, enabling the addition and deletion of hosts, listing available domains, and listing the hosts added by the user, using OpenID Connect tokens for authentication. The DS has been improved to use the new API calls to automatically assign DNS names to the deployed VMs.

## **2.4. Composability-based applications**

Three composability-based applications have been designed and implemented to demonstrate the use of the extended Execution Framework:

- Scholarly Communication Metadata Aggregator for EOSC Nodes: this application enables users to search and harvest metadata records from OAI-PMH compatible EOSC data sources, with email notifications and download links for collected metadata.
- Data Transfer: this application has been successfully integrated with the EOSC Deployment Service and EOSC EXPLORE, for convenient dataset selection transfer.
- Data Analysis Environments: this application combines customized virtual infrastructure deployment with dynamic data fetching, as demonstrated for the NFDI EOSC Pilot Node. It utilizes TOSCA templates to provision virtual machines, configure software (e.g., Jupyter), and automatically retrieve datasets via DOIs, coordinating various services to create remote data analysis environments.

Their architecture and interactions with the Execution Framework and other EOSC Core services are described in the sub-sections below.

### 2.4.1. Scholarly Communication Metadata Aggregator for EOSC Nodes

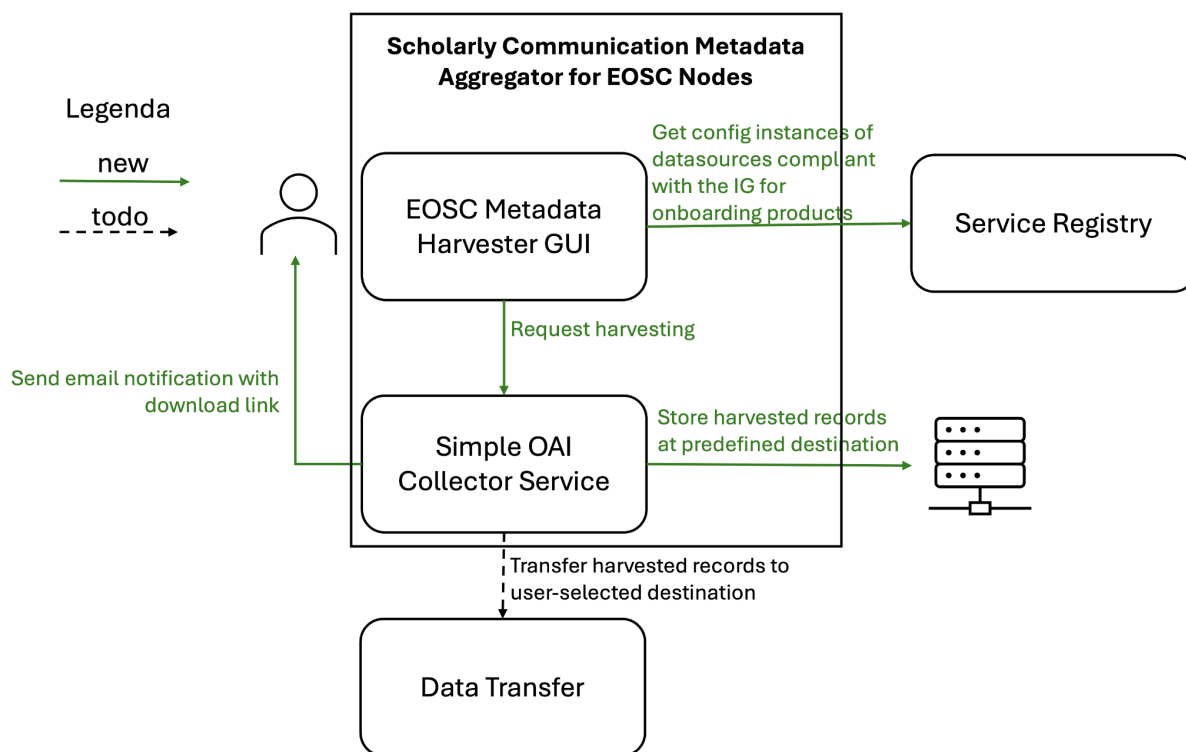


Figure 6 - Architecture overview of the Scholarly Communication Metadata Aggregator for EOSC Nodes

The Scholarly Communication Metadata Aggregator for EOSC Nodes demonstrates how the Configuration Instances introduced in the extended Execution Framework, can be used to automatise the task of harvesting metadata records from data sources compliant with the Interoperability Guidelines (IG) for onboarding research products.

The architecture in [Figure 6](#) shows the two main components of the application. The implemented functionalities are highlighted in green, with dashed lines the pending ones, to be implemented in the next phase of the EOSC-Beyond project. The EOSC Metadata Harvester GUI is the front-end of the application ([EBTR-344](#)). Users can search for data sources compliant with the IG for onboarding. The list of data sources is obtained with a query for Configuration Instances on the Service Registry. Upon request of the user, the GUI calls the API of the Simple OAI Collector Service ([EBTR-343](#)), the second component of the application, to launch the harvesting. Harvested records are stored at a predefined location. The user is notified about the completion of the process via email, with a message that includes a link from which the user can download the harvested metadata records.

## 2.4.2. Data Transfer

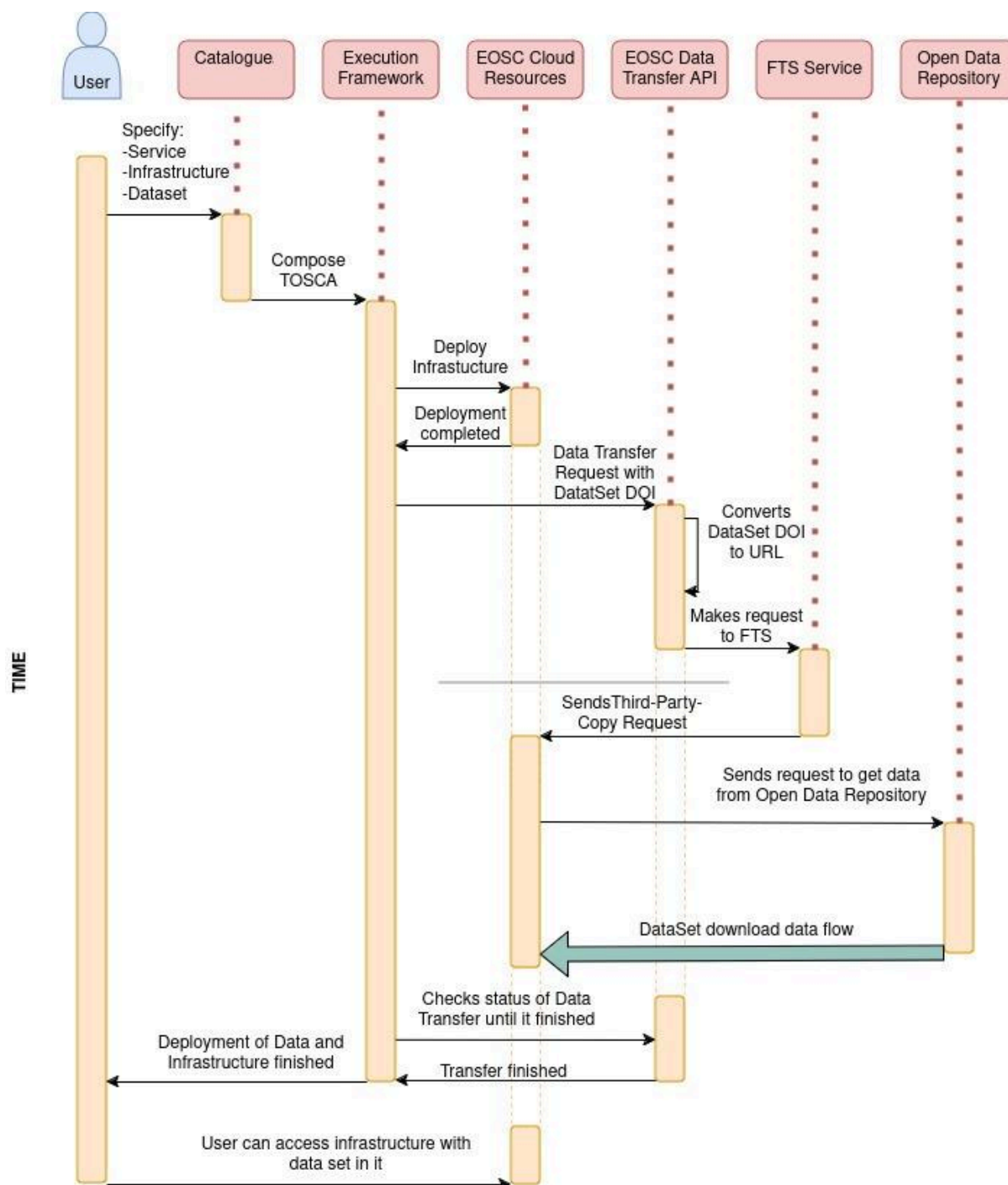
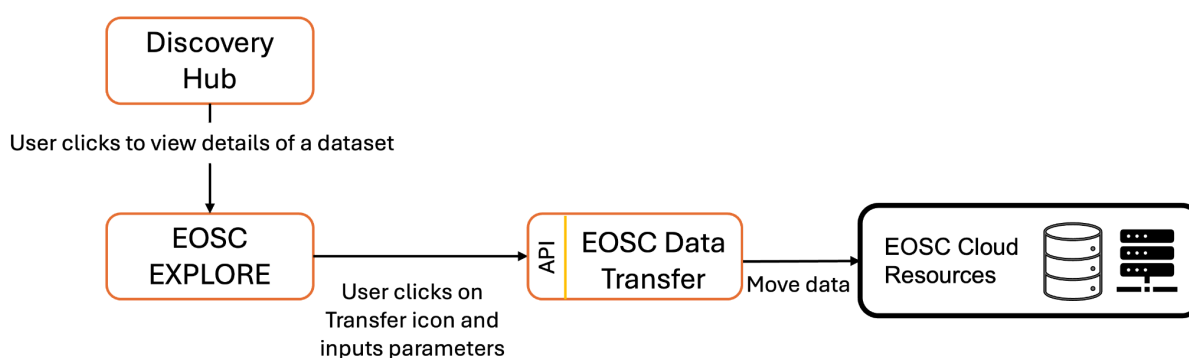


Figure 7 - Data Transfer application workflow: infrastructure with DOI-based dataset in it

The Data Transfer application is a more general and simpler case than the [Data Analysis Environment](#) described below ([Section 2.4.3](#)), a necessary one specially when handling legacy or non-interactive data analysis pipelines. This scenario focuses on the dynamic creation of an infrastructure (e.g., a virtual machine on a cloud provider) and loading with dataset(s) at the user's choice when ordering the new deployment.

**Figure 7** presents the workflow the Execution Framework goes through to provide the user the infrastructure with the dataset(s) requested in the first place through Discovery Hub/Front Office. When the user requests a new infrastructure, they specify the DOI(s) linked to a dataset in a supported platform (e.g., Zenodo). The Deployment Service will then customize the associated TOSCA template to include the given DOI(s) as well as the chosen cloud resources provider<sup>5</sup>. The deployment is carried out and data is transferred to the new infrastructure. There is a waiting time of some minutes, depending on the size of the dataset, when the user is free to do other things. When the deployment is ready it will appear as such to the user in their User Space.

The EOSC Data Transfer API is also integrated in the landing pages of datasets in the User Space (EOSC EXPLORE). **Figure 8** shows the interactions among the involved components and users.



*Figure 8 - EOSC Data Transfer API integrated in EOSC EXPLORE*

- Users search for datasets in the Discovery Hub and select one dataset with a DOI hosted by one supported platform (e.g. Zenodo).
- The landing page of the dataset, implemented by the EOSC EXPLORE component, features a “Transfer” icon that users can click.
- If the users are not already logged in, EOSC EXPLORE asks them to authenticate via the EOSC AAI.
- The “Transfer” icon opens a modal window where users must input the following information:
  - Confirm the DOI to use to fetch data (in case the dataset has multiple DOIs, e.g. different versions).
  - Destination storage type. Supported options are: S3 compatible object storage, S3 compatible object storage over HTTPS, StoRM, dCache.
  - Destination system (hostname and port).
  - Authentication info (when applicable).
  - Destination path.
  - Acceptance of the data protection policy.
- EOSC EXPLORE calls the EOSC Data Transfer API to request the transfer.
- The EOSC Data Transfer Service moves the data to the designated destination.

<sup>5</sup> As of this writing, we have fixed the provider to **IISAS**, project partner.

## 2.4.3. Data Analysis Environment

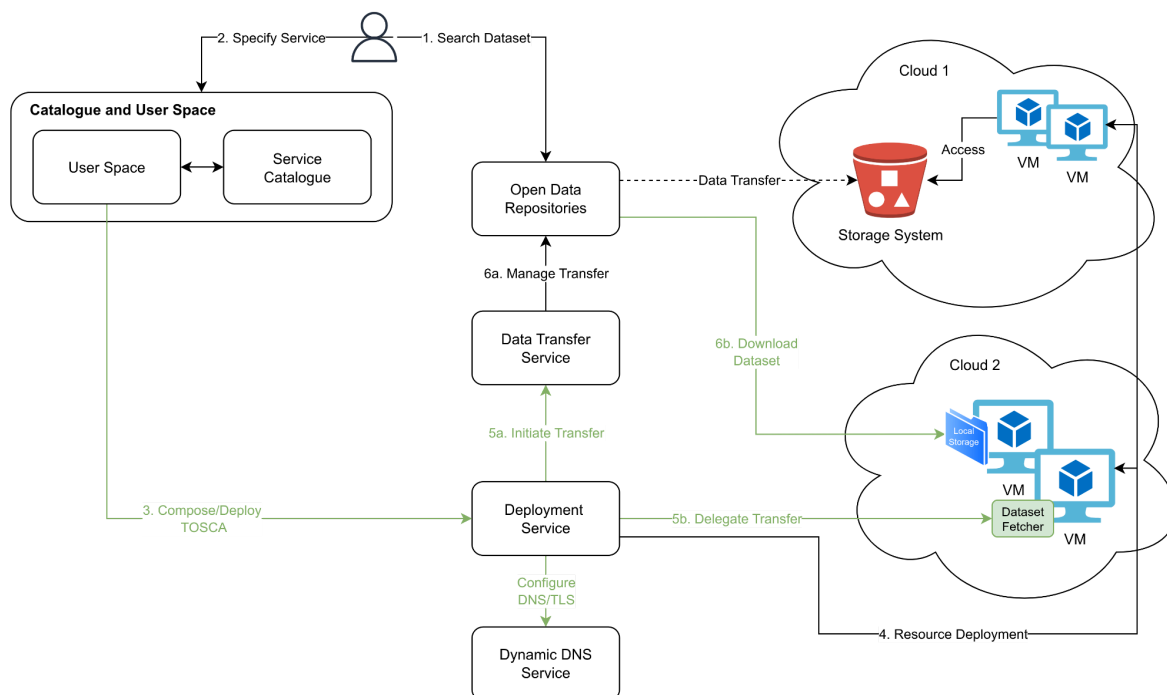


Figure 9 - Data analysis environment set-up workflow

Users can find research data, identify the service they may need for its analysis or processing, and deploy both services and data at the same site to enable data analysis.

**Figure 9** depicts the workflow to set up a data analysis environment, using the colour green to highlight new components or developments implemented (**EBTR-270**). The user finds a product of interest in the User Space (e.g., a dataset), which recommends the usage of a deployable service to perform data processing. This requires the DOI of the dataset to be staged into the storage allocated for the application. At this point, the User Space will compose the corresponding TOSCA template and contact the DS to perform the deployment of the cloud resources, configuration of the DNS name, configuration of the selected service and fetch the datasets into the deployed set of virtual machines. The user is provided with the endpoint of the application environment to perform the data analysis.

## 3. Capabilities of the EOSC Execution Framework

### 3.1. Capabilities of the EOSC Interoperability Registry

The EOSC Beyond Interoperability Framework Registry now adds a semantic layer on top of the EOSC Beyond Resource Catalogue that allows viewing (and discovery, browsing, etc.) EOSC resources in the EOSC Catalogue in terms of their compliance with IF guidelines. To this aim, the EOSC-IF Registry is now able to assign unique IDs and profile descriptions to IF guidelines for both the EOSC Core and the EOSC Exchange. IF guidelines also include the *Interoperability Framework Guideline Configuration Templates*.

*Configuration Templates*: “being compatible with a set of guidelines” is a relevant piece of information, allowing, for example, resource discovery by compatibility (e.g., all services compatible with OAI-PMH protocol); however, knowing the specific configuration of guidelines for a given service (e.g., URL of the OAI-PMH endpoint) is critical to enable orchestration at the service level. The EOSC-IF registry makes this now possible, so each EOSC IF guideline can provide a dedicated *configuration template* (set of properties required for compliance). Services onboarded in the EOSC Service Catalogue and declaring their conformance to EOSC-IF guidelines are able to provide the necessary info (*configuration instance*) as requested by the respective configuration template.

To complete the scenario, the EOSC Resource Catalogue data model is extended to express the compliance of a resource (service, product, or other types in the future) to an EOSC-IF guideline in the EOSC-IF registry. For example, EOSC service profiles express a semantic relationship with the EOSC-IF guidelines for Service onboarding (Core guidelines), which come with a given option (versions); e.g. service catalogues integrated with the EOSC Catalogue will (i) specify compatibility their compliance with the “EOSC-IF guidelines for service Catalogues” and (ii) provide a configuration instance that describes the “pull”/harvest APIs for the given service catalogue.

The EOSC Beyond Interoperability Framework Registry, as part of the EOSC Execution Framework, is now deployed in the EOSC Core Innovation Sandbox instances. As already said, at the beginning of the project the EOSC IF Registry was a simple collection of human-readable guidelines (which are instructions that EOSC Providers must implement to enable the desired functionality and interoperability between their EOSC services and/or research products) to a collection of guidelines that support machine-composability of resources. Capabilities offered for providers were:

- Onboarding of human readable guidelines: Providers can onboard structured metadata and links to human readable documents as Guidelines in the IF Registry.
- Linking of onboarded resources in the Service Catalogue to IF Guidelines onboarded in the IF Registry

For the new IF Registry currently deployed in EOSC Core Innovation Sandbox, new capabilities offered to providers include:

- Definition of Configuration Templates for an Interoperability Guideline. Guideline Providers can define one or more Configuration Templates for a specific Guideline based on a [specific definition schema](#). Currently, definition through the UI is done using a simple text editor, but this will change to a more structured and user-friendly editor in the second half of the project. Configuration Templates can also be created using the REST API of the Interoperability Registry.
- Definition of Configuration Instances for onboarded Services that adhere to Guidelines with Configuration Templates. Service Providers can instantiate configurations for their services based on the Guideline Configuration Template in the Providers Portal. Similarly, definition through the UI is done using a simple text editor for now. The same can be done using a REST API on the Service Registry.
- Configuration instances are now persisted in the Service Registry and are part of the data that the Service Catalogue shares with other Core Components (particularly the Front Office) through messaging or through API.

Thus, Guideline Providers can now create their own-defined structured metadata profiles for Guidelines that allow them to detail the actual access parameters of services that adhere to these Guidelines. Furthermore, Service Providers can follow these metadata to create machine readable configuration instances for their services. This allows machine-readable composability between services, an important step compared to the simple adherence to human-readable Guidelines.

### 3.2. Capabilities of the EOSC Deployment Service

The EOSC Deployment Service, as a component of the EOSC Execution Framework, has been deployed as part of the [EOSC Core Innovation Sandbox](#). For end users, this component is envisioned to be used from the Front Office to facilitate the deployment of complex application architectures on multiple Clouds. As such, the endpoint available at <https://deploy.sandbox.eosc-beyond.eu/> provides the following capabilities:

- A highly-available REST API to support Infrastructure as Code (IaC) orchestration to automate the deployment and configuration of virtualized infrastructures from high-level templates written in the OASIS TOSCA standard specification.
- Multi-cloud and hybrid-cloud support: Provisions resources seamlessly across a wide range of public and private clouds (OpenStack, OpenNebula, AWS, Azure, Google Cloud, Kubernetes, and more, including the EOSC EU Node).
- Interoperability and portability: Ensures vendor-neutral deployment through standardised TOSCA templates, enabling portability across heterogeneous cloud providers.
- Automatic configuration management: Integrates with the Ansible configuration tool to automate software installation and service configuration on the provisioned resources.
- Lifecycle management: The API allows scaling, updating, and terminating infrastructures during their lifecycle.

## D13.2 First report on EOSC Execution Framework Architecture and capabilities

- Dataset management coupled with infrastructure provision, to specify public datasets within TOSCA templates, enabling automated downloading and copying of reference data into the deployed infrastructure at runtime.
- Integration with the EOSC Data Transfer service to transfer publicly available datasets to storage systems closer to where the virtualized computing infrastructure will be provisioned from a Cloud provider.

In addition, since the EOSC Deployment Service is based on the Infrastructure Manager, the following additional capabilities are also available, and can be leveraged by users:

- A curated set of TOSCA templates (<https://github.com/grycap/tosca>) to deploy popular application architectures (e.g., Kubernetes clusters, SLURM-based clusters, Galaxy, HTCondor, Nomad, Scipion, Hadoop, etc.).
- A web-based dashboard that provides a catalog of these templates, allowing the user to manage the lifecycle of their deployment ([Figure 10](#)).

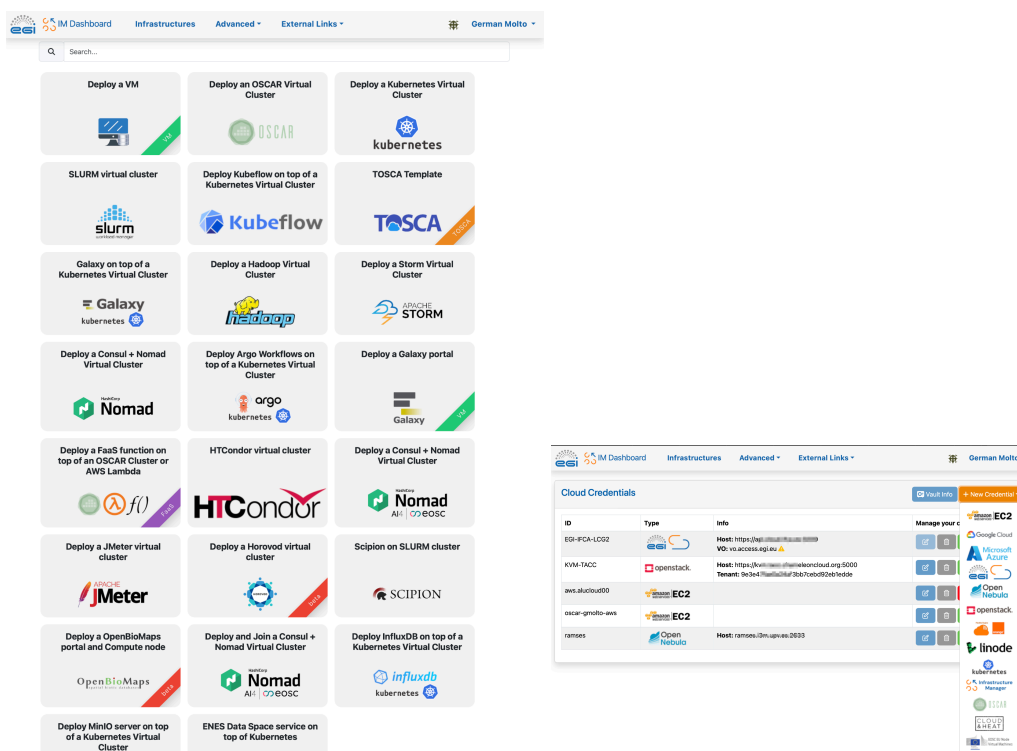


Figure 10 - Web-based dashboard of the Infrastructure Manager with the list of available TOSCA templates

## 4. Integration with the User Space

The EOSC Execution Framework is integrated with the User Space, so that providers and users can benefit from the capabilities described in the previous section.

### Provider Dashboard

The Provider Dashboard has been extended with functionalities to create and publish Deployable Services ([EBTR-270, Figure 11](#)).

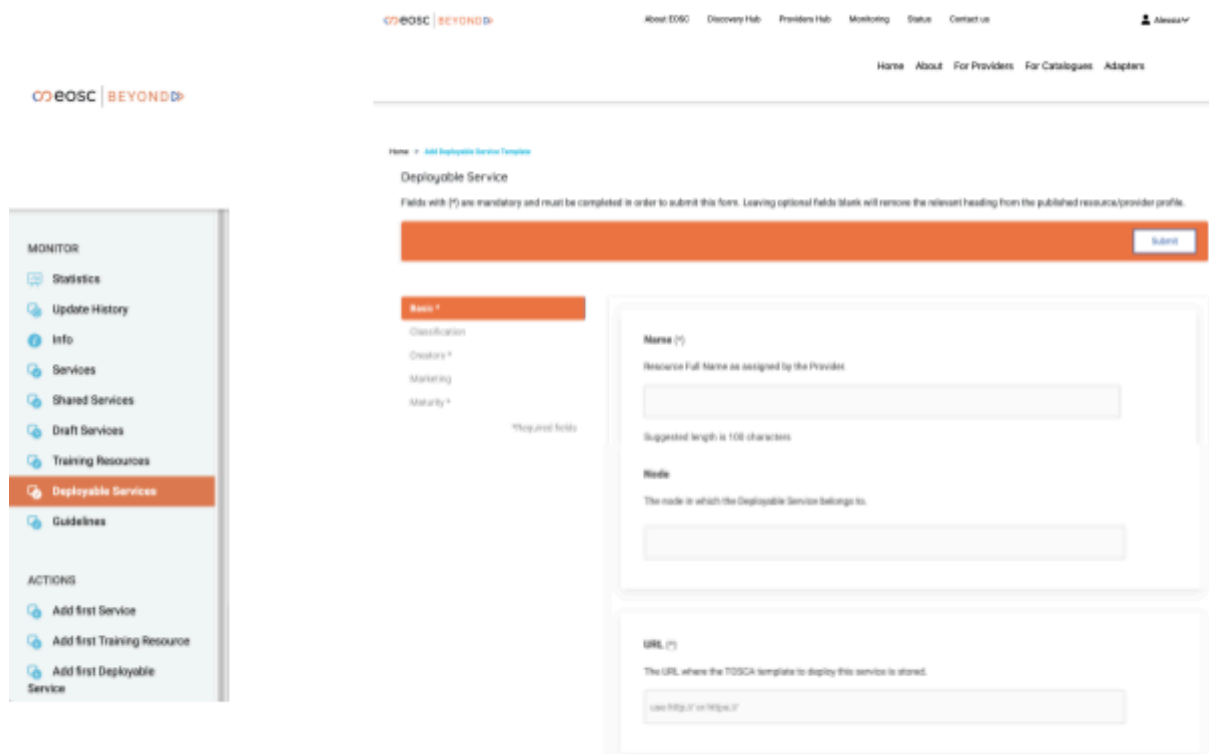


Figure 11 - Menu of the Provider Dashboard with Deployable Services and part of the form to create and publish a Deployable Service

### Discovery Hub

The Discovery Hub has been extended with:

- A dedicated list view for Deployable Services, with advanced search and filtering functionality ([Figure 12](#));
- A dedicated view with details of the Deployable Service selected by the user ([Figure 13](#));

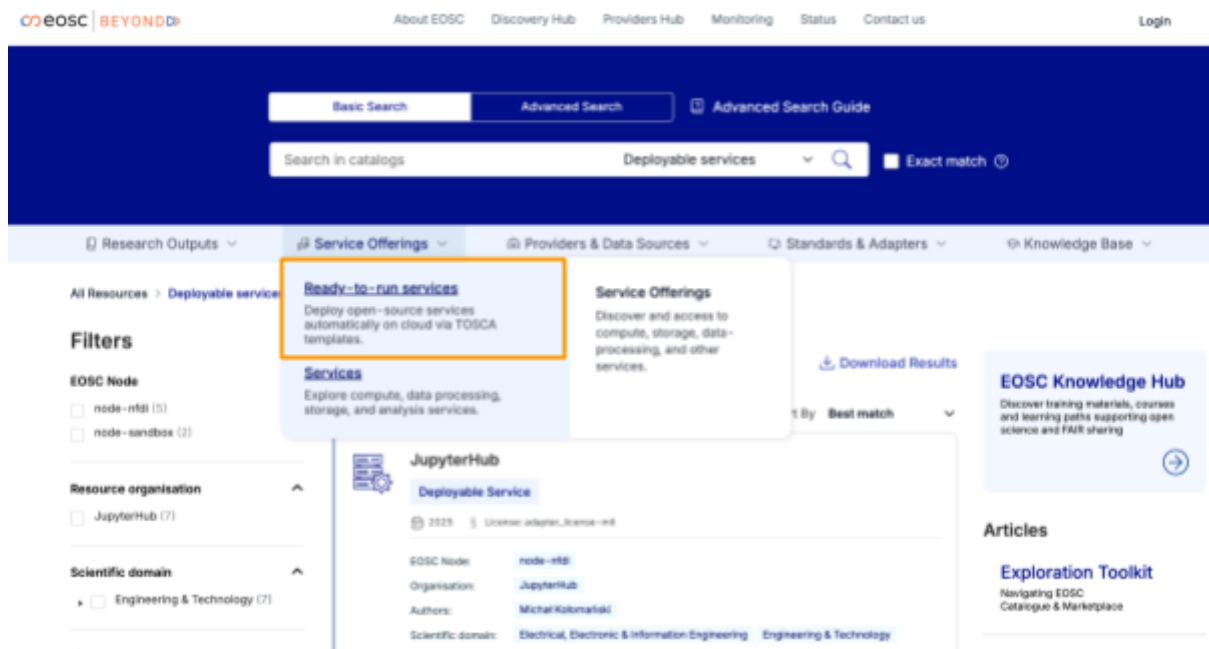


Figure 12 - Dedicated list view for Deployable Services in the Discovery Hub

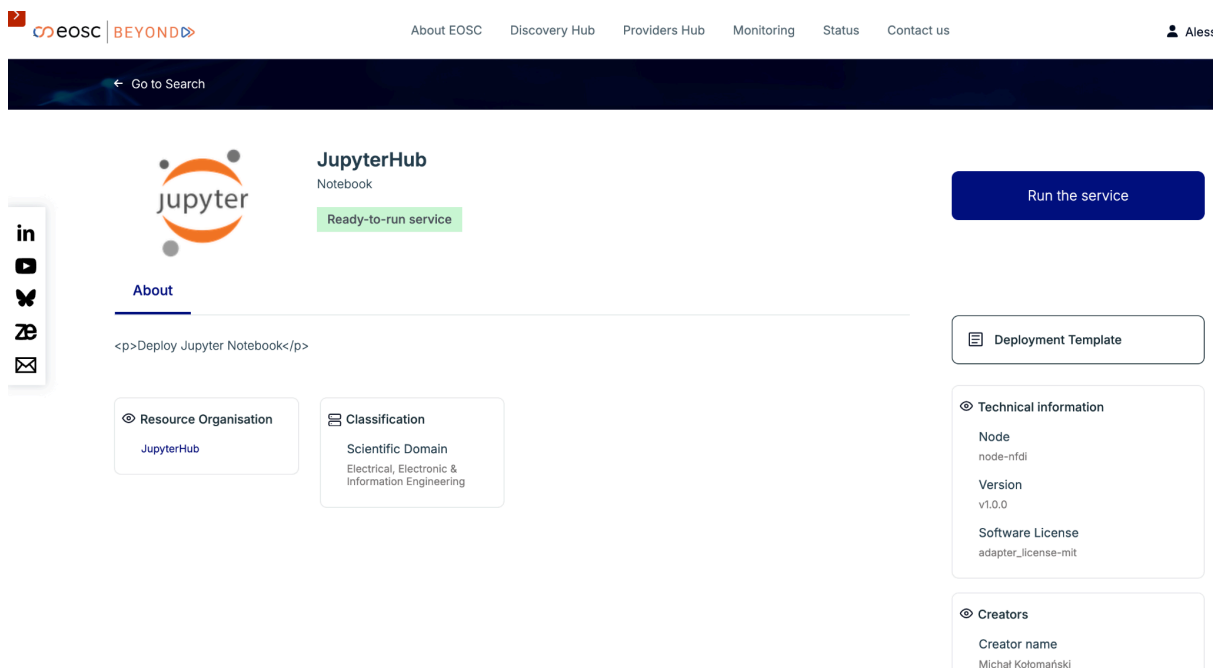


Figure 13 - Dedicated view with details of a Deployable Service

## Front Office

The Front Office has been extended to support the request of deployment of Deployable Services ([EBTR-270](#)):

- The Front Office automatically creates an offer for the Deployable Service selected by the user by fetching input values from the provided TOSCA template. The user must specify dynamic parameters of the TOSCA template and optionally provide DOIs of input datasets that will be automatically downloaded in the deployed infrastructure ([Figure 14](#)).

## D13.2 First report on EOSC Execution Framework Architecture and capabilities

- A section for users to view their projects and relative Deployable Service deployment requests (**Figure 15**);
- Detailed views for individual deployments of Deployable Services, with configuration information, order history and the option to contact the service provider (**Figure 16**).

The screenshot displays the EOSC BEYOND web interface for configuring a service deployment. The interface is divided into three main sections: 'Parameters', 'Access instructions', and 'Final details'.

**Parameters:** This section contains several input fields for configuring the service:

- Frontend CPU Cores:** A dropdown menu set to '2'.
- Frontend Memory:** A text input field set to '8 GB'.
- Frontend Disk Size:** A text input field set to '100 GB'.
- Number of Worker Nodes:** A text input field set to '2'.
- Worker Node CPUs:** A text input field set to '4'.
- Worker Node Memory:** A text input field set to '8 GB'.
- Worker Node Disk:** A text input field set to '20 GB'.
- Public DNS Hostname:** A text input field containing 'my-jupyter.eosc-beyond.eu'.
- JupyterHub Admin Password:** A password input field with masked characters.
- Dataset DOI List:** A text input field containing 'https://doi.org/10.5281/zenodo.722196'.

**Access instructions:** This section includes a 'Deploy JupyterHub' heading and a 'PARAMETERS' table that summarizes the configuration values entered in the previous section. Below the table, there is a 'MY PROJECTS' section with a dropdown menu for selecting a project (currently showing 'Data Analysis project') and an 'Add new project' button.

**Final details:** This section features a 'Send access request' button and a 'Ready-to-use research tool you can launch on cloud infrastructure in minutes, admin skills required.' message.

The interface also includes a progress bar at the top with three steps: 'Access instructions', 'Configuration', and 'Final details'. Navigation buttons include 'Cancel, Order and Quit', 'Next', and 'Back'.

Figure 14 - Forms of the Front Office to request the deployment of a Deployable Service

## D13.2 First report on EOSC Execution Framework Architecture and capabilities

The screenshot displays the user interface for a project named "Data Analysis project". At the top, there is a navigation bar with links for "About EOSC", "Discovery Hub", "Providers Hub", "Monitoring", "Status", and "Contact us". Below this is a dark blue header with a "Go to Search" button. On the left, a sidebar contains "My Dashboard" (with a right arrow), "My Projects" (with a sub-item "Data Analysis project"), and a "Create new project" button. The main content area shows the project title "Data Analysis project" with a "Created at 26.09.2025" timestamp, "Single user", and "CNR-ISTI" affiliation. Action buttons for "EDIT", "DUPLICATE", and "ARCHIVE" are visible. Below the project title are tabs for "RESOURCES", "PROJECT DETAILS", and "CONTACT WITH EOSC EXPERTS". The "Ordered services" section shows a "JupyterHub" service with a "DeployableService" label and a "Rejected" status. The "Recommended offers for your project" section features two offer cards: "Offer (Unlimited) ARIS - Archival service Offer" with a "Select an offer" button and a "DEEP REQUIRED" label, and "Offer (Unlimited) Incidence of alien species in Italian habitats and ecosystems Offer" with a "Select an offer" button and an "OPEN ACCESS" label. At the bottom, there is an "Add resource to this project" button.

Figure 15 - User projects and associated Deployable Service requests

The screenshot shows a detailed view of a deployment request for "JupyterHub". The navigation bar and header are identical to Figure 15. The sidebar shows "My Dashboard" and "My Projects" with "Data Analysis project" selected. The main content area has a "JupyterHub" title and a "REJECTED" status. A "back to Data Analysis project project services" link is present. Below the title are tabs for "DETAILS", "ORDER HISTORY", and "CONTACT WITH SERVICE PROVIDER". The "DETAILS" tab is active, showing a timeline of messages: "3 minutes ago: Your service request has been rejected. Please contact service provider if you have any questions" (Service request status: Rejected), "4 minutes ago: Your service request is under review by service provider" (Service request status: Under review), and "4 minutes ago: Your service request has been created" (Service request status: New).

Figure 16 - Detailed view of a deployment request

## 5. Conclusions and next steps

The initial release of the EOSC Execution Framework has achieved several key objectives.

The IF Registry, the Resource Catalogue and the User Space have been significantly enhanced to support the (manual) definition of Configuration Templates for Interoperability Guidelines (IGs). This also includes the creation of configuration instances for onboarded services by the Service Providers through the Providers Portal. Notably, specific Configuration Templates have been defined for the IG for onboarding research products and for service monitoring.

A new IG for TOSCA Deployable Services has been defined and added to the IF Registry. Services compliant with that IG are automatically deployable via the Deployment Service (DS). The User Space has been extended to allow (i) providers at creating and publishing Deployable Services; (ii) users at finding and requesting the deployment of Deployable Services.

The DS has been extended to automate the registration of fully qualified and secure domain names and to provide automated data staging capabilities, facilitating the fetching of datasets from open repositories directly to a dynamically provisioned virtual infrastructure.

Three composability-based applications demonstrate the practical utility of the extended Execution Framework.

The final implementation of the components and the composability-based applications will be carried out by Work Package 14 in the second and final phase of the EOSC Beyond project. The capabilities will be expanded to fully support the use case scenarios and epics for composability of services across the EOSC ecosystem outlined in the technical roadmap.

## Acronyms

Acronym	Meaning
AAI	Authentication and authorization infrastructure
API	Application Programming Interface
AWS	Amazon Web Services
DNS	Domain Name System
DS	Deployment Service
DTS	Data Transfer Service
EBTR	EOSC Beyond Technical Roadmap
EF	Execution Framework
EOSC	European Open Science Cloud
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
IF	Interoperability Framework
IG	Interoperability Guideline
IS	Integration Suite
NFDI	Nationale Forschungsdateninfrastruktur
OAI	Open Archive Initiative
OAI-PMH	Open Archive Initiative - Protocol for Metadata Harvesting
OASIS	Organization for the Advancement of Structured Information Standards
REST	REpresentational State Transfer
TLS	Transport Layer Security
TOSCA	Topology and Orchestration Specification for Cloud Applications
VM	Virtual Machine
WP	Work Package
YAML	YAML Ain't Markup Language™

Table 2: Acronyms