# D5.4: Final EOSC Service Architecture

| | |
|---|---|
| Author(s) | L. Candela (CNR), D. Castelli (CNR), F. Zoppi (CNR) |
| Status | Final |
| Version | v1.1 |
| Date | 08/04/2019 |

Dissemination Level

| | |
|---|---|
| X | PU: Public |
| | PP: Restricted to other programme participants (including the Commission) |
| | RE: Restricted to a group specified by the consortium (including the Commission) |
| | CO: Confidential, only for members of the consortium (including the Commission) |


Abstract:

*This deliverable is the second and final release of the EOSC Service Architecture. It sets the foundations characterising the EOSC System:*

- *(i) Its functionalities are provisioned as-a-Service;*
- *(ii) It is a highly distributed, evolving and heterogeneous hybrid cloud;*
- *(iii) Its operation and development is regulated by a set of Rules of Participation;*
- *(iv) It is modelled as an open and evolving System of Systems (SoS) where the component systems providing services include existing and emerging Research Infrastructures (including e-Infrastructures) and other types of Service Providers;*
- *(v) EOSC services provision is based on an open and evolving set of EOSC Nodes spread across several organisations and regions;*
- *(vi) EOSC Services should promote and support FAIRness.*

*The deliverable identifies 47 classes of services that can be considered at this stage of development as the "Minimal Viable Product" able to match the EOSC overall goal. Such services include cross-cutting services together with services specifically envisaged to serve researchers, research administrators, third-party service providers as well as EOSC managers, service providers and service suppliers.*

*This deliverable briefly highlights major contextual aspects already introduced in D5.1 and then describes the identified classes of services. The deliverable also discusses aspects related to "how" the system can/should be developed. The notions of "federation" and "interoperability" related to the building of this EOSC System are addressed highlighting the importance of dealing with these two concepts per-single service rather than from the perspective of EOSC as a whole.*



The European Open Science Cloud for Research pilot project (EOSCpilot) is funded by the European Commission, DG Research & Innovation under contract no. 739563

| Document identifier: EOSCpilot -WP5-D5.4 | |
|---|---|
| Deliverable lead | **CNR** |
| Related work package | **WP5** |
| Author(s) | L. Candela (CNR), D. Castelli (CNR), F. Zoppi (CNR) |
| Contributor(s) | T. Ferrari (EGI), D. Lecarpentier (CSC), A. Lukkarinen (CSC), L. Mack (JISC), L. Matyska (CESNET), C. Meghini (CNR), M. Molinaro (INAF), G. Papastefanatos (Athena), D. Schaap (MARIS), M. Williams (GEANT) |
| Due date | **31/01/2019** |
| Actual submission date | **08/04/2019** |
| Reviewed by | **EOSCPilot WP5 partners** |
| Approved by | **Mark Thorley (UKRI)** |
| Start date of Project | **01/01/2017** |
| Duration | **28 months** |

## Versioning and contribution history

| Version | Date | Authors | Notes |
|---|---|---|---|
| 0.1 | 03/12/2018 | L. Candela (CNR), D. Castelli (CNR) | Initial draft prepared |
| 0.5 | | T. Ferrari (EGI), C. Meghini (CNR), L. Mack (JISC), M. Molinaro (INAF), G. Papastefanatos (Athena), D. Schaap (MARIS) | Contributions to Appendix A |
| 0.7 | | L. Candela (CNR), D. Castelli (CNR), F. Zoppi (CNR) | Finalization of the document |
| **0.8** | | D. Lecarpentier (CSC), A. Lukkarinen (CSC), L. Matyska (CESNET), M. Williams (GEANT) | Comments on the document circulated |
| **1.0** | 20/03/2019 | L. Candela (CNR), D. Castelli (CNR) | Final version submitted to Project Office |
| **1.1** | 08/04/2019 | M. Thorley (UKRI) | Final typographic proof-read and edit. |

# TABLE OF CONTENT

**EXECUTIVE SUMMARY** ..... **6**

**1. INTRODUCTION** ..... **7**

**2. ACTORS AND EXPLOITATION PATTERNS** ..... **9**

**3. THE EOSC SYSTEM ARCHITECTURE** ..... **12**

3.1. A view of the overall architecture ..... 12
3.2. Federation and Interoperability considerations ..... 15
3.3. Deployment considerations ..... 16

**4. EOSC CLASSES OF SERVICES** ..... **18**

4.1. EOSC Cross Cutting Services ..... 18
4.2. Services for Researchers ..... 22
4.3. Services for Research Administrators ..... 26
4.4. Services for Third-party Service Providers ..... 28
4.5. Services for EOSC System Managers (including EOSC Service Providers) ..... 29
4.6. Services for EOSC Suppliers ..... 31

**5. CONCLUDING REMARKS** ..... **32**

**REFERENCES** ..... **34**

**APPENDIX A. SELECTED SERVICES DETAILED ARCHITECTURES** ..... **36**

A.1. Registries ..... 36
A.1.1: The IVOA Resource Registry (by M. Molinaro) ..... 37
A.1.2: The eInfraCentral Service Catalogue (by G. Papastefanatos) ..... 39
A.1.3: The EOSCpilot Policy Registry (by L. Mack) ..... 42
A.1.4: Considerations on Registries ..... 46
A.2. VRE Services, Science Gateways, Portals ..... 46
A.2.1: The SeaDataCloud VRE (by D. Schaap) ..... 47
A.2.2: The VRE4EIC VRE model (by C. Meghini) ..... 50
A.2.3: The D4Science VRE Manager (by L. Candela) ..... 53
A.2.4: Considerations on VREs, Science Gateways and Portals ..... 55
A.3. EOSC Resources Management, Hosting Platforms ..... 55
A.3.1: EGI Federated Cloud Compute (by T. Ferrari) ..... 56
A.3.2: The EOSC-hub Marketplace (by T. Ferrari) ..... 59
A.3.3: Considerations on EOSC Resources Management ..... 64

# LIST OF FIGURES

Figure 1. EOSC Exploitation patterns ..... 10
Figure 2. EOSC Services: the user perspective ..... 13
Figure 3. EOSC Services: the functional perspective ..... 14
Figure 4. Matchmaking between provisioning and consumption perspectives in EOSC ..... 15
Figure 5. The eInfraCentral Service Catalogue architecture ..... 40
Figure 6. The EOSCpilot Open Science Policy Registry service architecture ..... 43
Figure 7. SeaDataCloud VRE: Diagram of layers in the architecture ..... 48
Figure 8. VRE4EIC Technical Architecture ..... 52
Figure 9. D4Science VRE Manager ..... 53
Figure 10. EGI Federated Cloud Architecture ..... 57
Figure 11. EOSC Marketplace service providers' interface ..... 61
Figure 12. EOSC Marketplace users' interface ..... 62
Figure 13. EOSC Marketplace Service Architecture ..... 63

# LIST OF TABLES

Table 1. EOSC Cross Cutting Services (alphabetical order)..... 19
Table 2. EOSC Services for Researchers (alphabetical order)..... 22
Table 3. EOSC Services for Research Administrators (alphabetical order)..... 27
Table 4. EOSC Services for Third-party Service Providers (alphabetical order) ..... 28
Table 5. EOSC Services for System Managers (alphabetical order)..... 29
Table 6. EOSC Services for Service Providers (alphabetical order)..... 30
Table 7. EOSC Services for EOSC Suppliers (alphabetical order) ..... 31
Table 8. Policy Registry application use cases ..... 45

## EXECUTIVE SUMMARY

This deliverable is the second and final release of the EOSC Service Architecture. It builds upon the foundations defined in the Initial EOSC Service Architecture (deliverable D5.1) for the EOSC IT System. In particular, the following features characterise the EOSC System:

(i) *Its functionalities are provisioned **as-a-Service**, i.e. they are made available by online services operated by providers taking care of the technical and organisational approaches needed to deliver the planned functionalities;*

(ii) *It is a highly distributed, evolving and heterogeneous **hybrid cloud**, i.e. it is an evolving mixture of on-premises and off-premises IT resources and organizations mobilized to deliver the envisaged services;*

(iii) *Its operation and development is regulated by a set of **Rules of Participation**;*

(iv) *It is modelled as an open and evolving **System of Systems** (SoS) where the component systems providing services include existing and emerging Research Infrastructures (including e-Infrastructures) and other types of Service Providers;*

(v) *EOSC services provision is based on an open and evolving set of **EOSC Nodes** spread across several organisations and regions;*

(vi) *EOSC Services should promote and support **FAIRness**, e.g. the EOSC Services managing data should implement the FAIR principles.*

The starting point of the whole work performed in Task 5.1 has been the identification of the functionality that the EOSC IT System is expected to support. To meet this goal the typology of EOSC users (**user roles**) and their basic functional requirements have been first identified. The analysis of requirements has identified **47 classes of services** that can be considered at this stage of development as the "**Minimal Viable Product**" able to match the EOSC overall goal. Such services include cross-cutting services, together with services specifically envisaged to serve researchers, research administrators, third-party service providers as well as EOSC managers, service providers and service suppliers.

This deliverable briefly highlights major contextual aspects already introduced in D5.1 and then describes the identified classes of services. The deliverable also discusses aspects related to "how" the system can/should be developed. The notions of "federation" and "interoperability" related to the building of this EOSC System are addressed, highlighting the importance of dealing with these two concepts per single service rather than from the perspective of EOSC as a whole.

An appendix presents the concrete architecture of a few of EOSC identified services by describing how they are built in existing infrastructures - often as a federation/composition of different components. The aim of this appendix is to provide suggestions to those that are currently developing these kinds of services to serve the overall EOSC users, as well as to stress that the development of EOSC is a complex task both when considered as a whole as well as when considered per single service.

In fact, when designing and developing every single service possibly belonging to one of the envisaged service classes, it should be carefully considered that this service will be called to operate in the EOSC settings and this might pose additional requirements with respect to usual settings. For example, scalability, quality of service, compliance with standards and practices, and provisioning models are examples of aspects driving the development of EOSC Services. Moreover, the implementation of every single service might pose new requirements to every single component of the entire resource stack underlying it. For example, power acquisition, cooling and network connectivity are problems requiring sound solutions on par with the problems stemming from the functional aspects of every single service.

# 1. INTRODUCTION

The availability of new IT technologies and high-performance computing are changing the way research is performed and knowledge is shared. In parallel, the recognition that research results and services produced, and being produced, by decades of European and National public investments should become exploitable by everybody, is offering new opportunities to all European researchers, innovators and citizens. In this new Open Science framework data dissemination and analytics support are meant to be a catalyst for addressing big scientific challenges, economic growth and innovation across all economic sectors, particularly for small and medium-sized enterprises (and start-ups) and for society as a whole.

The European Open Science Cloud (EOSC) aims at becoming a trusted, open environment leveraging both existing technologies and available resources for making this framework concrete and achievable. In order to meet this objective, it will have to offer supporting services to all the actors both contributing to and exploiting Open Science.

The purpose of this deliverable – a revised and extended version of D5.1 "Initial EOSC Service Architecture[1]" – is to illustrate the evolution of the EOSC System Service Architecture as a result of the deeper analysis of the EOSC system and its scope, and of the experiences acquired in carrying out related initiatives during in the last twelve months. D5.1 aimed to lay the foundations for a shared interpretation of EOSC and of what functionality a system supporting EOSC should provide to its users. As a matter of fact, when the EOSCpilot project started, the interpretation of EOSC by the project partners themselves was quite heterogeneous. Through different meetings, events and interactive remote discussions a certain level of shared understanding was reached and presented in D5.1. In order to further progress this shared understanding, it was also decided to launch a joint effort for the construction of a common Glossary.

This second release of the EOSC System Architecture is meant to review and consolidate the results of the first release, also taking into account the project review recommendations. In particular, it improves the previous one primarily as follows:

- The interaction among the different actors playing a role in the Open Science Cloud and their activity workflows are clarified;
- The classes of services have been revised along multiple dimensions:
    - Requirements under the form of "characterizing principles" to be addressed have been better taken into account, e.g. FAIRness and Open Access, also analysing the different reports that have been released in the meantime by dedicated High Level Expert Groups [EC2018b];
    - Services offered to facilitate Service Providers in performing their activities have been revised. In particular more attention has been dedicated to services supporting Data Service Providers;
    - The distinction among different incarnations of the same class of services have been clarified (e.g. data and service catalogues).
- A more in-depth discussion about possible deployment models has been added envisaging the role that different infrastructures (i.e. e-Infrastructures and Research Infrastructures) and other service providers might play in building the EOSC.
- For selected typologies of services of particular relevance, a detailed design is presented, showing alternative architecture solutions for them as developed in existing contexts. In particular, it is shown how these services are often built by exploiting federated approaches. These solutions are intended to provide insights about how similar the architecture of this typologies of EOSC services should be structured.

[1] See [Candela et al. 2018].

For the sake of brevity, we have included in this deliverable only new and revised contributions. We assume that the reader is familiar with the content of D5.1.

The rest of the deliverable is organised as follows: Section 2 recalls actor roles and exploitation patterns. Section 3 presents the overall Architecture Model and discusses on its deployment. Section 4 describes the classes of services envisaged in the architecture and, whenever possible, give examples of suitable technologies or existing instances. Finally, Section 5 concludes by reporting some remarks and future steps.

Appendix A illustrates the detailed architecture for a number of selected services by analysing their existing implementation. The intent of such is twofold; on one side it aims at highlighting that the implementation of EOSC is a complex task both when considered from the 'system as a whole' perspective as well as from the 'single service' perspective. On the other side it aims at suggesting that the implementation of every single EOSC service will pose a number of research questions pertaining to the entire stack of resources needed to operate it, from power acquisition to network connectivity and computing, before addressing the functional aspects peculiar of the service. Thus, although EOSC system is likely to be an "as easy as possible" system to use for its users, its implementation and operation is an extremely challenging endeavour.

# 2. ACTORS AND EXPLOITATION PATTERNS

The overall EOSC IT system is obtained by exploiting a number of EOSC services that may evolve over time. An ***EOSC Service*** is a primary resource that provides ready-to-use functionalities enabling and facilitating scientific activities performed in line with Open Science principles.

Each EOSC Service is operated by an actor (usually an Organisation) that has agreed to play the role of ***EOSC Service Provider***. Such an actor is required to make available the target service in accordance to the established EOSC Rules of Participation. This means, for example, that it has registered the provided service in an EOSC compliant or compatible service catalogue visible to the global EOSC gateway, and that it has associated each EOSC service with a Service Level Agreement (SLA) that specifies the type and level of service that the user will receive from the Service Provider.

Note that, as indicated in the "Implementation Roadmap for the European Open Science Cloud" [EC SWD 2018], Research Infrastructures (RIs) can act as Service Providers. By making their services accessible through EOSC RIs widen the opportunity of their users reaching other market segments. In so doing, RIs may more widely disseminate the outcomes delivered by the researchers of their community and demonstrate the relevance of their services, thus also contributing to their sustainability.

One of the key mandates of the EOSC system is to empower and facilitate the tasks of the Service Providers. Nowadays services are rarely developed from scratch, rather they are built by exploiting and composing in well-defined workflows simpler components operated by external suppliers. This approach makes developments simpler and faster so facilitating the competitiveness of the service providers. In the EOSC context, providers of components are named ***Service Suppliers.*** To support this new service development approach the EOSC system offers facilities enabling components made available by different Suppliers to be combined in multiple ways, and to operate the obtained result according to EOSC Rules of Participation. The support can be of different degrees, from simple transformation of formats and interfaces, to the inclusion of components complementing the logic, to definition and implementation of specific credits and compensation. The relation between the provider of the new service and the suppliers of its components (in the context of the specific EOSC Service) is regulated by ***Underpinning Agreements.***

EOSC provide services to users playing different roles. These include:

(i) Researchers, i.e. the target beneficiaries of EOSC, and more in general, end-users that consume EOSC services for conducting their open science and innovation activities;
(ii) Resource providers that make available different type of services (e.g. computing, data, software) to EOSC;
(iii) Research administrators that are interested to analyse and monitor how research activities perform;
(iv) Third-party service providers that use EOSC services to develop new services, that in turn, may become EOSC services.

EOSC also makes available services for simplifying the work of those that are responsible for EOSC system management, like the *EOSC System Owner*, responsible / accountable for the establishment and maintenance of the EOSC System, and the *EOSC System Top Manager* responsible / accountable for the overall operation of the EOSC System. Both these roles can be collective roles, where multiple actors contribute to them.

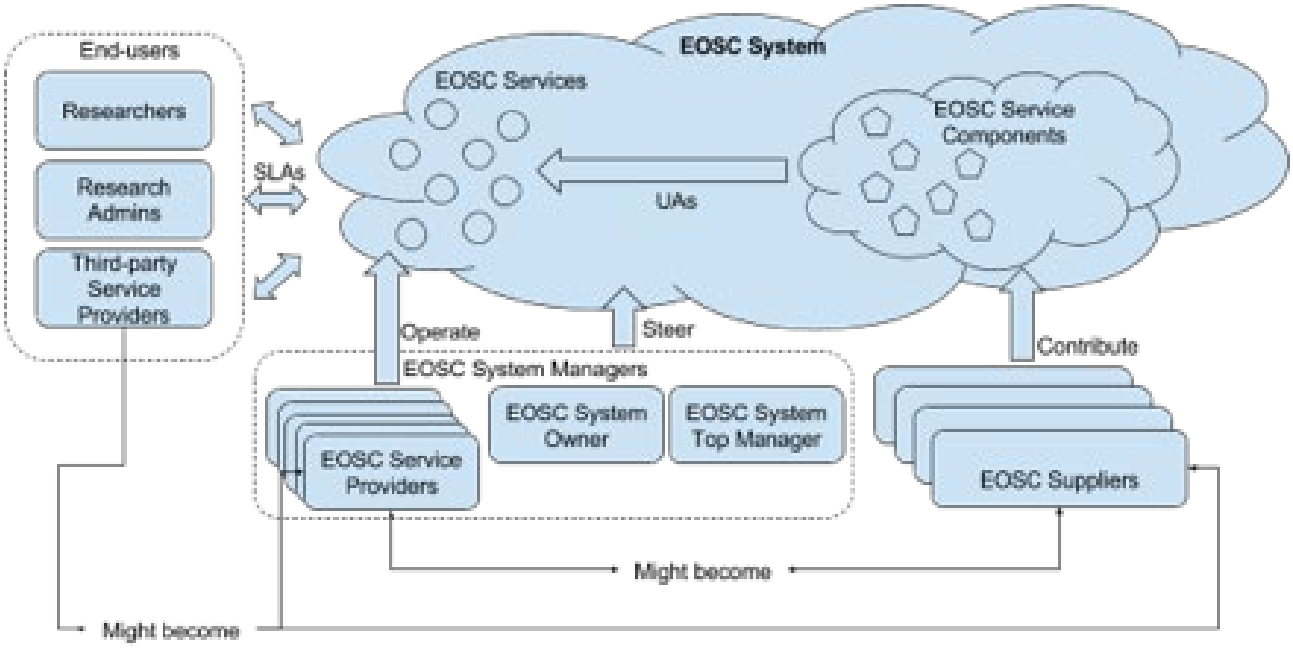


**Figure 1. EOSC Exploitation patterns**

Examples that illustrate typical EOSC usage and exploitation patterns are described below and shown in Figure 1.

***Example 1. Research Infrastructures as consumers and providers of EOSC services***

The manager of a Research Infrastructure (RI) identifies the need for a new service addressing the demand of a group of scientists affiliated with the RI. She commissions the development of this service by an organization (it will play the role of *Third-party Service Provider*) that chooses to rely on some existing EOSC services to develop and operate the target service. For example, by using the *EOSC Resource Management* (see Section 4.1) the Third-party Service Provider acquires the Hosting Platform (see Section 4.4) providing computing and storage capacity on EOSC premises. Once the new service is developed, the Governance of the Research Infrastructure decides to make it available to the corresponding community of scientists, agreeing to operate the service in accordance with the EOSC Rules of Participation, making the new service available as an EOSC resource. The RI thus becomes an EOSC Service Provider for the newly developed service, to be exploited by other users, including other Research Infrastructures.

***Example 2. EOSC Federated Service Catalogue***

The EOSC System Owner, analysing the needs of the EOSC Stakeholder communities, identifies the requirement to make available a catalogue of EOSC services (see Section 4.1). A Service Provider is identified for such a service, and it decides to implement this service by aggregating content from multiple complementary service catalogues operated by Research Infrastructures and other organizations. These individual catalogues are not necessarily exposed as EOSC services since their focus is specific to the RIs' communities, as are the rules regulating them. In this scenario, the RIs act as Suppliers for specific catalogue components and Underpinning Agreements regulate their relationship with the Provider of the EOSC Catalogue of Services.

***Example 3. Service Certification***

Organizations may decide to participate in the EOSC as Service Providers. They could agree to undergo certification and the establishment of a certified monitoring of service usage. EOSC provides EOSC System Managers, and in particular Service Providers, with dedicated (and in turn certified services) to perform such activities (see Section 4.5). The form of certification varies according to the nature of the service, for example, certain services may be certified for the level of FAIRness in data management, while others for reliability and timeliness of the response.

***Example 4. Becoming an EOSC Service Provider***

Links between dataset and literature objects, and dataset and dataset objects are emerging as core elements for facilitating reusability. Many organizations, like publishers, data centres, and other service providers (e.g. CrossRef, DataCite, and OpenAIRE), offer services to store and manage such links for a specific context. Given the importance of such information, the EOSC System Owner decides that it is strategic to add to the EOSC offer a service that enables the storage, search and retrieval of such links for the overall EOSC data space. The provider of the existing Scholexplorer[2] service could be a potential candidate for empowering and operating such service. This service aggregates link metadata harvested from a number of relevant data sources and out of these builds a harmonised and de-duplicated graph of scholarly objects. The Provider of this service would need to agree to participate in EOSC as one of the many "doers". Thought the Policy Toolkit (see Section 4.3), one of the EOSC services, the Scholexplorer Provider is validated with respect to the EOSC Rules for Participation and it can then become an EOSC Provider. In this scenario, these harvested data sources act as Suppliers for the EOSC Scholexplorer Service.

***Example 5. Open Science-friendly Research Activity by EOSC***

A researcher (Dr. Smith) is interested in investigating the impact of a certain alien species in the Mediterranean and announces this interest by a post (Social Networking, see Section 4.2). He also creates a dedicated working environment (VRE) relying on EOSC resources to facilitate this research endeavour (VRE Management, see Section 4.2). Two researchers, Dr. Green and Dr. Rossi, from two different research organizations, are interested in collaborating with Dr. Smith and decide to contribute to the VRE by organizing and populating a shared folder with suitable material, e.g. datasets, notes, papers (Workspace, see Section 4.2). Besides this, Dr. Smith and Dr. Rossi propose two diverse models aimed at capturing the effects of the alien species on the Mediterranean ecosystem. They implement their models and make these versions available for others by relying on the data analytics workbench supporting the VRE (see Section 4.2). The availability of these "resources" (namely datasets and models) leads another researcher (Dr. Bahl) to start a study on another species he had previously developed a model for, creating another workspace folder with specific material and producing another version of Dr. Rossi's model.

By relying on the VRE offering, Drs. Smith, Green and Rossi execute a large set of concurrent analyses, making available the resulting datasets by using several tools for sharing research outputs, including workspace, repositories and registries. They announce their findings by preparing a non-traditional paper with the support of the Publishing support services (see Section 4.2). Meanwhile, a fifth researcher (Dr. Wang) starts reusing the model(s) produced by Drs. Smith et al. as well as Dr. Bahl's model to analyse certain datasets she owns. She identifies a potential implementation issue affecting all of the models, produces and publishes corrected versions, and "annotates" the initial models with her findings. Alerted by Dr. Wang's annotations, Drs. Smith et al. decide to re-execute their experiments on other datasets by using both their version of the models as well as Dr. Wang's version. As a result, they realize that Dr. Wang's model better suits their initial hypothesis. All of this happens well before their paper is finally published. Moreover, all of this occurs without spending researcher's effort is setting up the working environment and the computing environment, which is done by EOSC.

[2] See https://dliservice.research-infrastructures.eu

# 3. THE EOSC SYSTEM ARCHITECTURE

The primary goal of Task 5.1 is to specify the architecture model of the EOSC IT system (the ***What***) rather than a specific implementation of it (the ***How***). This section presents a revised version of the overall architecture model initially presented in D5.1 [Candela et al. 2018]. In particular, it describes an EOSC architecture developed through collecting feedback on the version described in D5.1, from learning from the experience of more recently funded EOSC-related projects, and from the progresses of the EOSCpilot science demonstrators. In particular, Section 3.1 introduces the final version of the architecture envisaged for EOSC by proposing ***47 classes of services***. Section 3.2 discusses the two characterisations usually accompanying EOSC, namely ***federation*** and ***interoperability***. Section 3.3 includes a discussion about the ***deployment*** of the proposed architecture.

A detailed presentation of the services included in the proposed architecture is in Section 4, and some of the services are further analysed in Appendix A (i.e. Registries, Science Gateways, and Hosting Platforms) where possible concrete architectures (the ***How***) are illustrated, based on existing implementations.

## 3.1. A view of the overall architecture

The architecture model has been developed from the perspective of EOSC users, by identifying the components that are needed for supporting and facilitating all the actors involved in both implementing an open approach to science (including the FAIR management of data) and in developing and operating the overall EOSC system in a sustainable way.

The architecture model presented here is an update of the model presented in D5.1, and reflects the improved understanding of the functional needs, the core services, and the rules regulating EOSC developed since D5.1 was produced. This includes work undertaken in other project tasks, e.g. Task 5.4 [Ferreira et al 2019].

The final version of the model envisages ***47 classes of services*** needed to develop and operate a system suitable to support the EOSC mission and goal. The service classes have been updated, though the overall per-service role orientation has been retained as one of the perspectives leading to their identification. New classes of services have been added, whilst others have been modified or clarified. In particular:

- EOSC Core Services have been renamed as “EOSC Cross-Cutting Services” to stress that these are classes of services that any user requires. Services in this class can take different forms and operate on different resources according to the specific role of the actors they are intended for. A Portal service has also been added to this class. This represents the main and generic access point for users to all the EOSC provided services, e.g. registries, support services and, more in general, Open Science working environments.
- Services for researchers have been better described stressing more expected functionality for these services, like the automatic support for deposition, provenance, access to linked data, intelligent search, cross-domain profiling, etc.
- The class of services for Service Providers have been enriched by specifically focussing on the needs of the providers of data repositories in ensuring FAIR data management and its assessment - where “data” means not only datasets, but also software, articles, and any type of multimedia content.
- Services for Research Administrators have been extended. In particular, a new class of service named FAIRness Monitoring (see Section 4.3) has been envisaged to support the tracking of FAIR related policies.

It is important to note that this architecture model expresses a conservative view of Open Science services. It reflects practices and functionalities that are available today to support the Open Science approach. In this sense, it represents what is named a ***Minimal Viable Product*** [EC 2018a], i.e. a product with just enough features to satisfy early customers, and to provide feedback for future product development. It is expected that in the near future many innovative functionalities and classes of services will emerge, as a consequence

of a better understanding of new Open Science practices, and the expansion of the new market of Third-party service providers.

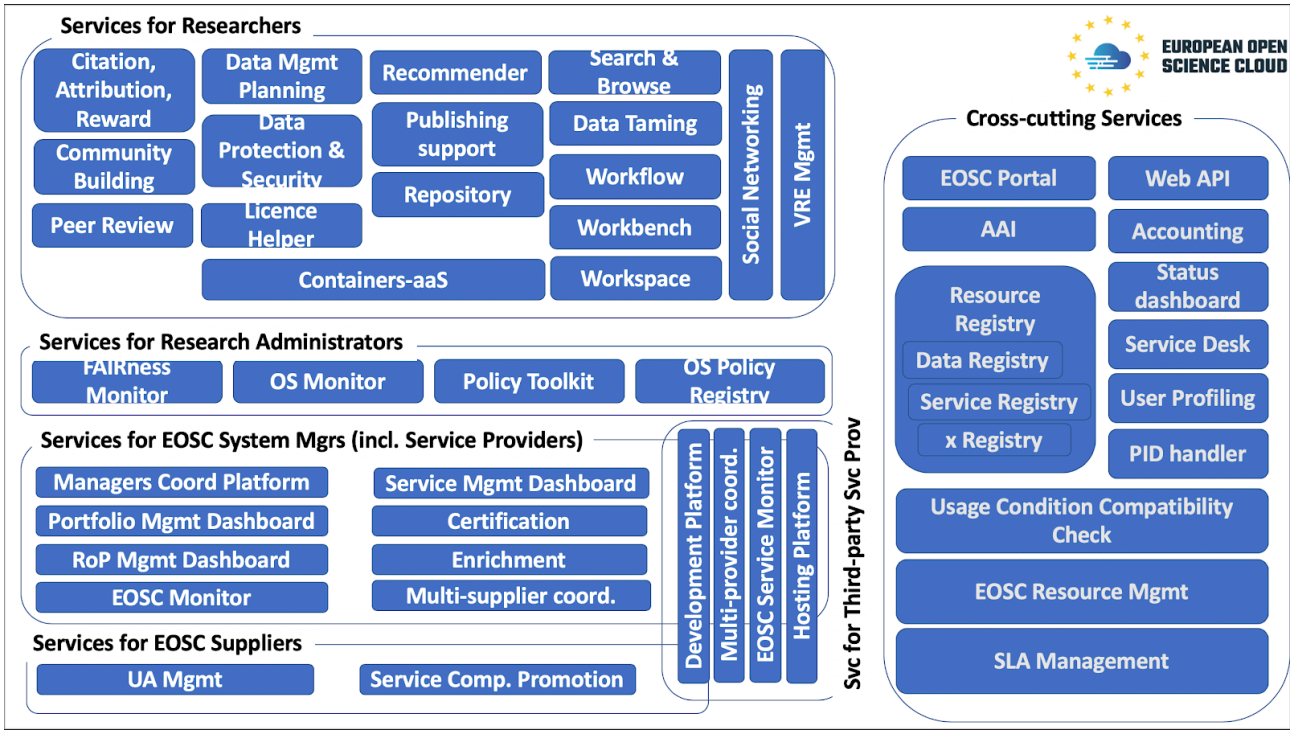


**Figure 2. EOSC Services: the user perspective**

The same set of service classes depicted in Figure 2 can be organised by clustering them according to a functional perspective (see Figure 3). Overall, the EOSC System is offering services belonging to the following classes of service:

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

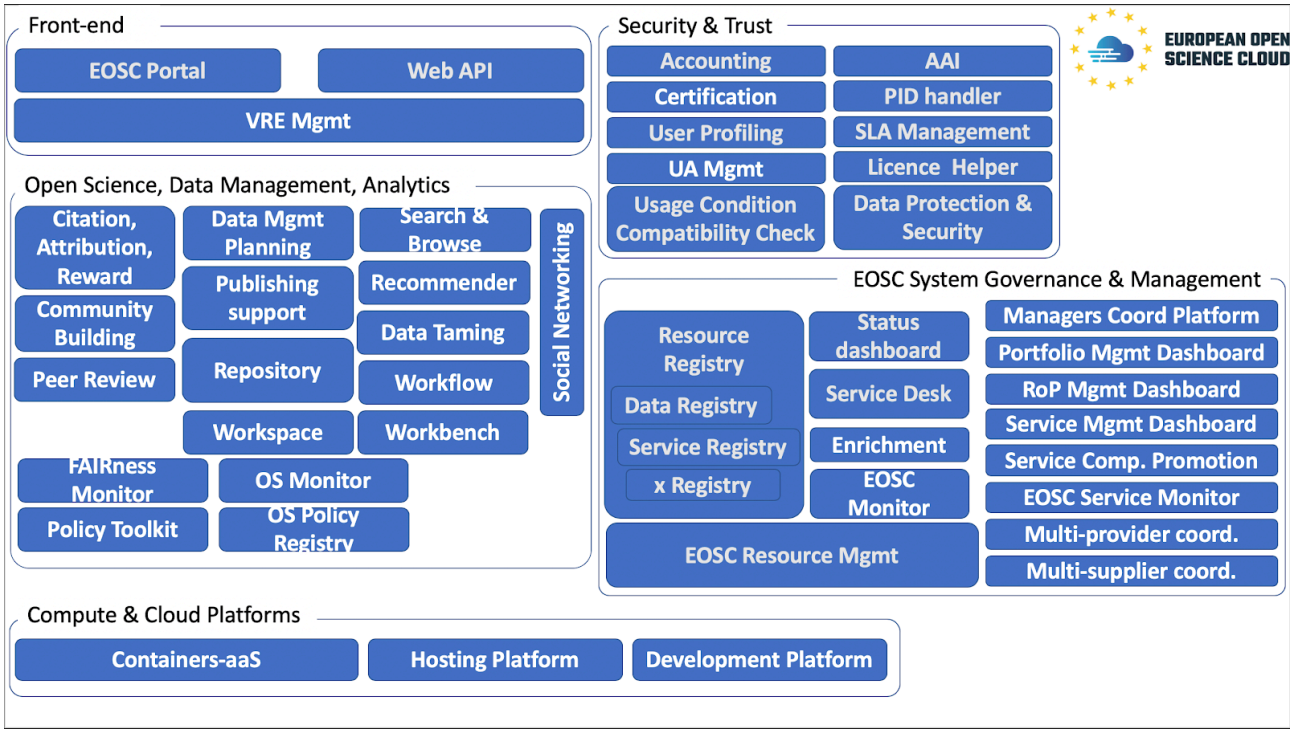


**Figure 3. EOSC Services: the functional perspective**

In addition to the discussions on functional characterization of the classes of services, it is worth clarifying that the EOSC is likely to include services conceptually falling in diverse ***levels of abstraction*** of a typical cloud stack. Figure 4 shows a layer-oriented generalization of the "as-a-Service" model as it appears in a number of articles in the literature, e.g. [Duan et al. 2015]. EOSC Services can be in any of these layers, depending on the per-service specifics. Figure 4 shows that EOSC Services as well as EOSC Service Components can be at any level of abstraction from infrastructure-facilities (including network transport connectivity) up to software, data, applications, etc.

When analysing the exploitation scenarios from the perspective of End-users, the use-cases range from the lower one in which EOSC services provide users with typical IaaS facilities (Containers-as-a-Service see Section 4.2) - in this case users use them to build and maintain the higher level facilities they need, to the upper one where EOSC is providing users with the entire solution they need at application level.

Similar cases emerge when using the EOSC Service Provider perspective: scenarios range from cases where an EOSC Service Provider is counting on IaaS facilities operated by EOSC Suppliers, and then has to build and develop what is needed to operate the EOSC Service, to cases where the EOSC Supplier is providing the entire facility to the EOSC Service Provider.

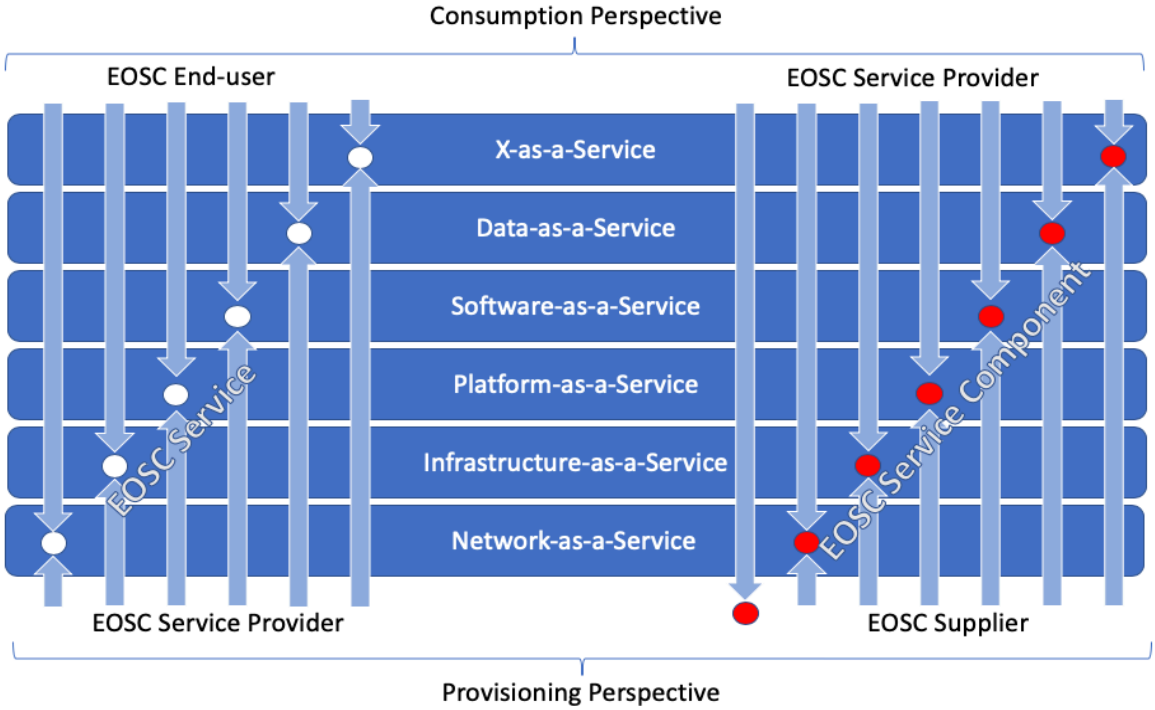


**Figure 4. Matchmaking between provisioning and consumption perspectives in EOSC**

## 3.2. Federation and Interoperability considerations

The EOSC System picture described in the previous section focuses mainly on the "***what***", describing what is needed to meet the EOSC's planned mission and goals. The transition from "***what***" to "***how***" requires several aspects to be addressed, ranging from organizational to operational. All these aspects deal with human oriented policies and procedures up to IT-based decisions and solutions.

One of the concepts discussed in relation to the "***how***" of the EOSC System is that of ***"federation"***.

As federation is a broad concept covering organizational and implementation aspects and, at times, it is difficult to identify what exactly different people mean when they use the term to describe the EOSC. This can lead to misunderstanding. However, very often, behind the use of "federation" to describe the EOSC there is the acceptance that ***it is neither feasible nor viable to develop the EOSC System from scratch***.

The reasons for this include:

*(i)* A lot of effort and resources have already been spent in developing (sustainable and scalable) solutions and services for some of the identified service classes;

*(ii)* A single provider is unlikely, on its own, to have the capacity needed to address the load and demand EOSC users will have;

*(iii)* The development of a new service for the envisaged classes with enough capacity might require an investment in time and resources that is particularly risky without a consolidated EOSC.

Thus, federation is likely to be used to suggest an EOSC System development model where there are a range of providers contributing to the development of the system and the delivery of the envisaged services. This is further discussed in Section 3.3, where the EOSC System is described as an evolving *hybrid cloud* of multiple resources.

Developing a "**federation**" requires that the system is constructed through making the federated entities ***"interoperable"***. However, interoperability is another broad and possibly problematic concept. For example, *technical*, *legal*, *semantic*, *syntactic and cross-domain* are all specific areas of interoperability.

Interoperability is not an absolute concept, rather it is highly ***partial*** and ***relative***. It is meaningless to state that two "systems" are interoperable. Two systems are interoperable given well-defined settings (an interoperability scenario) including:

*(i)* One of the two system is playing the role of provider while the other is playing the role of consumer;
*(ii)* The two systems are in agreement with respect to a specific item they want to "exchange";
*(iii)* The agreement is suitable for the consumer's intended use of the provider's item;
*(iv)* The approach/solution they decide to use is the same/compatible.

One of the consequences of this is that there is no generic interoperability solution suitable for any context.

There are two major classes on interoperability solutions with their individual pros and cons: ***"Agreement-based"*** approaches and ***"Mediator-based"*** approaches.

***"Agreement-based"*** approaches are based on the two entities following shared guidelines or implementing common standards. In so doing, the provider offers exactly the "the kind of plug" that fits the consumer's "kind of socket", so removing any heterogeneity barrier that may hamper the required level of interoperability. ***"Mediator-based"*** approaches work by exploiting adapters, i.e. special-purpose gateways that connects the producer's "kind of plug" with the different "kind of socket" supported by the consumer.

These classes are the basis of any solution adopted in practice. Real-world interoperability scenarios are often complex and require the combination of multiple simpler approaches to be resolved. These can belong to either of the two classes. In some cases, agreement and mediator-based approaches can also blend into each other. For example, a mediator-service is actually implementing part of its mediation function according to agreement settings and rules. The promotion and adoption of ***standards*** falls under the "agreement-based" approach and their value in achieving interoperability is well known. However, in the context of EOSC this class of solutions is affected by a number of issues, including:

*(i)* They can infringe the autonomy of the entities called on to adopt them;
*(ii)* There is no standard for every context, i.e. some, but not all possible standards exist;
*(iii)* Developing new standards aimed at "connecting" established "systems" often ends up being a complex combination of features reflecting the interests of the many parties involved.

When the discussion of EOSC System implementation starts, it is recommended that these two characterizations are investigated on a service-by-service basis rather than from the perspective of EOSC as a whole. Thus, the implementation and operation of every single EOSC Service should consider "federation" as a possible approach and discuss the "interoperability" solutions required to make it work. Some aspects/services seem to be already calling for federated solutions. For example, all the services where there is the willingness to offer a homogeneous view across several providers, such as AAI and Service registry.

## 3.3. Deployment considerations

Whilst the Architectural model highlights the typologies of services that EOSC should provide, it does not say anything about the underlying *Service Providers* that will operate and deliver these services, nor the topology of the EOSC structure and its allocations (e.g. global, regional, national). These aspects will progressively become clearer as the EOSC is built and, especially, when its sustainability model is clarified.

At this stage of EOSC development, we can state that services will be deployed on different ***EOSC Nodes***. EOSC Nodes are the settings where the elements of the EOSC system (either EOSC Services or EOSC Service Components) reside. The rules for the identification of nodes will vary, ranging from service specific constraints and peculiarities (e.g. some services are single-node by design) up to community-specific

deployment decisions (e.g. to have a national-based set of services each serving users based in a specific region).

The EOSC System is a truly distributed and heterogeneous "***hybrid cloud"***, i.e. it is an evolving mix of "*on-premises*" and "*off-premises*" IT Resources and organizations mobilized to deliver the envisaged services. "On-premises" means that an "EOSC Organization/Entity" (whatever form it will take) will be established, staffed, budgeted and provided with a certain amount of IT resources. This "entity" will operate - either by itself or by contracting this task to other organisations - a set of service instances (to be carefully identified) belonging to the identified classes. However, it is neither a desirable, feasible or sustainable scenario where this "Entity" will have the capacity to operate all and every instance of the service classes proposed. Rather, there will be a range of "Entities" (e.g. Research Infrastructures, Data Centres, Service Providers) helping the "EOSC Organization/Entity" to implement the overall EOSC by offering "off-premises" resources. It is also likely that the same, or similar, EOSC service is offered by multiple providers. For example, peer and federated services created for safety reasons.

At the current stage of EOSC development some patterns are emerging based on EU Commission funding opportunities, and Research Infrastructures and e-Infrastructure sustainability models. However, the situation is still unclear - not only due to the lack of an agreed understanding of what EOSC should indeed offer, and also because of the lack of well established rules of participation and governance model. It might be the case, for example, that the operation of the Services for EOSC System Managers and/or EOSC Suppliers and/or a large part of the Cross-cutting services comes under the responsibility of the "EOSC Organization/Entity". However, this solution does not necessarily imply that these services are all operated on "EOSC Organization/Entity" premises and they are all centralised. Services like, for example, 'Certification' or the 'Managers coordination' might indeed be built as a federation of similar services maintained by local or national institutions participating in the EOSC Entity. Similarly, services like the EOSC Portal could be commissioned or procured by organizations which have experience in developing such activities.

# 4. EOSC CLASSES OF SERVICES

This section presents in more detail the service classes previously introduced and summarized in Section 3. These are organized by individual user-role. In addition, there are a number of cross cutting service classes that are envisaged to support the activity of any EOSC System user, no matter the role they play.

It is important to recall that EOSC is meant to be built from the contribution of existing and future infrastructures, and of other third-party service providers that will enter in the EOSC System gradually on a voluntary basis. This will be based on the commitment of resources and on the capacity to comply with the established (yet evolving) Rules of Participation.

Data infrastructures would define the extent of their own involvement in the EOSC System, in terms of the data sets and services they would contribute to the EOSC. Their commitment and rule compliance would be limited to these data sets and services. [EC SWD 2018]

If not explicitly demanded, the distribution and diversity must be as transparent as possible to its users. The "cloud" must offer to users homogeneous virtual views of EOSC resources meeting their scientific needs, independent from how these resources can be accessed and where they are located. Also, as far as is possible, they must also be independent from the provider of the used resources. Thus, even if, for example, a researcher working in a particular area needs a given functionality that only a specialised service can offer, the access to that service must be as much as possible standardised. It is expected that the EOSC system, through its services, will offer complementary facilities to meet this goal.

EOSC can include several services of the same class, made available by different Service Providers. These could differ in the type of functionality offered, in certain quality parameters, in costs, and in other aspects including the user interface. This might be, for example, the case for a publishing support service or a service-hosting platform. It will be up to the user to decide which service to use among those available, according to their needs.

Each service can be self-standing or it can be built as federation of other services offering partial or complementary functionality. For example, it is envisaged that EOSC will have its own AAI service built as a federation of services operated by multiple identity providers. Similarly, major global services, like EOSC service and data registries, and cloud platforms will most likely be built as a federation of specialised services. Many of the services described in Appendix A are based on a federated service architecture.

Finally note that, as EOSC is intended to support an "as-a-Service" provisioning model, for each identified service (or class of service) there may be two diverse typologies of service: the ***factory*** and the ***instance***. The factory provides functionality for the creation and monitoring of the service instance (tailored to serve the client/user needs under a well defined SLA), while the instance is intended for the real consumption of the facility. A typical example of service class built upon factory services is the EOSC Resource Management (see Section 4.1).

## 4.1. EOSC Cross Cutting Services

The services of this class offer a set of functions supporting needs that are common to any user, independent from their role. In structuring the EOSC architecture, we collectively name the services providing these type of cross-role functions "***EOSC Cross Cutting Services***".

The function can be exactly the same for each role, e.g. Authentication, or it can correspond to different instances, each addressing the specific needs and view of the resources required by the user-role. For example, the same Service Registry containing service descriptions is expected to provide a specific list of services with specific descriptions when used by researchers and a different list of services and their accompanying descriptions when used by service developers.

A basic set of cross cutting service classes is detailed in Table 1.

**Table 1. EOSC Cross Cutting Services (alphabetical order)**

| |
|---|
| **AAI (Authentication and Authorization Infrastructure)** is a class of services that the EOSC System puts in place for authorization and access control. From a conceptual perspective, this is a centralised system, however, it must be implemented in a distributed way that guarantees that the existing solutions for authentication and authorization remain available and become interoperable.<br><br>Examples of existing services possibly falling in this class and/or exploitable to realize services in this class include eduGAIN, ELIXIR AAI, OAuth-based services, B2Access, INDIGO IAM and EGI Check-in. |
| **Accounting** is a class of service recording the consumption of EOSC Resources by users. From a conceptual perspective, this service collects and aggregates a number of statistics recorded by the services offering the resources. Moreover, diverse statistics about the same resource can be collected by different services. For instance, a dataset can be discovered by a Registry and downloaded by a Repository, thus the Registry should produce statistics on datasets discovery while the Repository should produce statistics on dataset downloads.<br><br>Finally, it is worth highlighting that the collected statistics are expected to be made available for consumption by others, including services. For instance, the statistics collected by the overall service can be used by the EOSC Governance (EOSC Portfolio Management) to take decisions on the decommissioning of services. By relying on consumption statistics, the registries can rank resources based on their usage or recommender services (see Section 4.2) can advertise resources with growing statistics.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include the PRACE Distributed Accounting Report Tool, the EGI Accounting Portal, the gCube accounting framework. |
| **EOSC Resource Management** is the class of services enacting EOSC Users to manage EOSC Resource instances (namely EOSC Services), i.e. to create/acquire new instances, to modify the capacity of a created instance, to decommission previously created instances. Services in this class are called to facilitate the dialogue between EOSC Service Providers and EOSC Users willing to make use of them.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include the EOSC-hub Marketplace (see Section A.3.2). |
| **EOSC Portal** is the class of services providing EOSC System users with a web-based entry point to EOSC Resources. It is expected that over time this class will have multiple instances. For example, in the EOSC framework there will be thematic portals and, most likely, also National Portals. In order to support cross-disciplinary activities there will also be one instance (or multiple instances operated by different providers), acting as the entry point to all EOSC resources. The exact functionality of this entry point is still to be defined, and, for example, it might offer direct access to the resources or it might redirect users to domain specific portals and/or services if needed.<br><br>The primary role of the portal is to be “the place” on the web where EOSC Resources, no matter their genre and provider, are expected to be “seamlessly” accessible. The implementation of this aspiration depends on several factors including the level of integration between the specific services and the portal. The level of integration of the various services into the portal GUI can vary a lot. However, the portal is conceptually called on to hide from the user - as much as possible/feasible - the issues resulting from the need to jump across several “sites” to complete a task.<br><br>The Resource Registry is among the EOSC Services that the portal should integrate. Through such integration the portal can offer a comprehensive view of the overall offerings of the EOSC System. The portal should also offer at least: |

*(i)* Services for logging into the EOSC System and to start using it (actually its Resources) seamlessly across the boundaries of the providers (*AAI and its single sign-on*);
*(ii)* Services enabling users to meet, discuss, share, collaborate, review and assess scientific outcomes and services, and perform activities together (e.g. *Social Networking, Community Building*, and *Virtual Research Environments* see Section 4.2);
*(iii)* Services enabling users to easily pass data across services (e.g. *Workspace*, see Section 4.2) and easily implement workflows combining services and data in user defined processes (e.g. *Workflows Management*, see Section 4.2);
*(iv)* Services getting support on EOSC System (e.g. *Service desk*);
*(v)* Services to get information about the operational state of EOSC (e.g. *Status dashboard*);
*(vi)* Services for managing the EOSC Resources instances (e.g. EOSC Resource Management).

In addition, the portal might also give relevant information on EOSC, e.g. rules and conditions, development plans, governing bodies.

It is of paramount importance that the portal provides users with a "personal" perspective of the EOSC System built by taking into account the user role(s) as well as the particular characteristics collected by the *User Profile* service (see below). This customisation is expected to be both (a) *user defined*, e.g. the user might decide how to organise the portal layout, what resources put in a bookmark for future references, and (b) *system assisted*, e.g. the system provides an up to date list of last/most used Resources and Services, suggestions for Resources of interest (e.g. *Recommender*, see Section 4.2).

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include the EOSC Portal as well as any science gateway, e.g. D4Science.

**PID Handler** is a class of services to generate and resolve Persistent Identifiers (PIDs). PIDs are envisaged to play a primary role in the EOSC. According to the Commission [EC2018b], all "data and other digital objects created by and used for research need to be findable, accessible, interoperable and reusable (FAIR)" and the very first principles of FAIR [Wilkinson et al. 2016] states that "(meta)data are assigned a globally unique and persistent identifier".

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include B2Handle, ePIC, ORCID.

**Resource Registry** is the class of services providing EOSC System Users with a list of live / ready-to-use descriptions of EOSC Resource offered by the EOSC System. The consumption of the list of resources is facilitated by search and browse operations. Every entry in a Registry must give relevant information (i.e. metadata) on the listed resources to enable its informed discovery and use, e.g. a description of resource intent and scope, a description on how to access it, a description of the licence governing its usage. Registries are the primary component called on to support EOSC Resources FAIRness.

Many EOSC Resource Registry instances may exist, each tailored to serve the needs of a designated community. Every Registry instance is characterised by its "coverage", i.e. the extent of the list the registry manages and its "view", i.e. the set of resource properties that are of interest to the designated community. Note that the two characteristics are independent, i.e. EOSC might operate different registries for the same set of resources, where each registry offers a different view of the same resources. At least one overall EOSC Resource Registry is likely to be operated to list the entire EOSC offering.

Two sub-classes of Registry are worth highlighting, the ***Data Registry*** and the ***Service Registry.*** The EOSC System should operate registries for services and data. However, the need for registries for a variety of other EOSC elements like researchers, stakeholders, policies, metadata schemas, ontologies, vocabularies and category definitions, is already envisaged.

Registries are expected to work in tandem with other services. They should liaise with the *Accounting* service to both produce statistics on operations performed by the Registry and to consume statistics from the items in their list, so providing users with more information on them. They should liaise with the *User Profiling* service to customise their behaviour according to user characteristics.

A detailed discussion of this class of services in given in Section A.1 by showcasing concrete instances.

**Service Desk** is a class of services aiming at providing EOSC System Users with help and support about the EOSC System as a whole. It includes a ticketing system for creating "tickets" documenting issues that have occurred, or other forms of requests, and to support and track their management, e.g. resolutions of errors, planning of features.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include the support pages offered by many infrastructures, e.g. OpenAIRE Support, EGI helpdesk, D4Science Support.

**SLA Management** is a class of services enabling, where appropriate, the definition and recording of the agreements that characterise the provision of Services. For some services there is only one agreement, for other services there can be an agreement per-user. It may be the case that some well-established externally provided services are made available through the EOSC with no specific agreement on service level. In this case default ones, in line with EOSC Rules of Participation, will have to be adopted.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include EGI SLA and OLA Framework.

**Status dashboard** is a class of services providing their users with an easy to understand view of the "health" status of the EOSC System. Services of this class are likely to display a matrix of services, dates and their corresponding status in an easy to read calendar view. The set of Services reported per-instance depends on the list of EOSC Resources for which users require timely status information. It caters for EOSC System Managers to easily create and track incidents, or schedule system maintenance, and provides a historical reference for performance calculations.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include Google's App Status, Amazon's Service Health Dashboard, Open Science Framework Status, ARGO Availability and Reliability Monitoring.

**Usage conditions compatibility checking** is a class of services providing users with facilities to help to assess the suitability of a resource of interest for a given exploitation scenario. In particular, these services should assess the suitability from the perspective of the usage-conditions characterising the resource. Such conditions are typically codified by licences, yet there might be usage conditions beyond a standard licence. The check usually falls under the umbrella of "legal interoperability" and should be formulated for each typology of the EOSC Resource, from service to dataset and software. In essence, it requires that every EOSC Resource is released and made available with a clear and accessible usage licence. The usage conditions should be readily determinable for each of the EOSC Resources, typically through automated means; they should allow for creation and use of combined or derivative products; and users should be able to legally access and use each resource without seeking authorisation from the resource holders. The licence or waiver assigned should be well-defined and internationally recognised to ensure that the conditions on access and reuse are comparable across jurisdictions.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include semantic representation of policies and automatic reasoning, e.g. [Daga et al. 2017].

| |
|---|
| **User profiling** is a class of services developing a "user profile", i.e. data capturing specific information about the user. The development of the profile should be as automatic as possible, i.e. the EOSC System should transparently establish characterising features and preferences out of user activity, though the user should have the ability to review and customise the data collected. |
| **Web-based APIs** is a class of services catering for programmatic access and exploitation of EOSC Resources, namely EOSC Services. These APIs are expected to follow the REST model.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include the wealth of APIs service providers have already started offering for accessing their facilities, e.g. OpenStack, D4Science dev, and tools for documenting and using these APIs, e.g. swagger. |
| **Mediators** are set of services developed to support intermediation among other services, where services no longer communicate directly with each other, and instead communicate through a mediator. Mediators in this context are used to disguise the heterogeneity of service interfaces, by replacing them with a common one. For example, hiding the heterogeneity of metadata in different formats by exposing them through a pivot format. Even if they are developed to satisfy needs in specific contexts, it will be useful to share them to avoid duplicating efforts in developing interoperability. For example, when they intermediate among two different standards. This class is almost open-ended. |

## 4.2. Services for Researchers

The classes of services included in this category implement an environment for facilitating cross-domain research, by enabling access to services for the *creation, deposition, processing, validation, interlinking and enrichment of research artefacts to enable reuse*, which equates to the FAIR management of research artefacts. The services must also enable *collaboration across domains, organizations, and along the entire scientific workflows*. These requirements can be linked, for example, by recommending the latest research results that meet an individual interest is a way to facilitate sharing and also a way to facilitate collaboration.

These services understand as primary entities "research artefacts" and "individuals making science". They offer functionality according to the semantics of these entities and the requirements related to their exploitation and management.

**Table 2. EOSC Services for Researchers (alphabetical order)**

| |
|---|
| **Citation, Attribution, and Reward** is a class of services for promoting and supporting actions aimed at maximising the citation (e.g. automatically keeping track and producing citation statements whenever an artefact is used in the context of a new artefact), attribution (e.g. automatically recording the specific contribution every single researcher is given in the context of the production of a given artefact) and reward (e.g. collecting indicators on the impact and contribution per-researcher). These services contribute to ensuring that the data required to assess the performance of researchers are collected and available. In the near future, a next generation of research assessment metrics is expected to emerge. This change may well require the introduction of other types of services to automatically records new parameters.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include repository facilities for citation (e.g. Mendeley) and metrics collected on deposited artefacts, services collecting bibliometrics, and altmetrics (e.g. Web of Science, Data Citation Index, Google Scholar, Altmetrics). |
| **Community Building** is a class of services providing their users with facilities for searching for contacts, organising lists of contacts, and automatically receiving suggestions for contacts of potential interest (e.g. |

because of similar interests or activities). From a conceptual perspective, this is a sort of Registry for contacts, with the facility to build personal views on top of it.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include LinkedIn, ResearchGate, and services automatically building co-authors map.

**Containers-as-a-Service** is a class of services providing their users with facilities for managing and deploying containers, applications and clusters through container-based virtualisation. In practice, this class goes beyond container-based virtualisation, to also include service providers offering virtual machines and mechanisms for managing them. It deals with the network connectivity and computing capacity.

Researchers should be provided with such a typology of service since it starts to spread the approach based on containers to the sharing of research artefacts, so enabling their "computational reproducibility". For other uses, e.g. using containers and virtual machines to deploy specific technologies, this document envisages that others perform such tasks. For example, third-party service providers and EOSC Service Providers are provided with a comprehensive hosting platform.

The acquisition of "computing", "storage" and "network connectivity" capacity for researchers is also hidden behind services such as those belonging to the Workbench or Repository classes (see below), as well as other services, without exposing researchers to the challenges resulting from the acquisition and management of such typologies of resources directly, so avoiding to distract researchers from scientific tasks.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include commercial providers (e.g. Google, Amazon Web Services, Rackspace and IBM), containers technology (e.g. Docker), orchestration platforms (e.g. Google Kubernetes, Docker Swarm, and Apache Mesos).

**Data Management Planning** is a class of services providing their users with facilities for defining and implementing data management strategies. Such data management strategies might be regulated / driven by funder requirements, so these services should ease the development of strategies to respond to funder requirements.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include tools for creating and sharing data management plans that meet institutional and funder requirements (e.g. DMPonline), and repositories enacting data management strategies.

**Data protection and security** is a class of services supporting researchers in sharing their research data according to policies and practices regarding privacy (e.g. GDPR) and, generally speaking, the safe sharing of sensitive and confidential data. These services enact to release data without violating the confidentiality of (personal) information while preserving useful information. Overall, these services aim to prevent issues related to unintentional information disclosure in research data releases.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include data anonymization tools (e.g. Amnesia), and data encryption tools.

**Data Taming** is a class of services supporting researchers in the development of suitable datasets for their research. This consists in identifying datasets of potential interest, extracting from the dataset items of interest, assess their fitness for purpose, and transforming the data originating from heterogeneous sources into an aggregated set conforming to a format suitable for the analysis to be undertaken. This class

of service is expected to follow the Search & Browse, as well as to work in tandem with several services including the:

(i) Workspace for organising some of the datasets to be collated, curated, and overall managed;
(ii) Workflow management for organising and executing data acquisition and transformation tasks by combining several services;
(iii) Workbench for assembling new research products out of the aggregated datasets and cater for their (re-)use;
(iv) Publishing support service for facilitating the inclusion of aggregated datasets into scholarly communications artefacts and workflows.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include OpenRefine, 'Extract, Transform, Load' solutions and (semantic) data linking solutions (e.g. Silk).

**Licence helper** is a class of services supporting the identification of suitable licences accompanying the release of any research product. They should clearly highlight the pros and cons of existing licences. These tools are expected to contribute to the implementation of data management planning services.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include wizard based approaches (e.g. choosealicense).

**Peer Review** is a class of services supporting the wealth and evolving array of peer review practices [Tennant et al. 2017]. Even though peer review is well established within scholarly communication, it is now evolving because of, for example, new research products needing to be assessed such as methods or datasets. In addition, peer review practices are diverse across communities.

This implies that the class will contain diverse instances of services covering the spectrum of peer review practices from "pre-peer review commenting" (i.e. where informal comments and discussions on a publicly available pre-publication manuscript draft are shared) up to "decoupled post-publication" (i.e. where comments or highlights are added directly to highlighted sections of the work) [Tennant et al. 2017]. In the EOSC settings, approaches realising open peer review should be endorsed and promoted, to become the norm across disciplines.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include traditional peer review platforms, GitHub-style [Tennant et al. 2017], and blockchain based approaches.

**Publishing support services** is a class of services aiming at reconciling the research activity performed by researchers "when doing research" with the activity they undertake "when communicating their research". In practice, these services will act as a bridge between researcher working environments and scholarly communication environments (often managed by Publishers). However, the scholarly communication environment underlying "open science" is expected to be more diverse with more practices and platforms for scholarly communication [EC2019]. These services are expected to work in tandem with Citation, Attribution and Reward services (see above), and Repository and Workbench services (see below).

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include editing platforms, e.g. Overleaf, Arpha Writing Tool, and web based publishing platforms, e.g. dokieli, OAI-ORE.

**Recommender** is a class of services envisaged to provide Researchers with objects of potential interest by predicting their preferences. This class shall include at least a cross-cutting recommender service aiming

to suggest suitable artefacts existing in the whole artefacts space. These artefacts include not only data products but also services, workflows, and people.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include basic frameworks supporting the development and testing of solutions, e.g. Apache Mahout collaborative filtering, recommenderlab and TensorFlow.

**Repository** is a class of services dealing with the long-term storage and availability of deposited artefacts. Specific service instances exist depending on the typologies of artefacts they support; for example, research paper vs data vs software vs mixed repository, and/or pdf paper repository vs "beyond paper" repository; the coverage, for example domain specific vs generalist; the preservation practices, and other features making the services diverse.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include generalist repositories (e.g. Zenodo, b2share, figshare, Dryad), domain-specific repositories (e.g. EMBL-EBI data repository), software repositories (e.g. CPAN, CRAN, git/github).

**Search & Browse** is a class of services to enable Researchers to find research artefacts (e.g. datasets, papers, methods, workflows) matching their interests. It is expected that, over time, these artefacts will become progressively more interlinked, providing a knowledge graph in which each artefact will be linked to its context. This class of services will include, at a minimum, a service dedicated to support the needs of a Researcher willing to seamlessly search over the entire assets space made available via EOSC, and to navigate across the knowledge graph. The search will make the effective source of data transparent and it will mediate across metadata, rights, usage licences and costs. It is also envisaged that in the near future, by exploiting new IT solutions developed in the information retrieval and semantics sectors, more and more data discovery will be automatically supported taking into account automatically generated user profiles.

Examples of existing services belonging to this class are the Google-like service, EUDAT B2Find, and the OpenAIRE Search.

**Social Networking** is a class of services supporting social networking interactions in a research context, e.g. the usage of posts to announce the availability of a research artefact, the usage of 'like' or comments to give positive or negative feedback to published artefacts, etc. These services are expected to work in tandem with others, e.g. they can be used for implementing new peer-review approaches, or they can be used to realise collaborative (crowd-based) approaches in research problems.

Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include well known and generic social networks (e.g. LinkedIn, Twitter, reddit), question and answer services (e.g. Quora), research oriented networks (e.g. ResearchGate, Mendeley).

**VRE Management** is a class of services supporting the creation and management of web accessible working environments dedicated to designated communities. Such environments are usually known as Virtual Research Environments, Virtual Laboratories and Science Gateways [Candela et al. 2013]. In this class of services fall all the end-user oriented services used to easily configure and deploy science gateways, virtual laboratories and virtual research environments. All of these services are oriented to serve the needs of a community willing to have a working environment for a specific investigation spanning organizational boundaries. Such environments should facilitate these communities in accessing the resources they need, that are often scattered across several providers, by a single and organised access point. Since this requirement is quite diffuse and it is expensive to implement solutions from scratch, a set of solutions and technologies have been designed and developed.

| |
|---|
| A detailed discussion of this class of services in given in Section A.2 by showcasing concrete instances. |
| **Workbench** is a class of services to enable Researchers to build new research artefacts out of their research activities. The workbenches Researchers are looking for range from environments enabling the collaborative editing of papers, algorithms, programs, notebooks, protocols, etc. up to advanced working environments enabling the (re-)use and combining of artefacts of any genre.<br><br>Examples of existing services belonging to this class are the collaborative editing environments (e.g. ShareLaTeX, Arpha Writing Tool), the code development environments (e.g. RStudio, Eclipse), analytics methods development and sharing (e.g. gCube Data Analytics [Assante et al. 2019]), and annotation environments (e.g. Hypothesis). |
| **Workflow Management** is a class of services to enable Researchers to define and execute scientific workflows, i.e. a series of data manipulation and computation steps. [Liew et al. 2016] have recently analysed selected Workflow Management Systems (WFMSs) that are widely used by the scientific community, specifically: Airavata, Kepler, KNIME, Meandre, Pegasus, Taverna, and Swift. Such systems have been analysed with respect to a framework aiming at capturing the major characteristics of WFMSs:<br><br>(i) *processing elements*, i.e., the building blocks of workflows envisaged to be either web services or executable programs;<br>(ii) *coordination method*, i.e., the mechanism controlling the execution of the workflow elements envisaged to be either orchestration or choreography;<br>(iii) *workflow representation*, i.e., the specification of a workflow that can meet two goals: human representation and / or computer communication;<br>(iv) *data processing model*, i.e., the mechanism through which the processing elements process the data that can be bulk data or stream data;<br>(v) *optimisation stage*, i.e., when optimization of the workflow (if any) is expected to take place that can either be build time or runtime.<br><br>The EOSC System will offer a rich array of WFMSs to accommodate all needs and use cases.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include WFMSs (e.g. Galaxy, KNIME, Taverna). |
| **Workspace** is a class of services to enable Researchers to conveniently store their own artefacts. Access to this storage space can be granted to others, so enabling the sharing of artefacts.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include Google Drive, file hosting services (e.g. owncloud, nextcloud), gCube workspace [Assante et al. 2019]. |

## 4.3. Services for Research Administrators

The classes of services included in this section are oriented to provide *Research Administrators* with dedicated facilities to track research activities of interest to their organisation, including research projects funded by the organisation. Research Administrators might be called to collect the results of these projects in order to produce metrics and indicators to assess the past (impact) and shape the future (trends). These classes of services complement the cross-cutting EOSC Services (see Section 4.1), and Research Administrators are expected to make use of the EOSC portal and the Registry for the identification of EOSC resources of interest.

**Table 3. EOSC Services for Research Administrators (alphabetical order)**

| |
|---|
| **FAIRness Monitor** is a class of services to assess FAIRness related policies. The implementation of the Open Science approach will largely depend on the extent that an effective FAIR data management will be put in place. Member States, funders, and in general research administrators, need information about FAIRness trends and landscape maturity to track the effect of their investments in FAIR management and to take corrective actions. EOSC is expected to provide services facilitating the gathering of such information and the production of appropriate reporting and statistics.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include those stemming from the just started FAIRsFAIR project. |
| **Open Science Monitor** is a set of services for supporting Research Performing Organisations (RPOs), Research Funding Organisations (RFOs) and Government Bodies in measuring:<br>• levels of compliance with European Union laws, regulations and policies regarding research and research results dissemination;<br>• levels of openness, trustworthiness and FAIRness that cover each stage of the research lifecycle for Open Science Resources (i.e. research artefacts, educational resources, research collaboration, citizen science).<br>• excellence of science, which includes quantitative and qualitative metrics of different levels (bibliometrics, webometrics, scientometrics, etc.);<br>• impact of science on society and economy.<br><br>Examples of existing services belonging to this class are Open Science Monitor, OpenAIRE Monitor, JISC Monitor, consisting of 'Monitor UK' which focuses in reporting APC's and 'Monitor local' for institutional compliance, and OpenAIRE Gold for FP7 activities as well as the EOSC Open Science Monitor envisaged by EOSCpilot [Papastefanatos et al. 2018]. |
| **Open Science Policy Registry** is a space where policies relating to aspects of open science can be submitted, assessed, and stored. The Registry provides combined facilities to store machine-readable policies, evaluate compliance with EOSC principles of engagement, and produce metrics and reports in support of the OS Monitor. It supports the EOSC externally and internally: as a service for users to assess EOSC compliance and compatibility, and as a reporting tool to produce metrics on the progress of open science in Europe.<br><br>Examples of existing services possibly falling in this class and/or exploitable to realize services in this class include the JISC SHERPA/ROMEO services for journals and funders policies as well as the EOSCPilot Policy Registry (see Section A.1.3). |
| **Policy Toolkit** is a collection of services, part of a larger set of instruments that include also resources, tools, and approaches, supporting the management of various policy issues. For example, Intellectual Property Rights (IPR) and exploitation, Technical provisions regarding machine-readability and interoperability issues. Policies will be interpreted within the scope and/or implementation procedures of different funders, publishers, libraries and other research and community stakeholders. The policy toolkit offers a comprehensive collection of functions that can be used to establish the machine-readability of policies.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include the EOSC Open Science Policy Toolkit [Mack & Papadopoulou 2018b]. |

## 4.4. Services for Third-party Service Providers

The classes of services included in this section are oriented to support EOSC Users willing to play the role of Third-party Service Providers. In particular, the classes of services envisaged should simplify the tasks of users who are willing to make use of EOSC Resources in order to deploy and operate their own services aimed at serving the needs of a specific community. These classes of services complement the cross-cutting EOSC Services (see Section 4.1). For example, Third-party service providers are expected to make use of the EOSC portal as well as of the AAI solution for exploiting EOSC Resources, and use the Registry to identify suitable resources and the EOSC Resource Management to acquire the resources (and capacity) needed to operate the target service.

**Table 4. EOSC Services for Third-party Service Providers (alphabetical order)**

| |
|---|
| **EOSC Development Platform (including APIs)** is a class of services implementing a platform simplifying the development of new services by leveraging EOSC Resources (namely EOSC Services). This platform should be (a) open and extensible to easily add new facilities made available whenever a new EOSC Service starts being provided and (b) cater for diverse exploitation models, e.g. to make it possible for users to use only the subset of facilities they need. These services should add value on top of the web-based APIs, for example, by offering "abstractions" on top of the APIs.<br><br>There are no real examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, as EOSC is still in early stages of development. However, similar efforts can be observed in other cloud providers that are making available web-based APIs and frameworks, to facilitate the development of services in compliance with the providers' settings. |
| **EOSC Service Monitoring** is a set of services to enable a Service Provider (Third-party Service Provider, EOSC Service Provider) to monitor the EOSC Services required to implement their own service. These services enable a dashboard that Service Providers can configure to control the compliance of the target EOSC Services with respect to the agreed SLAs, as well as any other constraint that it is useful to monitor for the health of the service provided by the Service Provider.<br><br>This class is a specialization of the Status dashboard (see Section 4.1). The specialization depends on the fact that it is oriented to a technical audience, thus the parameters to be collected and monitored can be a bit more specific. For instance, the ARGO Availability and Reliability Monitoring is used to monitor the status of the ARGO infrastructure. |
| **Hosting Platform** is a class of services enabling the acquisition and management of computing resources for the deployment of service(s) developed by Service Providers (both Third-party and EOSC). In practice, this class includes what cloud service providers use under the umbrella of Infrastructure-as-a-Service and/or Software-as-a-Service, as well as Network-as-a-Service, when specific network transport connectivity is needed. By using services in this class, Service Providers are allowed to acquire and manage the computing capacity (e.g. virtual machines, containers), storage capacity, database, and other resources needed for the service(s) under development and/or deployment to operate.<br><br>This class is a specialization of the EOSC Resource Management (see Section 4.1).<br><br>Among the instances of this service class there is certainly the ***EOSC Hosting Platform***, i.e. the platform conceptually operated by EOSC where the resources come from EOSC Nodes. |
| **Multi-provider Coordination** is a set of services aiming to coordinate the set of EOSC Services (and their providers) exploited by any Service Provider (both Third-party and EOSC) in the context of a specific service. Services in this class are expected to work in collaboration with those in the SLA Management (see Section 4.1) for the establishment of the necessary agreements. |

## 4.5. Services for EOSC System Managers (including EOSC Service Providers)

There are three sub-roles for EOSC System Managers:

- *EOSC System Owner*, accountable for the establishment and maintenance of the EOSC System;
- *EOSC System Top Manager*, accountable for the overall operation of the EOSC System;
- *EOSC Service Provider*, accountable for the delivery of one or more EOSC Services.

These sub-roles are expected to work with the support of services enabling communication and cooperation among them.

These classes of services complement the cross-cutting EOSC Services (see Section 4.1), i.e. EOSC System Managers are expected to make use of the EOSC portal and are expected to get information on the overall status by the Status dashboard.

**Table 5. EOSC Services for System Managers (alphabetical order)**

| |
|---|
| **Managers Coordination Platform** is a set of services enabling a smooth and timely communication and collaboration among the EOSC Managers. Such services include a repository of the documents and any other material supporting the management of the overall system, various communication channels for sending messages to selected members and/or groups, ticketing systems to assign tasks and monitor their execution. |
| **Portfolio Management Dashboard** is a set of services enabling the various EOSC Managers to collaboratively develop and maintain the EOSC Portfolio. Such a Dashboard should provide the managers with a comprehensive picture of the "performance" of the currently offered services (e.g. use and users, and incidents) as well as of the candidate services (e.g. audience and costs). |
| **RoP Management Dashboard** is a set of services enabling the definition, development and monitoring of the EOSC Rules of Participation. Such services include a repository of the established rules (possibly produced for both human-consumption and machine-consumption), a compliance checker to assess the extent an entity (e.g. EOSC Service) is matching the established rules, and an automatic alert system informing the managers whenever an entity is no longer adhering to the established rules. |
| **EOSC Monitor** is a set of services enabling EOSC Managers to be informed on the "performance" of the current EOSC Services. This class of services is a specialization of the Status dashboard (see Section 4.1) and is aimed to provide the EOSC Management with detailed and informative indicators on the overall system behaviour, including per-Service indicators like usage trends and incidents. This class of services is characterised by two phases: data collection and data consumption.<br><br>Data collection is intended to gather indicators of performance per-Service. This can be achieved in "push mode", where the Service Provider publishes the required information into a shared repository, or in "pull mode", where the EOSC Monitor Service Provider acquires the required information from the single Service Providers. In both cases, the interaction between the EOSC Monitor Service Providers and the rest of EOSC Service Providers is expected to be a collaborative effort, where whatever the approach used for collecting the required information it is understood and implemented on both sides.<br><br>Data consumption is intended to provide EOSC Managers with the information they require to monitor the performance of the system. It is likely to be implemented through a user-friendly graphical user interface that should be informative when needed, i.e. users should see aggregated views and, when required, enable inspection of the detailed information underlying the aggregated views. |

In addition, EOSC Service Providers are provided with a number of services specifically conceived to facilitate their role (see Table 6), including development platforms, management dashboards, and certification helpers.

**Table 6. EOSC Services for Service Providers (alphabetical order)**

| |
|---|
| **Certification** is a class of services offered to EOSC Service Providers to provide a formal accreditation or certification for the service(s) they offer with respect to a range of certification scheme. These services are specialised by type of service being offered and may also be specialised according to disciplinary practices.<br><br>A subclass of these services is that for supporting FAIR data management certification of data repositories. This is to assess if the data deposit and access practices conform to clear rules and criteria so that data are FAIR compliant. However, the detailed form of these services is awaiting definition. These services will probably support a semi-automatic process. More on these aspects will soon emerge from projects like FAIRsFAIR.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include Data Seal of Approval. |
| **Enrichment** is a class of services offered to EOSC Service Providers to complement and enhance the local "knowledge" services have of the services they provide, their users, etc by building upon the "knowledge" held by other EOSC Services.<br><br>A typical case involves user-related information, where a range of services collect, use and rely on information about their users. It is expected that individual services will collect a minimal level of user information; however, through liaising with other services, the knowledge every service has on "its" users can be enriched and improved. Though, in this case, due consideration must be given to GDPR and other information privacy regulations.<br><br>Another example relates to linked items, where for example, two papers cite each other or a paper cites a dataset held in different repositories. Every repository has relatively rich metadata on its own holdings, yet it might benefit from the information held in other repositories to enhance the metadata it has on its items.<br><br>Examples of existing services possibly falling in this class, and/or exploitable to realize services in this class, include OpenAIRE Content Providers Dashboard and OpenAIRE Broker. |
| **EOSC Development Platform** (see Section 4.4). This class of services is in common with Third-party Service Providers. |
| **EOSC Service Monitoring** (see Section 4.4). This class of services is in common with Third-party Service Providers. |
| **Hosting Platform** (see Section 4.4). This class of services is in common with Third-party Service Providers. |
| **Multi-provider Coordination** (see Section 4.4). In fact, this class of services is in common with Third-party Service Providers. |
| **Multi-supplier Coordination** is a class of services enabling the development and management of a service delivery plan across a range of suppliers. From a conceptual point of view, it is similar to the Multi-provider Coordination class, however, it is dedicated to facilitating the dialogue between an EOSC Service Provider and the set of EOSC Suppliers supporting the delivery of the target EOSC Service. This class of services will work in |

| |
|---|
| tandem with the Underpinning Agreements Management one (see Section 4.6) to support the establishment of a suitable set of Underpinning Agreements. |
| **Service Management Dashboard** is a set of services to enable EOSC Service Providers to manage their service, e.g. by being informed on incidents and requests for changes. This class of services is a specialization of both the Status dashboard (see Section 4.1) and the EOSC Monitor (see Section 4.5) and it is dedicated to provide every EOSC Service Provider with detailed and informative indicators on their own services when operated in EOSC. This service is not expected to replace the per-services management mechanisms, however, it aims at guaranteeing that when the specific services are operated in EOSC there are common practices and approaches across the diversity of providers for the collection of specified indicators, or for the way incidents are reported and managed. |

## 4.6. Services for EOSC Suppliers

EOSC Suppliers are actors who can supply the EOSC System with the set of EOSC Service Component(s) needed to enable one or more EOSC Service to operate. The aim is to facilitate a Business-to-Business dialogue between owners of technologies and other components that can be used to develop and operate one or more EOSC Services. Such owners may be unwilling to adhere to the rules and procedures enabling them to transform their components into fully fledged EOSC Services, and preferring to be the suppliers to other actors willing to play the role of EOSC Service Provider. Under the umbrella of EOSC Suppliers there are:

- *EOSC Service Component Suppliers* - suppliers willing to provide EOSC Service Providers with their components by the as-a-Service delivery mode. It is their duty to operate the technology needed to offer the asset on their own premises;
- *Data Service Suppliers* - suppliers willing to provide EOSC Service Providers with their own data by the as-a-Service delivery mode. This implies that it is a duty of the supplier to maintain the data and the technology needed to offer them on their own premises;
- *EOSC Service Developers* - suppliers willing to provide software suitable for the development and operation of EOSC Services. In this case, the as-a-Service delivery model is not envisaged.

**Table 7. EOSC Services for EOSC Suppliers (alphabetical order)**

| |
|---|
| **EOSC Development Platform (including APIs)** (see Section 4.4). This class of services is in common with Third-party Service Providers and EOSC Service Providers. |
| **Service Component Promotion** is a set of services enabling Service Component owners to make their services known to EOSC Users. In practice, this is a sort of Registry specifically conceived to enable potential and actual EOSC Suppliers to describe and publish their artefacts, so making the EOSC Service Providers aware of them and their characteristics. |
| **Underpinning Agreements Management** is the class of services enabling the definition, storage, access, and monitoring of the Underpinning Agreement governing the exploitation of a specific EOSC Service Component in the context of an EOSC Service. |

# 5. CONCLUDING REMARKS

The EOSC IT system is a complex and evolving platform expected to start taking its shape in the next few years. Its development will both influence, and be influenced by, several economic and societal forces such as:

- The community uptake both from the consumption and production perspectives;
- The IT services development market;
- The sustainability decisions taken by funders;
- Changes in the research contextual environment (e.g. new metrics for the evaluation of researchers).

The forces driving the evolution of this platform are very similar to the ones described in [Wittenburg & Strawn, 2018] for complex infrastructures:

> "New technological inventions or concepts with high potential for new types of complex infrastructures driven by early visions and often disruptive approaches are followed by a phase of creolization, i.e. the landscape of possible choices is being explored by many actors. Increasingly more options of the new concepts are investigated allowing the experts to better understand the underlying rules, principles and limitations. *These developments are driven by competition, overcoming reverse salients and efficiency considerations*.
>
> "Some tracks in this complex landscape turn out to be promising candidates and get more attraction**;** still heterogeneity and fragmentation hamper economy of scale effects. Therefore, some experts exploring the landscape start looking for universals that can act as a bridge between the different remaining solutions and as a stable basis for new developments and massive exploitation. Attraction and convergence are driven mainly by efficiency and economic concerns. The benefit of convergence is the belief of stakeholders that a stable fundament has been built, *on top of which new investments and developments can be made* to fully exploit the new technologies and infrastructures.
>
> "*It is finally economics and efficiency that define the direction of system development*. This also raises questions about the success of a top-down process when building infrastructures."

It clearly emerges from this vision that:

*(i)* The point of view of the users is key since adoption, and consequently sustainability, will ultimately depend on how well EOSC will satisfy user needs;

*(ii)* The realization of a complex system like the EOSC IT cannot be linear. It necessarily requires a stepwise, iterative and agile approach able to appropriately respond to the change of requirements, conditions and regulations of the surrounding context.

These considerations have been foundational for the work performed in Task 5.1. In particular, the former has influenced the identification of "what" the EOSC system should provide while the latter have informed the "how" it should be built.

The starting points of our whole work have been:

*(i)* The identification of the typology of EOSC users (roles);

*(ii)* Their basic functional requirements (see D5.1 [Candela et al 2018]).

We have assumed that the required functionalities are provisioned *as-a-Service*, i.e. they are made available by online services operated by providers who take care of the technical and organisational approaches needed to deliver the planned functionalities. This has driven the identification of 47 classes of services that can be considered at this stage of development as the "Minimal Viable Product" able to match the EOSC overall goal. Such services include *cross-cutting services* together with services specifically envisaged to serve researchers, research administrators, third-party service providers as well as EOSC managers, service providers and service suppliers.

This report has described the identified classes of services with the aim of providing a view of the functionalities that the EOSC IT system should provide (Sections 3.1 & 4). The report has also discussed aspects related to how this system can/should be developed (Sections 3.2 & 3.3).

The EOSC system has been modelled as an open and evolving *System of Systems* (SoS) where the component systems providing services include existing and emerging Research Infrastructures (including e-Infrastructures) and other types of Service Providers hosted on distributed *Nodes*. This system is thus expected to be a highly distributed, evolving and heterogeneous *hybrid cloud*, i.e. it is an evolving mash-up of on-premises and off-premises IT resources and organizations mobilized to deliver the envisaged services, whose operation and development is regulated by a set of *Rules of Participation.*

The notion of "federation" and "interoperability" related to the building of this EOSC system have also been discussed, highlighting the importance of dealing with these two concepts per-single service rather than from the perspective of EOSC as a whole.

In the last part of the deliverable (Appendix A) the concrete architecture of a number of EOSC identified services has been addressed by describing how they are built - often as federation/composition of different components. The aim of this has been to provide suggestions to those that are currently developing these kinds of services to serve the overall EOSC users. For the sake of effort and time required, the number of services for which this architectural analysis has been performed is necessarily limited. We recommend that future initiatives that continue the work done in the EOSCpilot project produce similar surveys for other typologies of services, and that these surveys are made openly available so that every service developer can look at them before making their own choice.

We conclude this deliverable by noticing that despite the "Minimal Viable Product" assumption taken, if we look at the services that infrastructures can currently make available to EOSC we see that they do not yet cover the entire spectrum of minimal required functionalities. Aspects like community building and collaboration, recommendation, reward and attribution and many others related to the new open science vision are not yet on the radar of major e-Infrastructures or of the thematic Research Infrastructures.

This gap is expected to be progressively filled in the coming years. Moreover, it is already anticipated that the EOSC services will largely change to reflect the evolving rules, policies and practices that the implementation of the Open Science vision will put in place. As in any ecosystem, in order to obtain a good quality and sustainable system this change will have to be open to competition. Multiple providers will have the opportunity to offer their services and their resources and the "market" of researchers and other users will decide those that will survive and progress. These considerations delineate a rapidly evolving landscape to which e-Infrastructures and Research Infrastructures will be required to promptly respond. This will also open a new market and a variety of new opportunities to third-party service providers.

## REFERENCES


[AARC 2017] AARC Consortium. ***AARC Blueprint Architecture***. https://aarc-project.eu/wp-content/uploads/2017/04/AARC-BPA-2017.pdf

[Assante et al. 2019] Assante, M.; Candela, L.; Castelli, D.; Cirillo, R.; Coro, G.; Frosini, L.; Lelii, L.; Mangiacrapa, F.; Marioli, V.; Pagano, P.; Panichi, G.; Perciante, C.; Sinibaldi, F. (2019) ***The gCube system: Delivering Virtual Research Environments as-a-Service***, In: Future Generation Computer Systems, Vol. 95, pp 445-453, DOI: 10.1016/j.future.2018.10.035

[Barker et al. 2019] Barker, M., Silvia Delgado Olabarriaga, Nancy Wilkins-Diehr, Sandra Gesing, Daniel S. Katz, Shayan Shahand, Scott Henwood, Tristan Glatard, Keith Jeffery, Brian Corrie, Andrew Treloar, Helen Glaves, Lesley Wyborn, Neil P. Chue Hong, Alessandro Costa, A. (2019) ***The global impact of science gateways, virtual research environments and virtual laboratories.*** Future Generation Computer Systems, Volume 95, 240-248 DOI: 10.1016/j.future.2018.12.026

[Candela et al. 2013] Candela, L.; Castelli, D.; and Pagano, P., (2013) ***Virtual Research Environments: An Overview and a Research Agenda***. Data Science Journal, 12, pp.GRDI75–GRDI81. DOI: 10.2481/dsj.GRDI-013

[Candela et al. 2018] Candela, L.; Castelli, D.; La Rocca, G.; Lukkarinen, A.; Manghi, P.; Pagano, P.; Papadopoulou E. (2018) ***Initial EOSC Service Architecture***. EOSCpilot Deliverable D5.1 https://eoscpilot.eu/content/d51-initial-eosc-service-architecture

[Daga et al. 2017] Daga, E.; d'Aquin, M.; Motta, E. 2017. ***Propagating Data Policies: a User Study***. In Proceedings of the Knowledge Capture Conference (K-CAP 2017). ACM, New York, NY, USA, Article 3, 8 pages. DOI: 10.1145/3148011.3148022

[Demleitner et al. 2014] Demleitner, M.; Greene, G.; Le Sidaner, P.; Plante, R. L. (2014) ***The virtual observatory registry***. Astronomy and Computing, Vol. 7-8, DOI: 10.1016/j.ascom.2014.07.001

[Dowler et al. 2017] Dowler, Patrick; Demleitner, Markus; Taylor, Mark; Tody, Doug (2017) ***Data Access Layer Interface Version 1.1***. IVOA Recommendation, https://ui.adsabs.harvard.edu/#abs/2017ivoa.spec.0517D

[Duan et al 2015] Duan, Y.; Fu, G.; Zhou, N.; Sun, X. Narendra, N. C. and Hu, B. (2015) ***Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends***. IEEE 8th International Conference on Cloud Computing, New York City, NY, 2015, pp. 621-628. DOI: 10.1109/CLOUD.2015.88

[EC SWD 2018] European Commission (2018) ***Implementation Roadmap for the European Open Science Cloud***. Commission Staff Working Document, SWD(2018) 83. https://ec.europa.eu/research/openscience/pdf/swd_2018_83_f1_staff_working_paper_en.pdf

[EC2018a] European Commission (2018) ***Prompting an EOSC in practice: Final report and recommendations of the Commission 2nd High Level Expert Group on the European Open Science Cloud (EOSC).*** *EU Publications, Directorate General for Research and Innovation.* DOI: *10.2777/112658*

[EC2018b] European Commission (2018) ***Turning FAIR into reality: Final report and action plan from the European Commission expert group on FAIR data.*** *EU Publications, Directorate General for Research and Innovation.* DOI: *10.2777/1524*

[EC2019] European Commission (2019) ***Future of Scholarly Publishing and Scholarly Communication: Report of the Expert Group to the European Commission.*** *EU Publications, Directorate General for Research and Innovation.* DOI: *10.2777/836532*

[Ferreira et al 2019] Ferreira, N.; van Wezel, J.; Williams, M. (2019) ***EOSC Service Portfolio Roadmap***. EOSCpilot Deliverable D5.5 https://www.eoscpilot.eu/content/d55-eosc-service-portfolio-roadmap

[Jensen 2013] Jensen H.T., ***EMI StAR – Definition of a Storage Accounting Record,*** Open Grid Forum, 2013, https://www.ogf.org/documents/GFD.201.pdf

[Liew et al. 2016] Liew C.S., Atkinson M.P., Galea M. et al. (2016). ***Scientific Workflows: Moving Across Paradigms***. ACM Computing Surveys; 49(4). DOI: 10.1145/3012429

[Mack & Papadopoulou 2018a] Mack, L. & Papadopoulou, E. (2018) ***Open Science Policy Registry***. EOSCPilot Deliverable D3.4 https://eoscpilot.eu/content/d34-open-science-policy-registry

[Mack & Papadopoulou 2018b] Mack, L. & Papadopoulou, E. (2018) ***Open Science Policy Toolkit***. EOSCPilot Deliverable D3.5 https://eoscpilot.eu/content/d35-open-science-policy-toolkit

[Meghini 2017] Meghini, C (2017). VRE4EIC Deliverable D3.5 ***Final Architecture Design***. https://www.vre4eic.eu/images/Public_deliverables/D3.5_Final_Architecture_Design_PU.pdf

[Papastefanatos et al. 2018] Papastefanatos, G.; Papadopoulou, E.; Meimaris, M.; Mack, L.; Thorat, R. (2018) ***EOSC Open Science Monitor specifications***. EOSCpilot Deliverable D3.2 https://eoscpilot.eu/content/d32-eosc-open-science-monitor-specifications

[Shahand et al. 2015] S. Shahand, A. H. C. van Kampen, S. D. Olabarriaga (2015) ***Science gateway canvas: A business reference model for science gateways***, in: SCREAM '15 Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models. DOI: 10.1145/2753524.2753527.

[Tennant et al. 2017] Tennant JP, Dugan JM, Graziotin D et al. (2017) ***A multi-disciplinary perspective on emergent and future innovations in peer review*** [version 3; referees: 2 approved]. F1000Research 2017, 6:1151 DOI: 10.12688/f1000research.12037.3

[Wilkinson et al. 2016] Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak A. et al. (2016) ***The FAIR guiding principles for scientific data management and stewardship***. Scientific Data, 3:160018. DOI: 10.1038/sdata.2016.18

[Wittenburg & Strawn, 2018] Wittenburg, P.; Strawn, G. (2018) ***Common Patterns in Revolutionary Infrastructures and Data***. DOI: 10.23728/b2share.4e8ac36c0dd343da81fd9e83e72805a0

# APPENDIX A. SELECTED SERVICES DETAILED ARCHITECTURES

In Section 3 we introduced a model of the EOSC system architecture from the perspective of its different class of users and briefly discussed potential deployment models for it. In this Appendix we go a step further by discussing the issues and possible solutions for selected cross-cutting services. In particular we have chosen three large classes of cross-cutting services often referred in recent HLEG reports:

*(i)* *Registries* of various genres (see Section A.1);
*(ii)* Virtual Research Environments, Science Gateways, and Portals aiming at simplifying the consumption of "resources" from diverse perspectives (see Section A.2);
*(iii)* Services facilitating the acquisition of EOSC Resources (see Section A.3).

Registries have been selected for further analysis because of their key role in enabling the dissemination and awareness raising of available EOSC Resources. The services discussed below are instances of the *Resource Registry* (see Section 4.1).

Virtual Research Environments and Gateways have been selected for further analysis because of their role aiming to simplify the task of researchers in accessing and consuming available EOSC Resources for scientific purposes. The services discussed below are instances of the *VRE Management* (see Section 4.2).

Services facilitating the acquisition of EOSC Resources have been selected for further analysis in order to clarify the wide spectrum of resources EOSC users have to deal with and how this acquisition can be simplified. The services discussed below are instances of the *EOSC Resource Management* (see Section 4.1) or of some of its specializations, namely the *Hosting Platform* (see Section 4.5).

For these typologies of services, we present existing alternative architectures, many of them built as a federation of components interoperating through different community *de-facto* standards, that are implemented in the context of existing infrastructures and projects. In the spirit of leveraging not only the services, but also the knowledge acquired in developing them in the context of existing Research Infrastructures and e-infrastructures, the intent is to provide examples to those that are developing and will develop cross-cutting services with similar functionality.

## A.1. Registries

The request for FAIR data management, and more generally, the need to easily "find" items of interest is increasing the importance of developing and operating different typologies of Registries as EOSC certified enabling resources.

A registry contains descriptions (metadata) of resources and offers, at a minimum, functionalities to upload, search and retrieve these descriptions. In the EOSC System, Registries are indeed a large class of services, each of can assume different 'shapes' according to the design and implementation decisions for features such as:

- Typology of items described, e.g. services, datasets, actors, organizations, other registries;
- Uploading approach, e.g. harvesting, insertion through a user interface, programmatic uploading from files;
- Centralised vs distributed architecture;
- FAIRness and preservation strategies for the maintained descriptions.

The majority of existing registries have been developed to support their primary expected functionalities, i.e. upload, search and retrieve, but there are also cases of registries created as the foundation to provide much richer and complex functionality, e.g. see the EOSC-hub Marketplace see Section A.3.2) where the typical catalogue facilities are used to enact the acquisition of the discovered resources.

The first example discussed below is the International Virtual Observatory Alliance (IVOA) (see Section A.1.1). Its scientists have built a complex network of registries dedicated to many different types of resources. To

be able to keep track of all these different, often distributed, registries and to retrieve resources across them, IVOA has decided to build the 'Registry of Registries' described below. The underlying architecture might be an interesting model for EOSC, which certainly has to face a variety of heterogeneous, distributed and multi-resource types of resources.

The second example analysed is the Service Catalogue implemented by the eInfraCentral project (see Section A.1.2). It collects service descriptions in a common metadata description format through registration and harvesting methods, performs validation checks and aggregates those service descriptions into a uniform service catalogue that can be queried and browsed through a web portal. The architectural solution is quite generic and is applicable to many other types of similarly focussed registries.

The third typology of Registry discussed is an example of service dealing with a very different type of resources, i.e. policies (see Section A.1.3). The section presents the designed architecture. At the time of writing this report a complete implementation of that architecture is not yet available. We have decided to include it here because registries for policies will certainly increase their relevance in the evolving EOSC framework, not only as themselves, but especially as foundation for more high level assessment functionalities.

Note that in many cases the service name "registry" and "catalogue" are used as synonyms. We have chosen the former to name the class of service and reserved the term catalogue to indicate a list of items that can be disseminated by a registry. However, in the rest of this section we will use the two terms interchangeably giving a preference to the one used in the context of the service described.

### A.1.1: The IVOA Resource Registry (by M. Molinaro)

The International Virtual Observatory Alliance (IVOA) provides a registry collecting the various resources that data providers in the astrophysical communities make available through IVOA standard protocols and other means [Demleitner et al. 2014].

The definition of the distributed infrastructure that manage those resources, from the landing page of its main access point: the Registry of Registries, is:

"***The Registry of Registries*** *(RofR, pronounced rover) is a web portal provided on behalf of the International Virtual Observatory Alliance (IVOA) and overseen by the IVOA Registry Working Group. It is targeted to Virtual Observatory (VO) registry providers and VO application developers that wish to interact with registries.*

*The key service provide by the RofR is an* ***IVOA publishing registry*** *that lists all publishing registries known to the IVOA. When a resource metadata harvester harvests from these publishing registries, they can discover all published VO resources around the world. The design and recommend uses of the RofR is documented in the IVOA Note, The Registry of Registries*. ..."

The following ***key features*** characterise the overall solution:

- ***OAI-PMH interface based cooperation***: IVOA Registries expose their resources through an OAI-PMH interface that allows for cross-harvesting and global resource coherence as well as querying by clients outside the astrophysical domain;
- ***SQL-like RDB interface***: to allow for finer and domain specific queries on the available resources;
- ***XML resource descriptor documents***: extending from Dublin Core and mappable to DataCite metadata for an appropriate machine readable and interoperable representation of the managed resources.

#### *Design*

A detailed description of the design decisions is given in [Dowler et al. 2017] and summarized below.

According to IVOA, a registry is first a ***repository of structured descriptions of resources***. In the VO, a resource is defined by the IVOA Recommendation, henceforth referred to as RM, as being a general term referring to a VO element that can be described in terms of who curates or maintains it and which can be given a name

and a unique identifier. Just about **anything can be a resource**: it can be an abstract idea, such as sky coverage or an instrumental setup, or it can be fairly concrete, like an organization or a data collection. Organizations, data collections, and services can be considered classes of resources. The most important type of resource is "application", i.e. a service that actually does something. A registry (lower case), then, is "*a service for which the response is a structured description of resources*" (RM).

The VO environment features different types of registries that serve different functions. The primary distinction is between ***publishing registries*** and **searchable ones**:

- A searchable registry is one that allows users and client applications to search for resource records using selection criteria against the metadata contained in the records. The purpose of this type of registry is to aggregate descriptions of many resources. A searchable registry gathers its descriptions from across the network through a process called harvesting.
- A publishing registry is one that simply exposes its resource descriptions to the VO environment in a way that allows those descriptions to be harvested.

Note that some registries can play both roles; that is, a searchable registry may also publish its own resource descriptions.

A secondary distinction is ***full*** versus ***local***:

- A full registry is one that attempts to contain records of all resources known to the VO.
- A local registry, on the other hand, contains only a subset of known resources.

As mentioned above, harvesting is the mechanism by which a registry can collect resource records from other registries. It is used by full registries to aggregate resource records from publishing registries. It can also be used to synchronize two registries to ensure that they have the same contents. Harvesting is modelled as a pull operation between two registries. The term harvester refers to the registry that wishes to receive records (usually a full searchable registry); it sends its request to the harvestee (usually a publishing registry), which responds with the records. Harvesting is a much simpler process than a fully-featured search interface, as only very few constraints need to be supported and only full records are being transmitted in responses. Consequently, different protocols are employed for the two types of registry operations.

From this description it is clear the IVOA Resource Registry is a distributed ecosystem of registries, each having a database (not necessarily an RDB) to store XML documents describing the managed resources. Full registries provide direct entry points to a global catalogue of resources.

***Implementation***

The resources available through the astrophysical Virtual Observatory (VO) are **XML documents** that follow a set of standards defined by the IVOA (see "ReR" section at the IVOA Documents Repository). Those standards define the XSDs needed to validate the content.

Each **Resource has a unique identifier** (IVOID, itself following an IVOA standard) that is a profile over the URI syntax. IVOIDs are composed of an "authority" part and a subsequent "local" part. Being authorities unique by construct, so are IVOIDs in general since the local part must be locally unique.

Resources are accessible through the OAI-PMH interface or the Relational Registry Interface (IVOA standard) that maps the XML documents into a relational schema.

***Deployment***

As reported above, the IVOA Resource Registry, coordinated through a Working Group within the Alliance and having a starting point in the Registry of Registries, is **a distributed environment of publishing services, sharing common standards and discovery and access interfaces**.

It currently consists of a set of full registries that harvest the full set of publishing registries to present the user (community and client applications) with a globally unique set of resources (currently more than 20,000).

Each publishing registry, through its harvestable OAI-PMH interface, acts as a local resource deployer. The full registries both assure redundancy over the network and allow full repository discovery queries.

***Use cases***

Astrophysical applications (e.g. TOPCAT, Aladin, VO-Spec, SPLAT) are able to access data collections and services on top of them through the metadata descriptions available via the Registry.

Major worldwide astrophysical databases and services (e.g. Vizier, Simbad, NED, HEASARC) have their resources described in the IVOA Registries alongside smaller communities and efforts.

The IVOA Registry acts as repository for resources and services discovery for an astrophysical virtual research environment and has been running for several years, with ongoing maintenance and upgrades discussed at the IVOA level.

The resource content of the IVOA registry is also being made available through the EUDAT B2Find service, proving its usefulness and interoperability at a wider, cross-domain level. It has also been accessed by FAIRsharing.org for possible inclusion in its standards repository.

**A.1.2: The eInfraCentral Service Catalogue (by G. Papastefanatos)**

The eInfraCentral Service Catalogue aims to provide its end-users (researchers, service providers, research administrators) with a central entry point/gateway to a harmonised and aggregated service catalogue and an additional distribution channel to e-infrastructure services offered by each e-Infrastructure, or by other aggregators. It will provide a collaborative web portal to act as the main entry point to European e-Infrastructure services. To achieve this, the eInfraCentral Service Catalogue collects service descriptions and related data through registration and harvesting methods, and aggregates the services offered into a uniform service catalogue. The catalogue is made available via a catalogue portal employing formal/standardised guidelines and Application Programming Interfaces (APIs) to offer up-to-date information to end-users. The following key features characterise the Service Catalogue:

- ***Service Catalogue Browsing***. Provides the functionality for catalogue viewing, browsing and service comparison as well as for the retrieval of service-related resources. It allows users to retrieve the list of services contained in the catalogue, retrieve the metadata for a specific service, as well as KPIs and statistics aggregated for it, or compare two or more services along specific characteristics. It also enables users to perform keyword search over the catalogue and navigate its content through faceted filters.
- ***Service Registration***. Enables the service providers to add the description metadata for a new service in the catalogue, as well as perform any subsequent updates of those metadata. Service registration offers the creation, validation, update and removal of service providers and services. Service metadata follows the service model developed in eInfraCentral. It consists of five building blocks of information, namely: *Basic Service Information* (e.g., service id, name, provider, description, service url), *Service Classification Information* (e.g., category \ subcategory, TRL, lifecycle status), *Service Support Information* (e.g., Order url, training material, helpdesk), *Service Contractual Information* (e.g., Service price, SLA, terms of use) and *Service Performance Information* (e.g., service availability, usage).
- ***Service Monitoring & Statistics***. Allows Service Providers to manage and monitor their service uptake in the service catalogue portal through a dashboard.
- ***AAI and catalogue personalization features***. An AAI (Authentication, Authorization and Identity) service enables users to login via several identity providers (institutional and social logins). It also

enables authenticated users to access personalized features such as rating a service, adding a service into their favourite list and subscribing to alerts and notifications.

- ***APIs for remote synchronization***. Enables the exchange of data from the service catalogue with external systems. It offers APIs for integrating service information from external systems of the service providers and providing the content of the platform (service catalogue and metadata) to third parties.

***Design***

The eInfraCentral Service Catalogue service architecture is depicted in Figure 5.

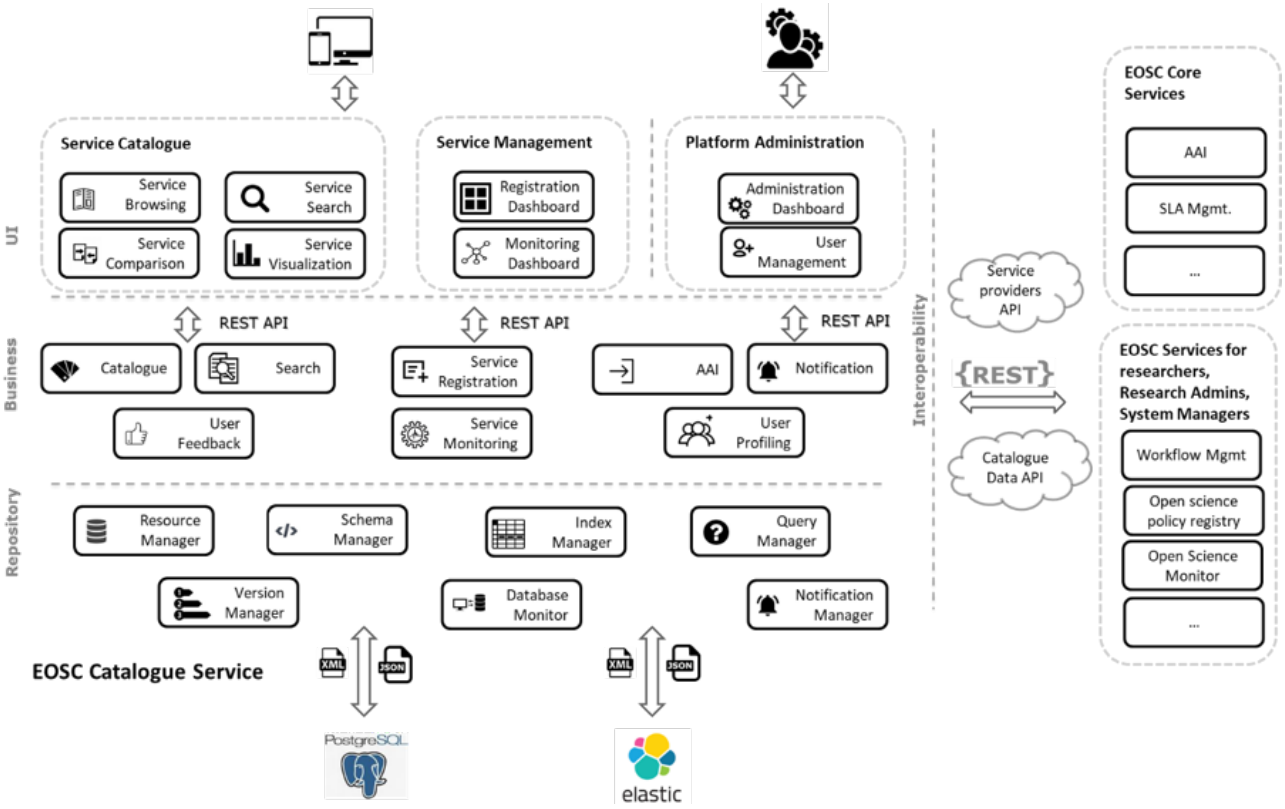


**Figure 5. The eInfraCentral Service Catalogue architecture**

The main components are:

- The ***Repository Layer*** is responsible for all underlying content management of the service. This layer comprises the following components:
    - **Schema Manager:** it is responsible for registering resource types, validating the schema provided, and preparing the database to accept new resources of the new types.
    - **Resource Manager:** it manages and validates resources against the schema and the constraints (referential, primary keys and unique values), and register them in the database.
    - **Query Manager:** it performs queries to the index (based on the index fields defined in the resource type) and returns the results. Whilst the resource manager is using the database to register the resources, the queries are performed at the index, which allows for far more efficient queries.
    - **Database Monitor:** it monitors the database for any changes in registered resource types or resources and notifies other modules that are interested in these changes.
    - **Version Manager:** it manages the different versions of the resources. Whenever a resource is updated, the Version Manager stores the previous version and assigns a new version

number to the modified resource. The **Index Manager** is then responsible for updating the full text index of resources.

- **Notification Manager:** it provides the connection between the Repository and the Notification service in the Business Layer of the platform. The Notification Manager allows for any other service in the platform to be asynchronously notified about the changes in the Repository without explicitly interacting with the Repository.

- The ***Business Layer*** implements and offers the core functionality of the catalogue service. It comprises of a set of modules that communicate with the UI Layer and the Interoperability Layer via REST APIs:
  - **Catalogue Module:** it stores and manages all service-related resources.
  - **Search Module:** it offers all keyword search and faceted browsing functionality. Facets are created automatically on top of specific attributes of the service catalogue, such as the service categories and the service providers and allow for easy filtering of platform content.
  - **User Feedback Module:** it is responsible for collecting, storing, and aggregating all user-feedback.
  - **Service Registration Module:** it manages all the registration process for new services and the update of existing services.
  - **Service Monitoring Module:** it manages the update of service metadata via the synchronization with remote service providers' systems.
  - **AAI Module:** it is responsible for interacting with the central EOSC AAI facility managing user registration, logins and authentication.
  - **Notification Module:** it manages message propagation and notification within the front end.
  - **User profiling Module:** it offers to authenticated users of the platform personalized features, such as favourite services.
- The ***User Interface (UI)*** layer contains the front-end UI modules, through which users interact with the platform. Based on the role of each user, different groups of UI modules are accessible. The main modules are:
  - **Service Catalogue UI modules** offer functionality to the public end users for accessing and browsing the full eInfraCentral catalogue. In addition, authenticated users can rate a service, subscribe to and receive notifications regarding a service or a category update, create collections of favourite services in their profile, etc.
  - **Service Management UI modules** offer functionality to the users of the service providers to register and manage their offerings in the platform, i.e., register an organization as a service provider or themselves as delegates of a service provider, register a new service in the platform, view the list of services corresponding to their organization and edit/update details or make active/inactive certain services.
  - **Platform Administration UI modules** enable administrators of the Catalogue service to configure various parameters and content within the service, such as the management of enumerated fields, the construction of FAQs and help pages as well as the management of users, organization and roles.
- The ***Interoperability Layer*** enables the communication and exchange of information between the catalogue service and the rest EOSC services, either being core EOSC services or services targeting researchers, research administrators or EOSC system managers. It offers a list of REST APIs for integrating service information from external systems of the service providers and providing content (service catalogue and metadata) to 3rd party services.

***Implementation***

The Repository Layer is developed using Java 8, the underlying relational database is PostgreSQL and the full text index used is ElasticSearch. The implementation of the modules in the Business Layer aims at offering a loosely coupled and modular architecture that can be adjustable by adding new features and services, and at the same time be scalable to large number of end users. All modules are implemented with the Spring Framework in Java 8, exposing and exchanging data via REST APIs with the components of the other layers. The UI's functionality is implemented in HTML5, CSS and JavaScript using Angular2 and UIKit. The UI follows an HTML responsive web design for enabling seamless browsing via web browsers and mobile devices.

***Deployment***

All modules of the catalogue service can be deployed and executed as dockerised software modules. Docker is a software technology providing containers, an additional layer of abstraction and automation of operating-system-level virtualization on Windows and Linux. There are several deployment options for the catalogue service.

1. **Single Deployment** at service providers' machines for operating their own service registry. A service provider can host the catalogue service at its own infrastructure and populate the catalogue with services offered by its organization.
2. **Aggregator Deployment**, where a service catalogue can be deployed by an aggregator organization (e.g., EGI, GEANT, OpenAIRE, PRACE, EUDAT, thematic aggregators, etc.), which aggregates and manages service descriptions offered by several different service providers.
3. **eInfraCentral *Catalogue as a Service***, where catalogue service is offered for both of the above options, in the form of a SaaS. A service provider or an aggregator can create and monitor custom-based cataloguing services in the EOSC system.

In all cases, exchange and synchronization of information are possible via the REST APIs offered by the catalogue service.

***Use cases***

The catalogue service has been successfully used for operating the eInfraCentral catalogue of e-Infrastructure services, which will be made available as EOSC service. eInfraCentral's mission is to ensure that, by 2020 a broader and more varied set of users (including industry) discovers and accesses the existing and developing e-infrastructure capacity. A common approach to defining and monitoring e-infrastructures services will increase the uptake of and enhance understanding of where improvements can be made in delivering e-infrastructure services.

Another use case is that the catalogue service has been deployed and used in OpenAIRE e-Infrastructure for managing and cataloguing all of its OpenAIRE Services.

### A.1.3: The EOSCpilot Policy Registry (by L. Mack)

The EOSCpilot Open Science Policy Registry is a system for EOSC service providers and policy stakeholders to submit and register policies as well as to validate their EOSC policy alignment [Mack & Papadopoulou 2018a]. It is a central service to help ensure that those who operate in the EOSC align themselves with the EOSC's policy requirements. The Policy Registry's core purpose is to support and monitor the adoption, as well as to validate the compliance of policies that external stakeholders and service providers implement in order to align themselves with the EOSC's policy requirements (as defined by the EOSCpilot Policy Recommendations and Rules of Participation). Together with the Open Science Monitor [Papastefanatos et al. 2018] and Open Science Policy Toolkit [Mack & Papadopoulou 2018b], it has been designed to complement the policy-supporting services of EOSCpilot. However, it is highly configurable and adaptable to serve other contexts and evolving typologies of policies.

The following key features characterise the service:

- A uniform, extensible policy metadata model that captures, structures, and represents policy information in relation to multiple use cases and policy types in the Registry's backend database.
- A modularised, interoperable architecture connecting to different EOSC services and external components.
- Decomposable, transparent policy evaluation workflows to determine the level of compliance or alignment.

The Policy Registry goes beyond metadata, i.e. it also implements a directory where policies would be recorded. The Registry's functions include:

1. **Submission and registering of policy metadata**: Users of the Policy Registry can submit and register metadata about their own organisational (e.g. institutional privacy policy) or contractual policies (e.g. terms and conditions). The required metadata entities are selected in relation to different EOSC policies (i.e. Rules of Participation and Policy Recommendations) and use cases.
2. **Assessing and validating policy compliance**: Once a user has submitted policy metadata, the supplied data is automatically assessed for compliance with the Policy Registry's metadata requirements. The exact metadata requirements depend on the Registry's two use cases. These can be either service providers seeking an assessment of their compliance with the Rules of Participation (use case 1) or stakeholders seeking a broader assessment of their alignment with EOSC Policy Recommendations (use case 2) (see subsection "Use cases" for more information).
3. **Storing / archiving of submitted policy metadata:** Upon submission, policy metadata, including assessments on policy compliance and alignment, are stored in the Policy Registry's database backend, serving as the EOSC's repository to preserve metadata on service providers' and stakeholders' policies;
4. **Provision of data:** Metadata stored in the Policy Registry's backend database would be accessible to secondary users via open APIs, e.g. to provide the Open Science Monitor, EOSC governance bodies, and external users with micro-data on policy implementation.

***Design***

Figure 6 depicts the service architecture of the EOSCpilot Open Science Policy Registry:

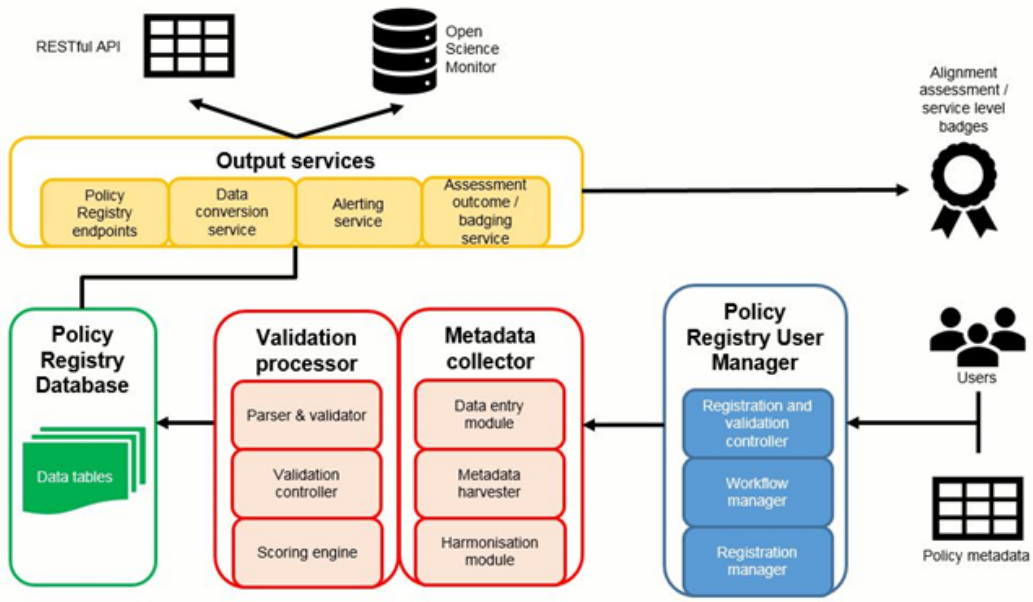


**Figure 6. The EOSCpilot Open Science Policy Registry service architecture**

The Policy Registry is composed of five modules that support the generic workflow of the Policy Registry.

The ***Policy Registry User Manager*** provides the user-oriented front-end module of the Policy Registry and supports users in the policy registration / submission processes. It is based on three components:

- **Registration manager:** it supports the user and policy registration process;
- **Workflow manager:** it supports one or multiple users in managing the overall workflow of the policy registration / submission and validation process (e.g. initiate, amend, repeat stop, and delete a policy registration process);
- **Registration and validation controller:** it provides users with feedback on the (automatic) assessment of submitted policy metadata, e.g. to control whether mandatory metadata is complete and provided in the correct schema. The registration and validation controller integrates with the validation controller in the backend.

The ***Metadata Collector*** module facilitates data collection. It consists of:

- **Data entry module:** for the submission of policy metadata through a dynamic metadata form;
- **Metadata harvester**: providing an automated mechanism (e.g. API queries) to retrieve relevant data from applicable service catalogue endpoints;
- **Harmonisation module:** ensuring that all programmatically ingested data are harmonised according to the Policy Registry's data model.

The ***Validation Processor*** module supports the data validation and assessment process. Its components include:

- **Parser and validator:** to parse submitted policy metadata, assess/confirm the machine-readability of data, and validate the compliance with metadata requirements;
- **Validation controller:** it allows administrators (e.g. EOSC Technical Committees) of the Policy Registry to control the validation process and amend the assessment outcomes of automated validation processes;
- **Scoring engine:** it conducts the aggregation and computing of data to assign an overall assessment outcome.

The ***Policy Registry Database*** stores all policy metadata in versioned data tables.

The ***Output Services*** is the module to supply policy metadata to relevant users and other EOSC services, ensuring that this data can be accessed in a programmatic manner. It consists of:

- **Assessment outcome and badging service:** providing a summary of the assessment outcome to the submitting users (for this the component integrates) with the Policy Registry User Manager;
- **Alerting service:** it alerts users on the completion of assessments;
- **Policy Registry endpoints:** it exposes a RESTful API to make the Policy Registry Database accessible to other EOSC services (e.g. Open Science Monitor) and external users of the Policy Registry database;
- **Data conversion service:** it ensures that data is accessible as JSON, CSV and RDF/XML.

***Implementation***

The Open Science Policy Registry is planned to be implemented as a simple web-service, using XML or JSON. Depending on the implementation of other EOSC service components, a PHP-based implementation to facilitate data exchanges could be an option, too. To ensure adaptable machine-to-machine communications between different Registry service modules, the Registry should rely on a series of RESTful APIs, potentially building on the OpenAPI specification in order to increase interoperability with external data users.

The Policy Registry's database requires a flexible implementation which can be adapted easily if metadata or data evaluation requirements change. A relational or NoSQL-based modelling of the database could facilitate this purpose.

With regards to the Registry's metadata model, it is noteworthy that no universal standards for policy representations exist. Given the specific focus of the EOSCpilot Policy Registry on the EOSC's policy requirements, a custom metadata model would be needed, designed along the lines of the proposal made in the D3.5 [Mack & Papadopoulou 2018b]. In the absence of universally agreed metadata standards, the metadata interoperability with other data models internal and external to the EOSC will however be crucial (e.g. data schemas of eInfraCentral and the EOSCpilot Open Science Monitor).

***Deployment***

The Open Science Policy Registry should be deployed as a single-instance, multi-tenancy architecture. The Policy Registry's focus is primarily on the efficient, simplified management of a validation mechanism and directory for the policies of EOSC service providers and stakeholders; there is no need to customise its functions for individual users. In short, this means that a multi-tenancy implementation appears as a more suitable, and cost-effective, choice than a multi-instance implementation.

Generally, the different modules of the Registry could be operated by different service providers or across different sites. However, given the relative simplicity of the service itself as well as the limited amount of static data which it produces and consumes, a distributed deployment of different modules appears only justifiable if this led to substantive cost savings or efficiencies in data sharing with other applications that reuse the Registry's data.

***Use cases***

The Policy Registry is currently focussed on serving the need of two main application use cases serving actors with different roles. These use cases are summarised in Table 8. Their common purpose is to enable users to validate how well their own policies align with the EOSC's policy requirements. In the case of the Service Providers (Use case 1), this means a policy compliance assessment with the Rules of Participation in order to ensure that services are classed as EOSC compatible and can be marketed in the EOSC. In the case of other stakeholders (Use case 2), this means a policy alignment assessment with the Policy Recommendations. This assessment on alignment allows stakeholders to compare and rate their progress on policy implementation according to the EOSC's own Policy Recommendations.

**Table 8. Policy Registry application use cases**

| | **Use case 1 – Service providers: Validation of service policies** | **Use case 2 – EOSC policy stakeholders: Registration of stakeholder policies** |
|---|---|---|
| **Use case summary** | Validation of service policy compliance with EOSC's Rules of Participation for third parties who want to provide their services in the EOSC | Registration of information on Open Science-related policies in order to monitor alignment with EOSC's Policy Recommendations |
| **Main users** | Third-party service suppliers (i.e. providers who want their services to be recognised as "EOSC compatible") | Stakeholders which use EOSC services and/or have an interest in being recognised as Open Science policy adopters |
| **What is assessed?** | Contractual policies (terms and conditions of service providers) and, if applicable, organisational policies (e.g. privacy policy of a service) | Organisational policies (e.g. institutional privacy or Open Science policies) |

| | | |
|---|---|---|
| **Who benefits?** | Service suppliers: get certified for EOSC service status level ("EOSC compatible")<br><br>EOSC: collect detailed data on compliance with RoPs, allowing the management of the EOSC service catalogue | Stakeholders (= RPOs, RIs, funders): assess implementation of policies relating to Open Science<br><br>EOSC: receives data on OS policy implementation by relevant stakeholders, facilitating development of EOSC policy framework and governance |

#### A.1.4: Considerations on Registries

The EOSC Registry service class is expected to be among the widest classes of services that are likely to exist in EOSC for several reasons, including:

- The "basic" nature of this kind of service for every community / domain, i.e. to be informed on what is available communities tend to create registry-like services;
- The lack of consolidated and cross-community standards and formats for describing resources of interest;
- The proliferation of domain specific approaches aiming at capturing the peculiarities of a community or a domain when describing resources of interest;
- The trade-offs between the willingness to provide the users with generalist discovery mechanisms and very specific discovery mechanisms.

Every registry instance represents a point in a multidimensional space having the above characteristics as axes. This suggests that it is almost impossible to envisage a "one-size-fits-all" solution when going to implement registries expected to serve EOSC. However, registries in EOSC should be envisaged with the following characteristics:

- Cater for low "*barriers to entry*" for both those willing to populate the Registry and those willing to access the Registry content. Whenever possible avoid asking for new descriptions of already published items and try to maximise the reuse of information already published;
- Pursue the "*fitness for purpose*" strategy, i.e. registries should make evident the scope and intent they are aiming for. Building overarching catalogues is of interest, yet there is always a trade-off between the "precision" of the discovery mechanism that can be supported and the "recall" of the results, the larger and more heterogeneous a catalogue is, the greater is the possibility to mix "apple and pears".

## A.2. VRE Services, Science Gateways, Portals

The EOSC Portal is envisaged as the entry point to the EOSC system for all the different actors willing to exploit it. It will support retrieval and access to the large variety of services and resources that will be made progressively available by EOSC Providers. The number of these services is expected to grow continuously and soon become very high. The variety and richness of services accessible through the portal can mean new capacity for researchers and other EOSC users. However, too many alternatives, the majority of which are of little or no relevance for the specific objectives of the users, can result in 'noisy' working environments that distract and confuse the researcher, so that they may end-up preferring to use their own 'familiar' desktop environments.

The class of services explored in this section respond to this issue by realising custom views of the large EOSC resource space, and by offering to their users through a Web Portal a tailored working environments focussed on specific application needs. The three services discussed implement different understandings and solutions. In the literature these classes of solutions are often named with terms including Virtual Research Environments (VREs), Virtual Laboratories and Science Gateways [Candela et al. 2013, Barker et al. 2019].

Section A.2.1 presents the solution developed by the SeaDataCloud project that makes available to its users a number of tailored applications all accessible through the same Portal. The applications rely on a layer of generic services common to many applications. This reduces the cost of expanding the set of applications that can be progressively built to meet specific user needs.

Section A.2.2 illustrates the solution proposed by the VRE4EIC project. The solution in some way resembles the previous one, in the sense that the focus is on the identification of a layer of enabling services that are intended as a common core for specific application case. The presence of these enabling services facilitates the development of application environments tailored to the needs of specific users, either single or collective. The solution presented by the VRE4EIC project is intended to be generic and applicable to any domain where digital science based on data analysis plays a key role. The generic services have been prototyped and are available at http://v4e-lab.isti.cnr.it:8080.

Section A.2.3 presents a completely different approach implemented in the context of the D4Science Infrastructure. D4Science operates a service, named VRE Manager, that acts as a sort of factory for VREs. Through exploiting these services tailored environments can be built dynamically and on demand to respond to personalised needs. Users, can thus, if they want, have access only to their own VREs.

Finally, Section A.2.4 concludes the discussion on these typologies of services by giving some remarks.

### A.2.1: The SeaDataCloud VRE (by D. Schaap)

SeaDataCloud is a project for further development of the pan-European SeaDataNet infrastructure for marine and ocean data management. As part of its project activities SeaDataCloud (SDC) is developing advanced downstream services for users to take more advantage of all the available data resources and to make the users' analytical processes easier. These services exploit the SeaDataNet Common Data Index (CDI) data discovery and access service (currently being upgraded by adopting cloud technology) by which users can search, order and download data aggregated by the large and comprehensive SeaDataNet infrastructure. The SDC Virtual Research Environment (VRE) is one of these services. It facilitates collaborative and individual research. In the VRE, users can use, analyse, process and visualise ocean and marine data meeting their specific needs. The products, as result of the analysis, can be integrated, visualised and published. For this purpose, a cloud environment has been deployed to host a number of advanced services that make use of subsets of the available SeaDataNet data resources. Data retrieved and downloaded from external data sources and/or users own data sets can be deposited as personal data pools (My Data) and can then be used in the advanced services of the VRE. The VRE is being set up in such a way that over time additional advanced services can be included without too much effort.

**Design**

The SDC VRE as a cloud processing environment is being deployed on the EUDAT platform that offers the following default services:

- The B2SAFE service offers the functionality to store and replicate datasets across different data centres in a safe and efficient way while maintaining all information required to easily find and query information about the replica locations. The CDI service related data cache are stored in B2SAFE.
- B2STAGE is the EUDAT service designed to transfer data to/from the EUDAT storage resources. It provides an HTTP-API interface, which enables users to programmatically access EUDAT services.
- B2HOST is the generic EUDAT service that allows communities to deploy and operate their own applications, or data-oriented services on EUDAT computing resources.
- B2DROP is a user-friendly and trustworthy storage environment which allows users to synchronize their active data across different devices. In addition, it provides researchers with a common service for exchanging active research data within a small group of researchers and with fine-grained access control mechanisms.

- B2ACCESS is the EUDAT federated cross-infrastructure authorisation and authentication framework for user identification and community-defined access control enforcement. It enables users to access EUDAT services with different authentication methods.

The SeaDataCloud Virtual Research Environment (VRE) needs a number of basic components to fulfil the functional and non-functional requirements as extracted from the use cases. When looking from a high-level perspective the components can be divided into generic layers with service components and dedicated VRE components (see Figure 7):

- Processing service layer
- Security/Authentication layer
- Data access layer
- Integration layer / Graphical User Interface (GUI)

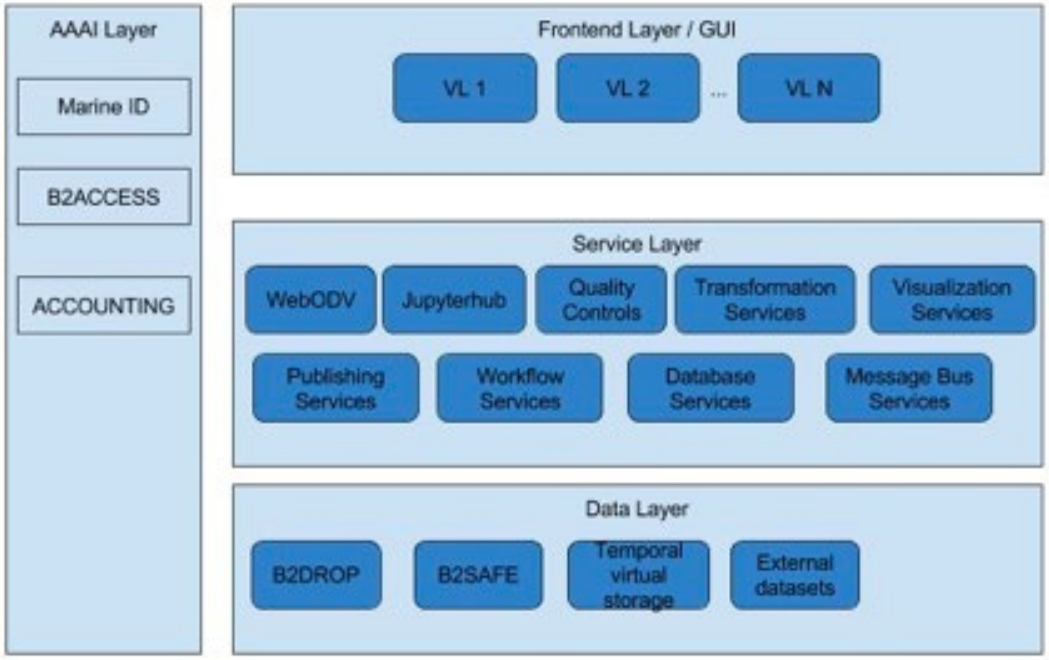


**Figure 7. SeaDataCloud VRE: Diagram of layers in the architecture**

Per layer, the SDC VRE contains the following main components:

**Service layer**

- SeaDataNet ODV web editor (https://webodv.awi.de);
- Notebook services like SeaDataNet DIVA interpolation service: Via Jupyterhub;
- Subsetting service via ERDDAP;
- Quality assessment, e.g. via SeaDataNet Octopus software, SeaDataNet Biology quality assessment tool;
- Transformation services, e.g. via SeaDataNet Octopus software;
- Server side visualisation, following OGC standards;
- User and resource use accounting.

**Security/Authentication layer**

- The functionality of this layer is provided by the B2ACCESS service, which is responsible for user identification, authentication and management of group and authorisation information. B2ACCESS is based on Unity IdM and is used in production within the EUDAT infrastructure since 2015. SeaDataNet uses Marine-ID for its registered users. Marine-ID users will be mapped onto B2ACCESS,

meaning that users at first time login will also be registered within B2ACCESS, after which they will be able to access data, services and tools in the VRE using their federated identity without the need for multiple login credentials;

- User administration: Roles and group management.

**Data access layer**

- User data workspace;
- CDI service data access components;
- Integration of external data sets via GeoDab (GEOSS) broker.

**Integration layer / Graphical User Interface (GUI)**

- Access to virtual labs related to use cases;
- Specific data viewers;
- Sharing and publishing;
- Creating workflows;
- Communication within group.

***Implementation and deployment***

To accommodate the very diverse applications that already exist, to easily integrate new ones, and to avoid dependency conflicts, all the applications run inside Docker containers. This approach also improves environment portability, scalability, security and scientific reproducibility. Application containers run on the EUDAT's computing service B2HOST, which provides scalable processing capability, fast access to storage volumes, and scheduling of containers execution. Every service exposes its own interface, which could be a GUI, a REST API or a command line interface in a Jupyter Notebook.

The application containers (i.e. services) interface with the data layer in the backend, with the user interfaces in the frontend, and with the VRE controller. The frontend is being developed as a responsive JavaScript layer, based on state-of-the-art JavaScript libraries. It is responsible for authentication and authorization of the users (Marine-ID on top of B2Access), and for routing them to the interfaces of the services. Services that are common to all processing services can also be accessed from here, such as dataset management component, process chaining (workflows), user communication, and version notification.

The main data backend of the VRE is the EUDAT service B2DROP. B2DROP allows users to upload, store and securely share datasets. It is accessible to all the applications by the WebDAV protocol, allowing the applications to interact with the B2DROP content as a file system.

Behind the frontend, a controller makes up the heart of the VRE and glues all the components together. It sees to the mounting of the user's private data from EUDAT's nextcloud-based B2DROP service, mounting other data and volumes, launches the correct service containers on-demand and it is responsible for all inter-container communications.

The first version of the VRE is being released for internal users from the SDC data product groups. A later release will contain more functions and will be accessible for a wider audience, although within the limitations of the current SeaDataCloud project.

***Use cases***

The SDC VRE is targeting five main use cases, two of which will be carried out in close collaboration with EMODnet Chemistry and EMODnet Bathymetry. Each use case concerns a specific community which brings together resources (data collections, processing, tools) in a Virtual Lab for a targeted group of users and applications:

- *"SeaDataNet Temperature & Salinity water column analysis":*

- It supports the development, update and publication of data products for European sea regions consisting of regional climatologies for temperature and salinity observations in the water column.

- "*EMODnet Chemistry analysis*"
  - It concerns the development of data products similar to Temperature and Salinity but now for Biogeochemical parameters (ammonium, chlorophyll-a, dissolved oxygen, phosphate and silicate concentration).
- "*SeaDataNet Biology quality assessment*"
  - It offers a Virtual Lab that processes ODV files in the biological data format (BIO-DEF). The user can select quality control procedures regarding the data format, taxonomic quality, completeness and validity, geographic quality and outlier analysis.
- "*EMODnet High Resolution Seabed Mapping*"
  - It facilitates the data provision from data providers and the collaborative processing of regional DTMs between Regional Coordinators and their interaction with the Integrator using the software components (like Globe software) as in use in EMODnet HRSM.
- "*Processing and Visualizing data*"
  - Demonstrate the capabilities of the VRE to more "generic" users, not part of a certain community. It aims to demonstrate what the user can do with the data collections / datasets provided by SeaDataCloud and quickly aggregate, process and view the datasets downloaded via SDC data access services.

#### A.2.2: The VRE4EIC VRE model (by C. Meghini)

VRE4EIC develops a ***Reference Architecture*** and ***software components*** for VREs – see [Meghini 2017]. This e-VRE bridges across existing e-RIs (e-Research Infrastructures) such as EPOS and ENVRIplus, both represented in the project, themselves supported by e-Is (e-Infrastructures) such as GEANT, EUDAT, PRACE, EGI & OpenAIRE. The e-VRE provides a homogeneous interface for users by virtualizing access to the heterogeneous datasets, software services, resources of the e-RIs and also provides collaboration/communication facilities for users to improve research communication. Finally, it provides access to research management /administrative facilities so that the end-user has a complete research environment.

***Design***

For the basic infrastructure of e-VRE, the VRE4EIC model implements a set of basic functionalities grouped into six ***conceptual components***:

- The e-VRE management is implemented in the **System Manager** component. The System Manager can be viewed as the component enabling Users to use the *core* functionalities of the e-VRE: access, create and manage resource descriptions, query the e-VRE information space, configure the e-VRE, plug and deploy new tools in the e-VRE and more.
- The **Workflow Manager** enables users to create, execute and store business processes and scientific workflows.
- The **Linked Data (LD) Manager** is the component that uses the LOD (Linked Open Data) paradigm, based on the RDF (Resource Description Framework) data model, to publish the e-VRE information space - i.e. the metadata concerning the e-VRE and the e-RIs in a form suitable for end-user browsing in a SM (Semantic Web)-enabled ecosystem.
- The **Metadata Manager (MM)** is the component responsible for storing and managing resource catalogues, user profiles, provenance information, preservation metadata used by all the

components, using extended entity-relational conceptual and object-relational logical representation for efficiency.

- The **Interoperability Manager** provides functionalities to implement interactions with e-RIs resources in a transparent way. It can be viewed as the interface of e-VRE towards e-RIs. It implements services and algorithms to enable e-VRE to: communicate synchronously or asynchronously with e-RIs resources, query the e-RIs catalogues and storages, map the data models.
- The **Authentication, Authorization, Accounting Infrastructure (AAAI**) component is responsible for managing the security issues of the e-VRE system. It provides user authentication for the VRE and connected e-RIs, authorisation and accounting services, and data encryption layers for components that are accessible over potentially insecure networks.

***Implementation and deployment***

A Technical Architecture has been defined in VRE4EIC in order to implement a specific eVRE called ***Canonical Reference Prototype*** **(CRP)**, which complements the Reference Architecture by selecting a set of suitable implementation techniques and open-source components.

The key point in the derivation of the technical architecture has been the VRE4EIC non-functional requirements:

- The developed Virtual Research Environments must be a dynamic system: it should reuse and integrate existing VRE tools, services, standardized building blocks and workflows where appropriate and develop new innovative ones where needed;
- The eVRE should be applicable to different multidisciplinary domains, i.e. it can be potentially used in every research domain;
- The eVRE functionalities should be exposed as services in a standardized way to enable developers to easily use them to develop new applications;
- The eVRE must provide innovative standard software services to be retro-fitted to existing VREs to enhance them for their own domain purposes and for interoperability.

From the architectural point of view the above requirements mean that the eVRE system must be expandable (by adding or replacing software components), modular (every architectural component should be independently deployable) and capable of supporting technology heterogeneity. VRE4EIC adopted the Microservices approach for this technical architecture, as the two key concepts of Microservices architecture fits the above requirements:

- Loose coupling: every service knows as little as it needs to about the components with which it cooperates; this enables the microservices to be independently deployable on existing VREs or replaceable in different domains;
- High cohesion: components with related behavior stand together (i.e., related logic is kept in one service); changing the technology used to implement a microservice does not affect other microservices.

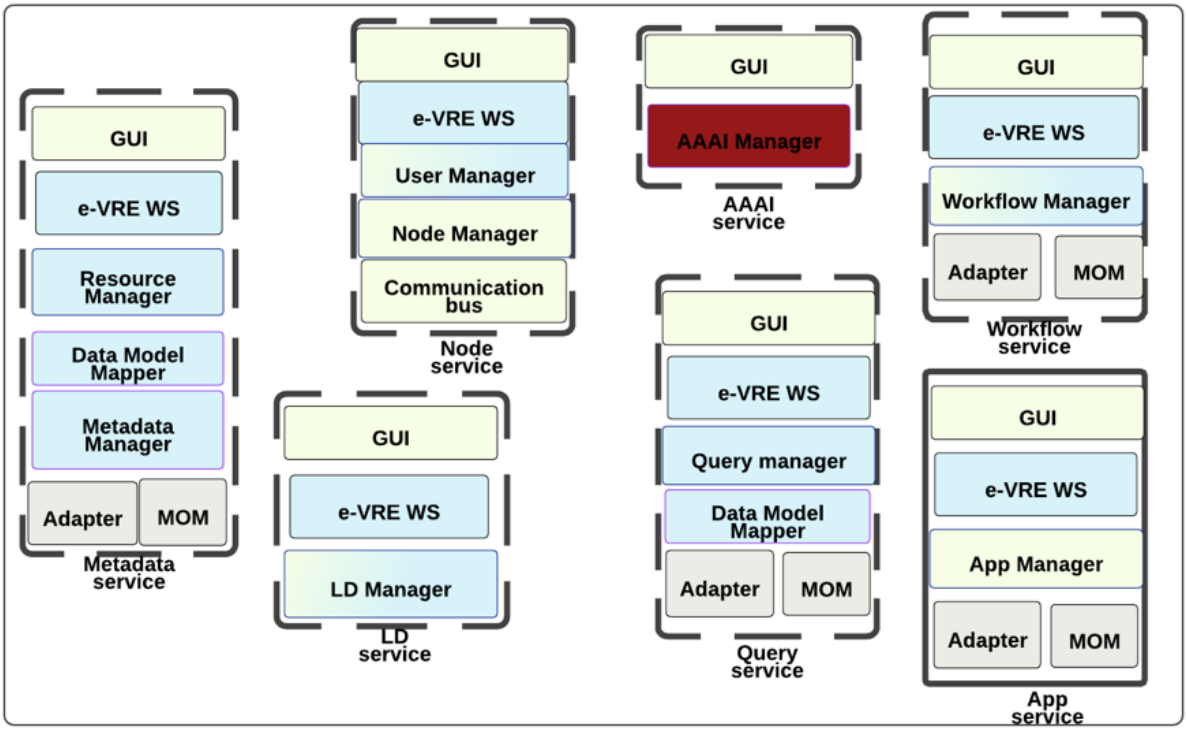


**Figure 8. VRE4EIC Technical Architecture**

The set of conceptual components functionalities, defined in the reference architecture, have been partitioned according to the non-functional prerequisites, and the resulting subsets have been implemented as *microservices*.

The e-VRE source code is available on GitHub. It can be downloaded and re-used according to the Apache V2 licence.

The Canonical Reference Prototype (CRP), used by demonstrators and Metadata portals, is available at http://v4e-lab.isti.cnr.it:8080.

***Use cases***

The reference architecture developed by VRE4EIC has been demonstrated in the form of a canonical reference prototype whose components have been used in EPOS and ENVRIPlus:

- EPOS, the European Plate Observing System, is a long-term plan to facilitate integrated use of data, data products, and facilities from distributed research infrastructures for solid Earth science in Europe. The collaboration and interaction between EPOS and VRE4EIC was carried out along two main dimensions: EPOS integration and EPOS enhancements. In the first one, EPOS contributed to make its assets (metadata, datasets, etc.) available in an integrated way through the VRE4EIC system prototype. In the second one, EPOS took advantage of existing building blocks from VRE4EIC that implemented missing functionalities in EPOS.
- The ENVRI community represents a cluster of environmental and earth science research infrastructures. The Data for Science theme within the ENVRIplus project is concerned with providing common technical solutions and recommendations to many of the problems shared by the ENVRI community, for example with regard to metadata cataloguing, provenance, identification and citation of persistent resources and data processing. Thus, the e-VRE architecture and building blocks solutions have both been exploited to the ENVRIplus community as part of the ENVRI service portfolio of technologies, standards and recommendations. More specifically, e-VRE developments

have been applied to the problems of enhancing i) cross-RI data and service discovery, ii) cross-RI workflow composition, and iii) cross infrastructure workflow execution and provenance.

More information about VRE4EIC software and demonstrators can be found at: https://www.vre4eic.eu/evre/software.

### A.2.3: The D4Science VRE Manager (by L. Candela)

The D4Science VRE Manager is a service that enables the definition, deployment and operation of *Virtual Research Environments on demand* on D4Science infrastructure premises [Assante et al. 2019]. D4Science-based Virtual Research Environments (VREs) are ***web-based***, ***community-oriented***, ***collaborative***, ***user-friendly***, ***open-science-enabler*** working environments for scientists and practitioners willing to work together to undertake a certain (research) task. From the end-user perspective, each VRE manifests as a web application comprising several components and running in a plain web browser. Every component is aimed at providing VRE users with facilities implemented through relying on one or more services provisioned by diverse providers. In fact, every VRE is conceived to play the role of a ***gateway*** giving seamless access to the ***datasets*** and ***services*** of interest for the designated community and their tasks, whilst hiding the diversity originating from the multiplicity of resource providers.

The following ***key features*** characterise the service:

- ***wizard-based VRE characterisation***: the service offers a wizard-based mechanism enabling authorized users (aka VRE designers) to easily select the features (e.g. datasets, facilities, policies) characterising the required VRE;
- ***dynamic context management****:* the service automatically creates the security context needed by the service instances contributing to the VRE to work in a secure and organised manner;
- ***open and extensible resource model***: the service relies on a resource model to know what are the available features to be proposed at *VRE definition time* (i.e. what are the features and the capabilities supportable by the currently available services) and how these features have to be deployed at *VRE creation time* (i.e. what are the services to be configured and how they should be instructed to support the requested features);
- ***per VRE customisable UI***: the service offers facilities enacting authorised users to customize the UI of the VRE, e.g. to define the pages it should be structured in, to allocate the VRE UI components per page, to add web content. Moreover, it provides VRE users with the web app needed to use the working environment and its facilities;
- ***ready to use basic services***: the service equips every VRE with key services enabling the VRE members to cooperate by common facilities, i.e. (a) a *shared workspace* to store and organise items of interest; (b) a *social networking area* to, e.g., post messages, have discussions, express opinions; (c) a *catalogue* to publish artefacts resulting from the VRE activity; and (d) a *user management area* to deal with VRE membership (e.g. invite new members), create groups, assign roles.

***Design***

The D4Science VRE Manager Service Architecture is shown in Figure 9.

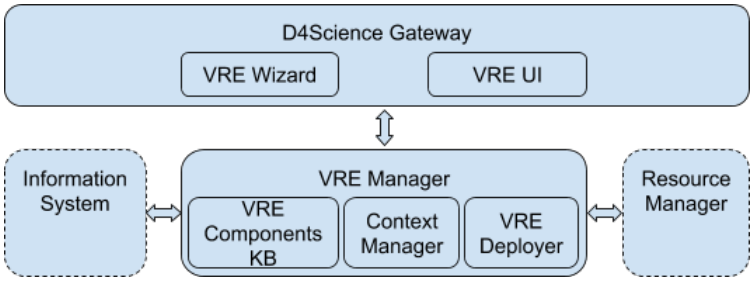


**Figure 9. D4Science VRE Manager**

The main components are:

- ***VRE Manager Service*** implements the entire business logic related with VRE management. It comprises three subcomponents:
    - ***VRE Components KB***, called to build the knowledge base consisting of potential features (and accompanying services) that can be instantiated at VRE definition time. These features will be built by exploiting the information stored into the *Information System*, namely services and their capabilities, datasets, software components, hosting nodes;
    - ***VRE Deployer***, called to transform the VRE specification produced by the *VRE Wizard* into a concrete deployment plan consisting of services (and their accompanying configurations) deployed to satisfy the specified features. This service is also responsible for implementing the deployment plan by either instructing/configuring existing service instances, or creating new ones to serve the VRE application;
    - ***Context Manager***, called to interact with the *Resource Manager* to create the application context needed by the target services to work together and behave as expected.
- ***D4Science Gateway***, is the front-end of the service. It hosts two sets of portlets:
    - ***VRE Wizard***, to support authorized users to specify the features a new VRE should have, by selecting them from a dynamic list of possible features resulting from the D4Science offering, captured by the Information System;
    - ***VRE UI***, to form the specific VRE working environment. These portlets include those providing access to the basic facilities (e.g. user management, shared workspace) as well as those providing access to specific services deployed in the VRE.
- ***Additional services*** enabling the VRE Manager to implement the VRE in the overall D4Science infrastructure settings. These services include the:
    - ***Information System*** providing the VRE Manager with a comprehensive and dynamic list of services and resources currently forming the overall D4Science infrastructure and its operational state;
    - ***Resource Manager*** enabling the VRE Manager to configure existing instances of services or create new ones needed for the VRE operation, and to monitor their availability and behaviour.

***Implementation***

The VRE Manager service, the Information System and the Resource Managers are all based on the homologous software components of the gCube software system, namely Java-based Web Services contributing to the gCube system.

The D4Science Gateway is mainly based on the Liferay portal technology. A rich set of portlets (UI components) have been developed to act as access points to the underlying services; as well as the portal which has been equipped with additional software components integrating it with the rest of the D4Science services. Foe example, components dealing with AuthN and AuthZ, and components interfacing with the Information System.

***Deployment***

The components described are designed to be allocated on many nodes and to exist in multiple instances. In particular, the *VRE Manager Service* can be deployed on a machine other than that hosting the D4Science Gateway. Moreover, many VRE Managers can be deployed in the infrastructure, with each serving a specific *virtual organisation*. This deployment option is key for multi-tenancy scenarios where diverse communities are provided with their own features set at VRE definition phase.

The *D4Science Gateway* has been designed to be deployed on a cluster, with an instance per-node, plus a proxy acting as a unifying access point. Every instance can be configured to give access to a number of VREs (e.g. a community gateway contains all the VREs created for the needs of such a community) and to host the VRE Wizard enabling the creation of new VREs. Every VRE consists of a number of portlets organised according to the VRE specification.

The *Information System* is a conceptually centralised service yet its architecture is highly distributed and scalable, thus enabling it to serve many communities and cases. The resources are registered per-virtual organisation and per-virtual research environment (thus implementing the "application context" created by the VRE Manager).

The *Resource Manager* is a conceptually centralised service having actuators on every node hosting a D4Science service. A hierarchy of interoperating instances can be built, thus having instances taking care of coordinating the management of services at the level of virtual organisation, with instances taking care of resources management at the level of every VRE.

***Use cases***

The D4Science VRE Manager service has been used to deploy and operate hundreds of VREs on D4Science premises. These VREs have been deployed to serve very diverse scenarios from application contexts ranging from agri-food (AGINFRA+) to social sciences and humanities (PARTHENOS), environmental science (ENVRIplus), fisheries and conservation, aquafarming (iMarine, BlueBRIDGE), social mining (SoBigData.eu). A comprehensive list of currently supported VREs is available at https://services.d4science.org/explore.

#### A.2.4: Considerations on VREs, Science Gateways and Portals

The need to develop these services was quite diffuse and many frameworks have been developed to support such tasks. [Shahand et al. 2015] have identified eleven frameworks explicitly exploited to develop Science Gateways including Apache Airavata, Catania SG Gateway, Globus, HUBzero(+Pegasus), ICAT Job Portal, and WS-PGRADE/gUSE. Such frameworks are quite diverse, e.g. Apache Airavata offers its facilities via an API while the Catania SG Gateway offers its facilities via a GUI and a RESTful API. However, they share certain characteristics that make them operate at a certain level of abstraction. For data management, these frameworks mainly focus on files rather than "research objects". Moreover, such specific services are conceived to make it easy to collect data from / interface with existing data providers, thus to make their content available to VRE members. For data processing, the frameworks analysed by Shahand et al. focus on executing jobs.

The D4Science approach and solution discussed in Section A.2.3 has proved to be effective and flexible enough to successfully serve the needs of communities across several domains. The mechanism it offers for the creation of a VRE is unique. In essence, authorised users can simply create a new VRE via a wizard supporting them to produce a characterisation of the needed environment in terms of existing resources. The software (including GUI constituents) and the data needed to satisfy the VRE specification (as well as the compute capacity) are automatically deployed, and no sysadmin intervention is needed.

In the near future, EOSC should put in place a class of services where the deployment of these typologies of service is exposing its users to as limited as possible technical developments. In practice, users interested in having custom views of the EOSC resource space (i.e. the wealth of resources including data and services offered by EOSC) accessible by a dedicated gateway, should only be called to define the "what" (i.e. the EOSC resources they are interested in) and how this custom space should be organised. This model should make it possible for users to easily set up their personal virtual space, as well as a virtual space aimed at serving a designated community.

## A.3. EOSC Resources Management, Hosting Platforms

One of the perspectives characterising EOSC is that of a cloud resource provider called on to provide its users with "resources" enabling them to pursue "open science" efficiently. In fact, this report was informed by

taking into account that EOSC functionalities are provisioned ***as-a-Service***, i.e. they are made available by online services operated by providers taking care of the technical and organisational approaches needed to deliver the planned functionalities. In addition, this report has promoted the notion of EOSC Resource, defined as "*any asset made available (by means of the EOSC system and according to the EOSC Rules of Participation) to EOSC System Users to perform a process useful to deliver value in the context of the EOSC. EOSC Resources include services, datasets, software, support, training, consultancy or any other asset".*

The implications of these two statements imply that there is the need to have services enabling EOSC users to acquire the resources they need for their tasks, e.g. EOSC Resources Management (see Section 4.1), EOSC Hosting Platform (see Sections 4.4 and 4.5). In the remainder of this section, two instances of these services are discussed.

The EGI Federated Cloud Compute (see Section A.3.1) presents a solution for the acquisition of computing capacity at scale. The EOSC-Hub Marketplace (see Section A.3.2) presents a solution for the acquisition of a rich array of services.

#### A.3.1: EGI Federated Cloud Compute (by T. Ferrari)

The EGI Cloud service is implemented in the form of a Federated Cloud. This EGI Federated Cloud is a standards-based, open cloud system federating institutional clouds in order to offer a scalable computing platform for data and/or compute driven applications and services in research and science. The main features are:

*Elastic computing infrastructure*

This feature of the service allows the execution of compute and data intensive workloads (both batch and interactive), hosting of long-running services (e.g. web servers, databases or applications servers), or creation of disposable testing and development environments in VMs and containers. Applications or services can be scaled within a single provider, or across multiple providers of the federation (within providers supporting a given research community), and selecting VM configurations (CPU, memory, disk) and ready-to-deploy application VMs that best fit the user-needs.

*VM image sharing and distribution*

The EGI Federated Cloud allows customised VM images to be easily shared to multiple clouds via the open 'Applications Database' library of Virtual Appliances. Community-curated VMs and VM appliances are securely and automatically replicated across the infrastructure. EGI provides generic, baseline VM images, while user communities can offer specialised VMs and applications.

*Unified view of federation*

The EGI Cloud provides: Single sign-on (SSO) for authentication and authorisation across all resource providers; federated accounting with an integrated view of the resource and service usage; a distributed information system for delivering a real-time view of the capabilities and allow workflow engines to consume this information to distribute VMs in a programmatic way; and federated monitoring to compute metrics for availability and reliability reporting.

*Beyond VMs*

EGI Federated Cloud supports Docker applications on the EGI resources; allowing the use one of the already integrated PaaS and SaaS solutions; the deployment of Hadoop, Docker Swarm etc. to access Object Storage and other IaaS capabilities.

***Design***

The architecture of the federation was defined after a two-year period of development and based on a set of user requirements describing operations on a cloud infrastructure, and was officially launched in May 2014. The EGI Federated Cloud operates as a federation of heterogeneous IaaS clouds and enforces cloud technology agnosticism and service portability in a hybrid environment via the adoption of open standards.

The architecture defines cloud specific capabilities and interfaces and a set of interaction ports with a number of central services that provide the cloud to the cloud federation. The cloud specific capabilities are: (i) Virtual Machine (VM) Management and Block storage management; (ii) Data Management, and (iii) Image Management, provided with the Open Virtualization Format (OVF) and the HEPiX image lists format. The Cloud federation relies on access enabling services such as: (i) Federated AAI, (ii) Federated Accounting, (iii) Information System, (iv) Federated Monitoring, and (v) a federated Service Registry.

The IaaS Cloud capabilities, defined in the architecture, are integrated with the Image Management subsystem, provided as part of the Federated Cloud infrastructure. The EGI Federated Cloud currently integrates resources from OpenNebula, OpenStack and Synnefo providers.

At the infrastructure layer, the service is hybrid and collectively delivered by various heterogeneous providers that are integrated in Service Integration and Access Management system. The providers offer community, private and/or public cloud IaaS capabilities. PaaS and SaaS services are built on top of the IaaS layer. Portability of data and compute workflows relies on a single authentication and authorization federated infrastructure that is supported by the Check-in services, a federated AAI solution that complies to the AARC federated AAI blueprint [AARC 2017].

The service architecture is shown in Figure 10.

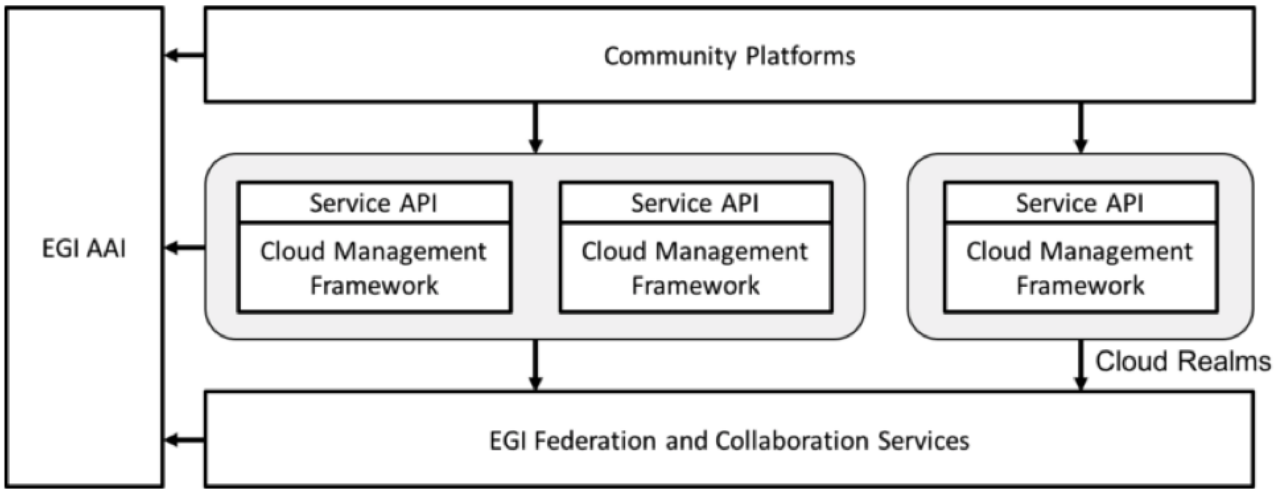


**Figure 10. EGI Federated Cloud Architecture**

Cloud capabilities are provided by the individual cloud providers, while AAI and the Federation and Collaboration Services are deployed to enable the federation. Users can choose to use a single cloud provider or multiple providers, depending on the locality of the data they need to process and the type of workflow to be executed.

***Implementation***

The EGI cloud technology stack comprises open source federation components, maintained by an open consortium, the Federated Cloud Task Force. The cloud middleware stacks supported include OpenStack, OpenNebula and Synnefo.

Security and operational policies support IT security management in the federation.

The EGI Federated Cloud Infrastructure as a Service (IaaS) resource centers deploy a Cloud Management Framework (CMF) that provide one or more of the following end-user capabilities:

- Management of Virtual Machines and of persistent Block Storage devices that can be associated to the Virtual Machines within a single resource center.
- Object storage to manage data as objects with a variable amount of metadata and a globally unique identifier.

These end-user capabilities must be provided via community agreed APIs that can be integrated with the following EGI services:

- AAI to provide Single Sign-On for authentication and authorization across the whole cloud federation. The AAI component supports different authentication and authorization protocols, such as OpenID Connect, OAuth, SAML and X.509.
- Configuration Database to record information about the topology of the e-infrastructure.
- Accounting to collect, aggregate and display usage information. Accounting is based on the StAR standard of OGF [Jensen 2013].
- Monitoring to perform federated service availability monitoring and reporting of the distributed cloud service endpoints, and to retrieve this information programmatically. Integration with monitoring is a passive activity of the resource center, the monitoring is performed using the end-user APIs with regular user credentials from EGI AAI.

Additionally, realms of the EGI IaaS Cloud can integrate with:

- Information Discovery, allowing users and tools to have information about capabilities and services available in the federation.
- AppDB Community-curated catalogue of Virtual Appliances (Virtual Machine Images) and distribution of appliances to the providers of the infrastructure.

Users and Community platforms built on top of the EGI Federated Cloud IaaS have several ways of interacting with the cloud providers:

- Directly using the IaaS APIs to manage individual resources. This option is recommended for pre-existing use cases with requirements on specific APIs.
- Leveraging IaaS Federated Access Tools that allow for managing the complexity of dealing with different providers in a uniform way. These tools include:
    - IaaS provisioning systems that define infrastructure as code and manage and combine resources from different providers, thus enabling the portability of application deployments between them (e.g. IM or Terraform);
    - Cloud brokers, that provide matchmaking for workloads to available providers (e.g. the INDIGO-DataCloud Orchestrator); and
    - Cloud Management Software that provides a unified console for accessing resources and deploy workloads following a set of user-defined established policies (e.g. Scalr or RightScale).
- Using the AppDB VMOps dashboard, a web-based GUI that simplifies the management of VMs on any provider of the EGI infrastructure. AppDB VMOps in turn relies on the Infrastructure Manager, a federated IaaS provisioning tool.

The set of technologies adopted to enable the cloud federation are described on the “Federated Cloud Technology” pages of EGI (https://wiki.egi.eu/wiki/Federated_Cloud_Technology).

***Deployment***

The EGI cloud federation currently includes cloud sites from all across Europe. Providers are organised into realms, with each realm having homogeneous cloud interfaces and capabilities. Community Platforms provide community-specific data, tools and applications, which can be supported by one or more realms. New realms can be defined as needed by the user communities, by agreeing with the providers which interfaces to expose and which of the EGI core services to use for the federation. EGI integrates and maintains a flexible solution portfolio that enables various types of cloud federations with IaaS capabilities and seeking to expand to PaaS and SaaS capabilities.

These clouds are available for users through community allocations, so called Virtual Organisations. Each Virtual Organisation (VO) can get access to a subset of the federated cloud sites according to their local policies, and makes those available for the community members through generic and/or community-specific

policies and protocols. Members of a scientific community have to join the VO to access the cloud capabilities offered by the federated VO sites. The EGI federation model ensures single-sign on (i.e. after a user registers to the VO he/she is able to access every VO cloud); uniform interfaces (i.e. each VO cloud can be accessed via the same/harmonized interfaces) and application portability (i.e. every VO cloud uses the same Virtual Machine (VM) image and contextualization format). VO members can deploy new VMs on the cloud sites through the EGI AppDB VM marketplace (https://appdb.egi.eu/), and can instantiate VMs and block storages via the graphical AppDB VMOps Dashboard or using the API and command line interfaces offered by the cloud sites. High level tools, such as orchestrators and application portals can offer additional, and science domain-specific capabilities for users.

***Use cases***

The flexibility of the Infrastructure as a Service EGI cloud can benefit various use cases and usage models. Besides serving compute/data intensive analysis workflows, Web services and interactive applications can also be also integrated with, and hosted on this infrastructure. Contextualisation and other deployment features can help application operators fine tune services in the cloud, meeting software (OS and software packages), hardware (number of cores, amount of RAM, etc.) and other types of needs (e.g. orchestration, scalability).

Since the opening of the EGI Federated Cloud, the following typical usage models have emerged:

- **Service hosting**: the EGI Federated Cloud can be used to hosts any IT service as web servers, databases, etc. Cloud features, such as elasticity, can help users to provide better performance and reliable services. Example: NBIS Web services or the Peachnote analysis platform.
- **Compute and data intensive**: applications needing considerable amount of resources in term of computation and/or memory and/or intensive I/O. Ad-hoc computing environments can be created in the FedCloud sites to also satisfy very hard HW resource requirements. Example: VERCE platform, The Genetics of Salmonella Infections or The Chipster Platform.
- **Datasets repository**: the EGI Federated Cloud can be used to store and manage large datasets exploiting the large amount of disk storage available in the Federation.
- **Disposable and testing environments**: environments for training or testing new developments. Example: Events conducted on the cloud-based EGI Training Infrastructure.

For more information see: https://wiki.egi.eu/wiki/EGI_Federated_Cloud

### A.3.2: The EOSC-hub Marketplace (by T. Ferrari)

The mission of the EOSC-hub Marketplace is to become the delivery channel connecting the demand-side and the supply-side, and which showcases the potential of integrated and coordinated access to European services, data and other scientific outputs.

In the Marketplace the demand-side of the EOSC (the Customers) and the supply-side (the Providers) meet and cooperate. The Marketplace supports the vision that researchers from all disciplines need to have easy, integrated and open access to the advanced digital services, scientific instruments, data, knowledge and expertise they need to collaborate to achieve excellence in science, research and innovation.

To realise this vision, the Marketplace aims to contribute to overcoming the current fragmentation of national and international providers by:

- Providing turn-key solutions to customers who have advanced digital needs. This includes the provisioning of access to integrated and composable products and services from the EOSC Service Catalogue, and the required expertise necessary to enable users. The EOSC Service Catalogue is defined as "*the list of all live EOSC Services that can be requested by EOSC System Users. It is a subset of the EOSC Service Portfolio, and it populates the EOSC Service Registry*". Exploitation of EOSC services is promoted and enabled by actively engaging with users in open calls to research projects and businesses, and by providing technical support.

- Promoting interoperability and operational maturity of services and products that need to be combined together to support multi-disciplinary science: by working with research communities through the Marketplace, providers get insight on user's interoperability needs and actively evolve their services to adhere to EOSC interoperability guidelines that may emerge as the EOSC development evolves.
- Provide high-level, community-specific interfaces for running workflows involving EOSC services.
- Directly connecting service providers to customers, by allowing providers to get insight about the research organizations and projects that are interested in their products and services, and get credits.
- Giving European-level visibility to the EOSC service products and services of European relevance, and specifically to those who will participate in the EOSC by complying to its policies (FAIR, open science, and conformance to EOSC-selected standards).
- Offering services under homogeneous terms of use, acceptable use policies, and in different configuration options, so that customers are guided in their choice.
- Gathering usage accounting information for the EOSC funders, supporting present and future EOSC business models.

The Marketplace also plays the role of the service provider integration framework, by supporting various EOSC service management processes depending on the level of integration chosen, such as:

- Customer relationship management: the Marketplace maintains information about the customers, funding agencies and research projects that engage with EOSC in compliance with GDPR rules.
- By collecting technical requirements, feedback about EOSC policies and the quality of the offered solutions, customers are actively engaged in EOSC.
- Service level management and service report management: in a personalized dashboard, customers are informed about the availability of the required services, have access to the applicable Service Level Agreements and can review service performance metrics.
- Capacity management: service providers who decide to integrate at a medium/high level allow users to get accounting aggregated views that provide, in a confidential way, aggregated information about the usage of the requested services. By allowing users to place orders, providers can estimate the current and future capacity needs of EOSC users.
- Incident and problem management: providers are offered the opportunity to use a central incident management system to answer support requests from users. The EOSC helpdesk system can be integrated with an in-house helpdesk system if the provider already has a custom platform available. Rather than requesting support to the multiple suppliers participating in the EOSC offer, users are offered a single entry point to submit support requests.
- Service request and order management: customers are provided with the possibility of ordering services online where applicable, and to bundle these together in projects. The providers are offered an online platform for managing orders, if they don't have one in place.

The value proposition of the service is to:

- Facilitate service and resource discovery and access at the institutional and inter-institutional level.
- Collaborative improvement of services and resources.
- Facilitate inter-disciplinary research by providing access to technologies typically considered outside of a particular field.
- Increase competitiveness by providing a low-cost of entry to expensive technologies for small academic institutions and businesses.
- Allow researchers and institutions to focus on value creation as opposed to maintaining redundant resources.
- Researchers can discover expertise that can be tapped into, based on usage of resources available.

- Ensure efficient resource usage at the institutional, national, and international level.
- Remove administrative burden from technology platforms allowing the developer of new services to focus on technology delivery instead of administration.
- Allow cost sharing with accounting, billing, and enabling of fair usage of resources.

***Design***

The Marketplace provides user and customer capabilities delivered through a user interface, and service provider capabilities delivered through a service provider interface. Each interface provides capabilities that rely on service components as detailed below.

**Service Provider Interface** (see Figure 11):

- Register service metadata and get support:
  - Service component: Jira.
- Negotiate Service Level Agreements associated to service access requests:
  - Service component: Service Management Dashboard.
- Profile service orders and check status:
  - Service component: EOSC-hub Operations Portal (providers' interface) and My Services personal dashboard (user provider interface).
- Provide support:
  - Service component: EOSC-hub Helpdesk.
- Provide accounting information:
  - Service component: EOSC-hub Accounting portal.

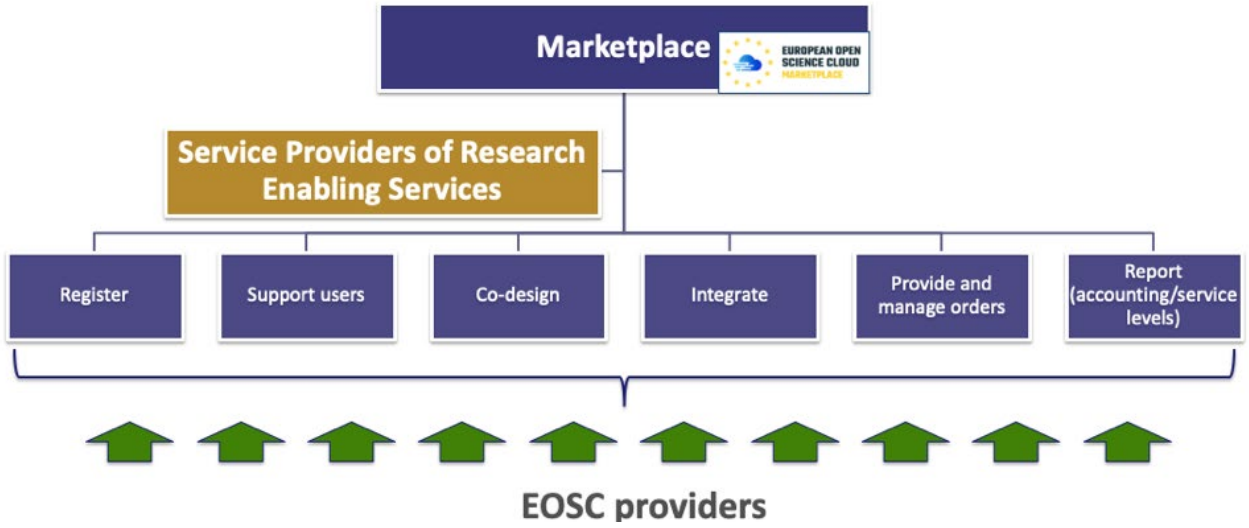


**Figure 11. EOSC Marketplace service providers' interface**

**User Interface** (see Figure 12)

- Discover and compare multiple resources and services such as scientific outputs, applications, research-data exploitation platforms, research-data discovery platforms, data management, compute services and thematic services. For each service, find information on access policies, the maturity level, and other related interoperable and programmatically composable services:
  - Service components: third-party metadata catalogues, research infrastructure and e-Infrastructure service/tool/data registries; the EOSC service portfolio management tool (SPMT).
  - Resources: Data, software, applications, workflows, publications and other research outputs.

- Allow customer to select services, service options and attributes and submit related orders:
  - Service component: Marketplace GUI.
- Be registered, authenticated and authorised using the academic/social account of choice, with the possibility to choose various federated identity providers and Research Infrastructure Identity Providers (compliant with AARC):
  - Service components: the EOSC AAI, and federated identity providers.
- Reuse data that conforms to community-defined FAIR best practices:
  - Service components: FAIR-conformant certified data repositories and metadata catalogues, community-verified repositories and metadata catalogues.

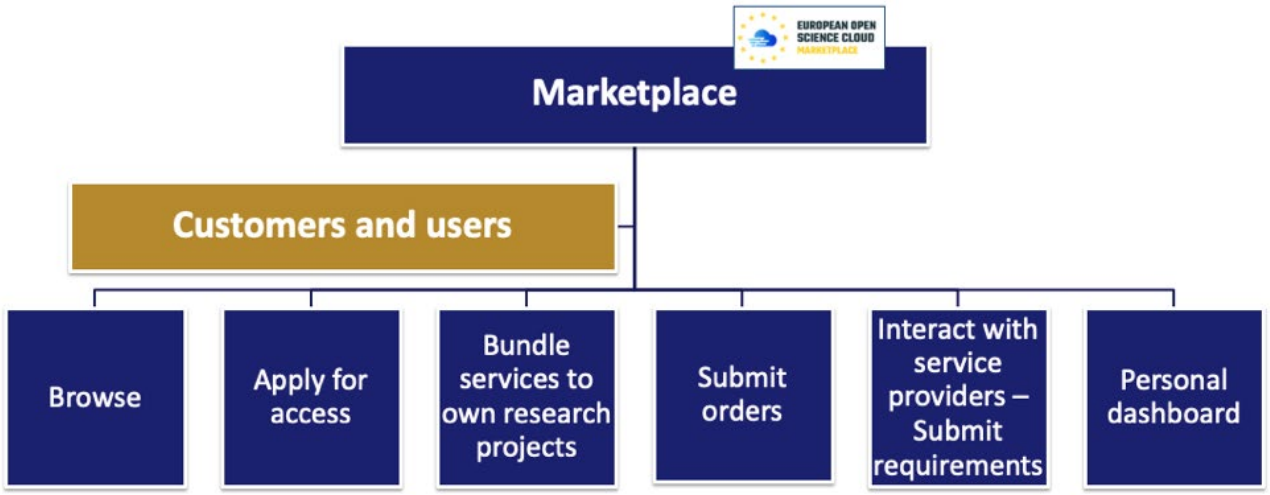


**Figure 12. EOSC Marketplace users' interface**

The EOSC Portal Marketplace is designed as an electronic market - it is a platform where services can be advertised and where customers can easily order and access them. In addition, the Marketplace will enhance visibility for resource and service providers, raising awareness of what they can provide as well as helping to promote cross-disciplinary research.

The Marketplace service architecture is depicted in Figure 13.

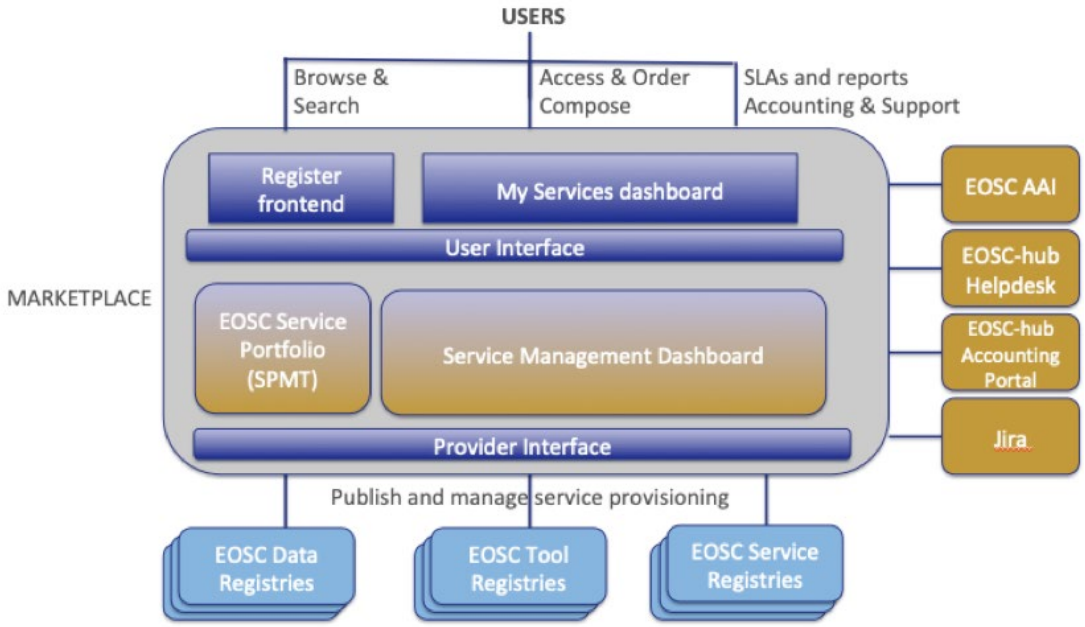


**Figure 13. EOSC Marketplace Service Architecture**

***Implementation***

The Marketplace is an open source project maintained in GitHub. The implementation is in Ruby on Rails technology with PostgreSQL as the main database. Additionally, ElasticSearch is used to support full text search.

***Deployment***

Since the official launch of EOSC on 23 November 2018, the EOSC Portal Marketplace is accessible at the following URL (http://marketplace.eosc-portal.eu) and it currently comprises 60 service entries, and the set is progressively growing.

The application is built in alignment with good practices for scalability of the installation of the application. Currently, the Marketplace is running in HA mode delivered from two sites.

The Marketplace is currently a central access-enabling service of EOSC. Mechanisms for the active pulling and pushing of information from external registries is currently in design phase. Similarly, a distributed order management Marketplace component is in design.

**Use cases**

The typical workflows supported by the Marketplace are as follows.

- *Authentication*: the user browses services and can login in order to perform actions that require authentication. The ESCO AAI allows customers to use the credentials of their home organization. Customers are required to register during their first login into the Marketplace to create a customer profile in the database. Part of the data is retrieved by the user IdP of choice, and additional data is gathered through a form as necessary.
- *Discover and order services*: Customers browse the Marketplace, find what is needed and select the services they require. Different service configuration options and related attributes can be supplied according to the specificity of the service and the needs of the user.
- *Check-Out*: The service access requests and orders are submitted with information complemented from the customer profile. The order then follows the appropriate service order management. The user is directly redirected to the service end point/API in case of services that do not require a customized configuration and authentication.

- When placing access requests, the services are bundled into research projects and connected to the user identity and home organization. The My Services dashboard is the place where the user can check the status of the access procedure. The user is notified by email about relevant information necessary to publish the service.

Service providers can:

- Publish, share and advertise services and resources to a wider user base.
- Inform about the protocol and standard support capabilities of their services.
- Get information about received access requests, and related organizations and research projects.
- Get user feedback.
- Get a free online platform, included in the marketplace, to publish their services and register the services that successfully meet EOSC service portfolio policies in the marketplace; Where providers who conform to the EOSC policies, can also manage service requests, interact with users and provide support to them, and agree the most suitable service levels.
- Join the group of providers that meet EOSC policy requirements.

**A.3.3: Considerations on EOSC Resources Management**

The implementation of Services for Resource Management depends on the variety of resources needing to be managed. The larger the set of typologies that can be acquired by the user, the more challenging are the basic functionalities of resource discovery and resource configuration.

With resource discovery, it is known that approaches aimed at offering a single classification of the resources (namely services) to enable discovery by browsing is not effective for every user. One of the consequences of this is to envisage approaches where the resources can be tagged by using terms belonging to diverse classification schemes.

Another issue to be tackled when implementing this typology of service in the context of EOSC is related to the hybrid cloud nature of this system. This implies that the resources (namely services) that can be acquired by the Resource Management service will come from diverse providers. To make this effective a strong coordination approach must be established between the service provider of the Resource Management service and the providers of the resources offered by the Resource Management service.

Last but not least, this service should support the acquisition of bundles of resources by a single transaction. In essence, EOSC users will benefit from a service enabling them to select a set of resources by taking into account that these resources are likely to work together. This has implications on resource compatibility (e.g. the system should alert users that they are trying to acquire resources not designed to work together) and dependencies (e.g. the system should alert users that the resources they are selecting require other resources to work properly).