Project acronym: CS3MESH4EOSC

Deliverable D2.1 Implementation of federated identity and group management

Contractual delivery date	31-12-2020
Actual delivery date	12-03-2021
Grant Agreement no.	863353
Work Package	WP2
Nature of Deliverable	D (Demo)
Dissemination Level	PU (Public)
Lead Partner	SURF
Document ID	CS3MESH4EOSC-20-008
Authors	David Antoš (CESNET), Milan Daneček (CESNET), Daniel Müller (WWU), Renato Furter (SWITCH), Ron Trompert (SURF)

Disclaimer:

The document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 863353

Versioning and Contributions History

Version	***Date***	***Authors***	***Notes***
0.1	04.12.2020	David Antoš (CESNET), Milan Daneček (CESNET), Daniel Müller (WWU), Renato Furter (SWITCH), Ron Trompert (SURF)	Initial version
0.2	11.12.2020	David Antoš (CESNET), Ron Trompert(SURF)	Textual improvements and section 7 updated
0.3	15.01.2021	David Antoš (CESNET), Ron Trompert(SURF)	Removed some links. Adapted section 4.2.1 and 7.
0.4	09.02.2021	Pedro Ferreira (CERN)	Added some extra formatting and did some textual improvements.
0.5	16.02.2021	David Antoš (CESNET), Ron Trompert (SURF)	Textual improvements

Index

Index of Figures

Index of Tables

1 Introduction

This document accompanies the Work Package deliverable D2.1, which is a **demo** available as a video (screen recording) at https://www.youtube.com/watch?v=I0PlqiUBT0I. The demonstration itself would be of little information value without proper explanation. Moreover, the demo format does not provide opportunities to explain design decisions that led to the approach taken to build this federated EFSS infrastructure. Besides the demo description, this document sums up the **first project year** in **Work Package 2**, reporting on its progress during this time period. Note that this document summarises the work done in the first year and some topics in this document are still subject to discussion and have thus not been finalised yet.

As of the inception of the project proposal, which was written between December 2018 and January 2019, it was expected that deliverable D2.1 would be about a demonstration of implemented **federated identity management**. We then held the view that federated identity management was crucial for the project and the **sharing of data and applications** over different Enterprise File Sync and Share (EFSS) systems **could not exist without it**. There has been extensive insight ever since that this may **not** be the **best approach to tackle the problem**. The **Interoperability Platform (IOP)** that is currently under development and has been deployed at majority of partner sites, plays a significant role in federating the infrastructure. Amongst other things, we have considered the option that the IOP would implement an **invitation workflow**, which would be of key importance to solving this issue in an elegant way. This workflow would result in an **exchange of tokens** which would allow users to **share data and applications** over different EFSS installations at different sites where they only need to know each other's email addresses or similar contacts. While still agreeing that federated identity and group management systems can be used for this end, we have realized that they are **by no means a necessity** in order to create the **mesh functionality** that the CS3MESH4EOSC project aims to deliver.

In **Section 2** we describe the **demo** itself. The description should be self-contained for readers and demo viewers who are just interested in the status of things. More in-depth information may be found in the subsequent sections, especially reasoning behind the design decisions. User, group and application workflows are described in **Section 3**. Building the Science Mesh federative infrastructure with AAI, monitoring, accounting, security, mesh governance and requirements related to the ingress of new sites are described in **Section 4**. In **Section 5** we list the sites which have currently deployed the interoperability platform. **Section 6** deals with entering EOSC and the requirements that need to be fulfilled. Finally, **Section 7** contains a brief discussion about future work.

2 The Demo

The **demonstration** is available at https://www.youtube.com/watch?v=I0PlqiUBT0I as a video showing concepts and currently implemented tools to establish trust between users based on sites federated in the Science Mesh, and to use this trust to share resources. There are currently no graphical user interfaces yet (they are part of the short-term plan, though). All operations are currently performed through a command line interface, so the process requires thorough textual explanation.

First, it is necessary to introduce the **components** which play a role in the demonstration. The basic prerequisite is the **OpenCloudMesh API (OCM API)** defining the protocols to create shares, retrieve shares, and send and retrieve notifications. The OCM API expects the authentication to be already established between the sites/users. Moreover, the OCM API does not solve how user IDs are found, it just expects the communicating actors to have this information. It does not deal with accepting or rejecting the shares, this functionality may be implemented in the end systems.

We have two users of EFSSs, one called **Originator** (**O** for short) and using CESNET's ownCloud instance, and one called **Target** (**T**) using CERNBox. Both systems in the demo run the Interoperability Platform (IOP) Reva, which acts as an abstraction layer between the particular EFSS and the OCM API, regardless of the protocol the system uses for sharing. The same functionality could be implemented by any other system respecting the same set of protocols, however.

Moreover, Reva implements the functionality of the **Execution Module** as described in Section 3. Specifically, Reva acts as a user interface for the "Where Are You From" (WAYF) phase of establishing trust, and thus as a "service discovery" service, having knowledge about Science Mesh topology and services (applications) running at Science Mesh sites. Reva further implements orchestration of data transfers and access to collaborative tools like editors.

The configuration of the Science Mesh is kept centrally in the GOCdb database. The database stores the Science Mesh topology (in a structure very similar to, e.g. identity federations). The topology information includes sites and applications running at the sites that are registered in the Science Mesh. Mesh topology is propagated to all sites regularly when changed. Note that just including a site and/or application in the Science Mesh database does not necessarily mean it is accessible for all; Reva will implement permission rules for the local site (both outbound and inbound).

The user's view of the demonstrated scenario is simple: it can be described by the user's actions and divided into two phases:

1. establishing trust between a pair of users, and
2. accessing the resource itself, be it setting up a share or gaining permissions to access a collaborative application.

Note that the share itself (and similarly for any other resource that can be accessed) is then expressed as a database record in **User O**'s EFSS system, representing the permission of the target user to access the resource. (See the CS3 API documentation[1] for details.)

Establishing trust between the users expects no previous knowledge about **T** from the part of **O**, except a way to contact **T** through some form of reasonable text communication like, e.g. email, instant messaging, but also QR codes. As email is still the *de facto* communication standard and an email address the most likely contact information the users know, the demo depicts sending an email invitation.

An example scenario

Let us suppose that O wants to share a resource (e.g. a folder) with T. User O does not have to know T's system nor ID in the system. User O uses the interface for sharing setup in the system (here CESNET's ownCloud) to send an email invitation to T. User T receives the email containing a link to the WAYF component of O's system IOP. User T opens the link and is provided with a list of Science Mesh sites to choose from, similar to a list of identity providers in federations, T chooses the site (here CERNBox) and logs into the home system. During this procedure, T reveals the user ID in the system as well. At this point, everything needed to establish a trust relationship between O and T is known: both systems and user IDs there. Tokens are exchanged between the two systems (T's system contacts O's one via a public endpoint and produces the original token there to show legitimacy of the request) and trust between O and T is set.

[1] https://cs3org.github.io/cs3apis/#cs3.sharing.collaboration.v1beta1.Share

Currently, tokens and emails are generated through a CLI interface. In the final implementation, this will, of course, be done through graphical interface and caching of previously established relationships between the users, including means to manage the cached information.

> *Sharing the resource can be done after trust is established between the users. User O can now create a share for T by means of Reva CLI.*

In the demonstration, establishing the trust and sharing the resource are separate actions, which is obviously not intuitive for general users. In the final implementation, they will be connected into a single procedure so that upon the first time O shares something, that will trigger the trust establishment procedure automatically. Moreover, the relationship will be cached in O's system once established and the trust establishment phase will be skipped for future shares. All the procedures will be equipped with suitable graphical user interfaces.

3 User, Group, and Application Workflows

This section describes the design of workflows for users and groups, that allow them to establish data sharing, application access, and access to resources in the Science Mesh in general. We discuss the final state of the design in this deliverable. For more detailed information on development and design choices, we refer to the proposal drafts and working documents[2] that cover design choices in greater detail (although in a much less structured form). A short description of the reasoning behind the design can be found in section 4.1.

This design is the result of several fundamental decisions. First and foremost, the place to keep resource sharing information has been chosen to be always the originating system, keeping it both for local as well as remote users - there is no central authority for that. Moreover, after considering an architecture with a centralised sharing broker, we realised that the same functionality can be achieved with a distributed design. In this document, we describe just the distributed version - both for clarity and brevity.

3.1 Overall System Architecture

There is a single central entity, the **Configuration Database (CD),** keeping the metadata structure of the Science Mesh (this role is implemented by a GOCdb[3] instance. GOCdb is developed and maintained by the EGI[4] project). Each end EFFS is equipped with an **Executive Module (EM)** that provides communication in establishing trust between users and sharing resources, and also as a service discovery interface for the users. This role is implemented by the Interoperability Platform (Reva), but it may be natively implemented by the EFFSs themselves. The proposal expects the sites to have a functional Public Key Infrastructure in place, with keys verifiable in the metadata.

Resources are files, folders, and access to applications (e.g. a collaborative editor). Access can be granted to an individual user or to a group of users. Note that authorised actions (read vs. read-write, permitted options of a shared editor, etc.) are configured in the originating system and completely

[2] https://surfdrive.surf.nl/files/index.php/s/e0OzLdzzOrdMzXD
[3] https://wiki.egi.eu/wiki/GOCDB
[4] https://www.egi.eu/

dependent on the originating system and are therefore beyond scope of this proposal.

The **Originating system** is an EFSS that holds the resource-to-be-shared and the sharing is initiated by an **originating user** (Originator, O for brevity; we also talk about O's EFSS)**.**

The **Target system** is a home EFSS of a user who is to be permitted to access the resource, the **target user** (**T**). Note that shared data typically resides in the originating system (unless some caching is involved). A person may naturally be a user of multiple EFSSs, but, for the purpose of this document, we concentrate on a single one acting either as an origin or a target. In other words, O and T should be understood as roles of particular actors.

For all scenarios and use cases described here, keep in mind that sharing a resource to a set of users just means executing the scenarios repeatedly for all of them; the same holds for sharing to groups from various target systems.

Sharing policy is a machine-interpretable set of rules describing what types of sharing and to what partners (or types of partners) are permitted to and from an EFSS. It may range from "anything is permitted" to strict rules for confidential content. Sharing policy is defined by the operator of a particular EFSS. Sharing policies are defined both for originator as well as target roles of the system.

3.2 "User Invitation Workflow"

Let us recall that the OCM API supports establishing sharing and expects O to know not only T's EFSS, but also T's exact identity there, which is exactly the type of information we cannot reasonably expect users to know. The goal of the invitation workflow is to provide a user-friendly discovery mechanism of user identities which is private by design and thus GDPR-compliant. We expect O to have a textual way of communicating with T (a text-based side channel outside the Science Mesh), like an email address (or, in fact, any other capable of transferring sufficiently long tokens, like chat).

The process is as follows:

1. **User O** will either
    a. type **User T**'s email address into the **originating EFSS** and then the **originating EFFS** will **send an invitation to T**. The message will contain an authorisation code and an invitation link (also encoding the authorisation code in it). The authorisation code will be signed by the EM of the originator. Then continue with step 2.
    b. or type an **email address** and be offered a list of previously associated identities of **User T**.[5] **User O** chooses one from those identities directly. Since **Users T's** identity is already cached in the **originating EFSS**, trust between two particular users has been established before. **O** can then select the resources to be shared and share it with **T**. The procedure is then finished.
2. **User T** will receive an email with the **invitation link**. The email will be signed by the **originating system** and the list of systems and their public keys will be well documented (to prevent phishing attacks etc.), it will be accompanied by **User O**'s personal message.

[5] Note that trust relationship is established just between the pair of users, i.e. O is offered a list of previously established target users for which O personally initiated the procedure. The trust cannot be reused by other users of the originating system.

a. **User T** will log into the **target EFSS** and will copy the invitation code into it. The **target system** verifies authenticity of the link and contacts the originating system **EM** with a message containing:

   i. an authorisation code,

   ii. identity of the target system,

   iii. **User T**'s user ID.

   The message will be signed with **the target system**'s private key.

   In addition to that, User T will be asked for consent with storing this information at the **originating system**.

   User O's (originating) system now has the full information necessary to establish the trust needed for sharing (i.e., information of what shall be shared encoded in the authorisation code, the target system and T's ID there).

   Once sharing is established, and if consent has been given, the relationship between **User T**'s email, **T**'s system, and **T**'s ID in the **target system** will be recorded in the **originating system**. Without consent to cache the information, it is used only a single time, to establish permissions to access a resource.

b. **User T** can open the link in the email and will reach the originating system's EM web interface providing the WAYF service. **T** is offered a list of Science Mesh sites to choose from. **T** chooses the **target system** (similar to identity federations). They are then redirected to the target system and logs into it. At this point, all variables are known, the authorisation code, the target system as well as **T**'s ID, and we can proceed as described in the previous option, passing this information on to the **originating system**, to initiate the establishment of a share.

Policies of both systems are checked during the trust establishing procedure by both EMs (outgoing policy of the originating system by its EM, incoming policy of the target by the target EM).

Note that this procedure incorporates privacy-by-design: using User T's email address is beyond the scope of the system (it is User O's responsibility to use it legally). Should T not consent to caching the identity, it is used just for a single case of resource access setup procedure. T is free not to reveal the identity or to reject the shared resource. In the latter case, T's identity is forgotten by O's EFSS.

The procedure can be further enhanced by providing a special link for EFSS mobile applications, but we leave it beyond scope of this document.

3.3 Group Workflow

The purpose of group sharing is to delegate group management onto a representative of users of a target system. E.g. user O from CESNET creates a share and provides access to it to other users from the same system directly. User T from CERN creates a group in CERNBox and user O adds the group to the share on CESNET ownCloud side. Thus, user O delegates management of CERN colleagues in the share to user T. Note that identical workflows may be deployed for basic file sharing as well as for accessing any other resource in the Science Mesh. The procedure is completely agnostic about what the resource is.

After discussions about possibilities of group enumeration from either side, we have concluded that if enumeration of the group is of concern to the originating user, there is always the invitation scenario, which allows to manage all group members directly, with no need for enumeration. Implementing enumeration would bring in complexity that is beyond reasonable when compared to the potential benefits.

The group in the target system is defined there and is not exported anywhere else, except that the target system grants members of the group access to resources shared with the group in the originating system. Group membership is thus evaluated at the target side. Note that the group may be actually defined in an external identity management system on the target side (e.g. in an infrastructural LDAP) and the target EFSS may just be controlled by the infrastructure.

In cases when above-described practical limitations in member enumeration are not acceptable, i.e. when detailed control over access to a resource is requested, the originating user is advised to use the invitation scenario instead and to manage access to the resource purely on the originating site. We nevertheless still consider this use case to be meaningful as it provides delegation of group management.

Should a discovery procedure be deployed for group sharing as well, the process may be similar to the user workflow described in the section 3.2. User O sends an email invitation to the person who acts as a group admin in the target system. The rest of the procedure is a direct analogue of sharing to a single user. The share will be accepted by a group administrator in the target system.

3.4 Application Workflows

We recognise two use cases: first, a user needs to access data from a Science-Mesh-enabled application. In the second use case, the user intends to use a remote application, e.g. a collaborative editor running at another site.

3.4.1 Accessing Data from a Mesh-enabled Application

Let us consider this use case: user O using CESNET's ownCloud creates and edits a file with a MyFancyEditor application. He realises that he actually wants to store the file in CERNBox where he has an account as well, but he likes to edit it with MyFancyEditor that is not available there.

In more general terms, the user is using an application in the Science Mesh (e.g. in the user's home institution ecosystem or wherever the user has access to) that is able to keep the data in an EFSS solution and intends to save the data from the application to another EFSS in the Science Mesh. There must be a discovery procedure for file load and save operations to establish which is the EFSS to use (i.e. WAYF equivalent).

As the originator/target site terminology is extremely confusing in this case, we use the terms *data site* for the system where the data should reside, and *application site* for the site with the application. We again rely on the Executive Modules of the EFSSs. The procedure is as follows:

1) The application is running at the **application site**. The user chooses the option "save to the Science Mesh/load from the Science Mesh" in the application.
2) The **user** is redirected to the **EM** running at the **application site**, to the WAYF component.
3) The **user** chooses the **data site**.

4) The EM of the **application site** checks its policy; if such usage is permitted, it lets the user log into the **data site** and passes a signed request to access the data.
5) The **data site** checks the request and if it complies with the policy and is factually correct, it creates a token to access the data and passes the token to the **application site**.

3.4.2 Using a Remote Application

A user has data in an EFSS and wants to use a service running elsewhere in the Science Mesh (e.g. "open this file kept in CESNET ownCloud in CERN's Jupyter Notebook"). In that case, the user chooses the option "Science Mesh/open this file with a remote application", is redirected to the application list maintained by the Science Mesh and available through a service discovery (WAYF) and selects an application[6]. The procedure handles token exchange for the application to access the data. The permission to use an application can be handled in a similar way as a permission to access a data share. Let us suppose the application resides on the originating system (here we can keep the terminology consistent with data sharing, the application is the resource to be shared). Permissions to use the application are kept on the originating system in the same way as for data sharing. An application administrator can invite a user from a target system by means of an email invitation (slightly modifying use case 3.2 as described above).

Let us use an example. **User O** using CESNET's ownCloud (the originating system, i.e. where the resource lives) is a MyFancyEditor application administrator. User O wants to invite user T from CERN (the target) to use this application.

The procedure goes as follows:

1. **User O** uses an application access control interface in the originating system to construct an **invitation email for User T**, using just T's **email address**. The email will contain an authorisation code included in an invitation link to the originating site EM.
2. **User T** will receive the email and use the link to get to the WAYF component of the originating system's EM. **User T** chooses their target system there. The originating EM verifies such a combination conforms with its application access policy.
3. The WAYF redirects **User T** to the **target system** and **User T** logs into it. The target system verifies conformance with the application access policy.
4. If the sharing is allowed, the **originating system** inserts **Users T's ID** in the target system into the group allowed to access the application.

4 Building the federation

This section describes the work done in turning the EFSS installations operated by CS3MESH4EOSC partners into a federated infrastructure. Within the creation of EOSC, things are still very much in development and therefore subject to change. On the other hand, the CS3MESH4EOSC project only has a three-year lifespan and at the time of writing there is only two years left to go. It is our goal that within the coming two years the CS3MESH4EOSC project will deliver a federated production quality (TRL9) EFSS-based infrastructure. This implies that we cannot wait for the dust to fully settle but instead need to make some of the design choices now, based on what is the *status quo* and the

[6] A list of applications will then need to be included in the Science Mesh metadata database in addition to the list of sites necessary for basic file/folder access scenarios.

Project's vision of itself as part of the European Open Science Cloud. The former will continue to follow the developments around EOSC closely and make its choices in such a way as to not preclude being an integral part of EOSC's realized infrastructure further down the road.
This has led us to the design decisions described in this section.

4.1 AAI

The ultimate goal of the the Project is to enable controlled group sharing, data replication, and other resource access throughout a set of heterogeneous EFSSs. In order to do that, the infrastructure must necessarily support identity and group management in a manner comprehensible to users.

4.1.1 Development of the AAI design

During development of the proposal on how to tackle the AAI design, we have discussed a list of **functional** and **non-functional requirements** in order to establish which properties of identity management are necessary for the project. Those discussions covered mainly the following areas:

1. There must be **a way of establishing trust** among EFSS instances, without the need to create peer-to-peer deals with all other instances involved.
2. **Group management and group membership management** over **heterogeneous resources**, so that a group accessing a particular resource can consist of user identities from various EFSSs.
3. **Sponsored accounts** would be useful to cover cases where users have a legitimate reason to access the resources while being unable to prove their eligibility through direct technical means (e.g. producing relevant attributes in an identity federation).
4. **Identity consolidation** would allow a person with **multiple digital identities** to **connect** them and use the information to access resources in a comfortable manner.
5. **Service accounts** for data and/or access that "does not belong" to that person but rather to a group, laboratory, department etc.

During the writing of the project proposal and until early stages of the project, it was generally expected that the overall goal would be to unify and federate group management into a single distributed system. While there are significant efforts in EU and EOSC in particular to unify user management based on identity federations and group management system (e.g. eduTEAMS, group managements based on ELIXIR ID, etc.), we realised that there are **three major challenges** that make deploying them on global scale of the Science Mesh difficult:

1. Those systems are **not yet, with regards to interoperability**, at a development stage where **unifying groups defined in them would be easy**, convenient, or even sometimes possible, as they usually cover distinct communities defined by science fields. That cannot be reasonably expected to change during the project time frame.
2. Those systems cover a **slightly different use case**: accessing a **single resource by a group of federated identities**. The Science Mesh needs means to control accessing a set of diverse distributed resources by groups of users.
3. There are **EFSS** in operation that **do not use external sources** of group membership information at all.

While we hope for group management under EOSC to get gradually more and more integrated and interoperable, that **cannot be realistically expected** to happen during the **coming five to seven years**, not just for **technical reasons**, but also because of the inevitable need for **wide political agreement**

on this topic, throughout an extremely diverse community with a broad spectrum of needs. A solution is necessary **at least to cover the transition period**.

Taking those practical limitations into account, we developed an **orthogonal approach** to the problem, that allows users to share data among EFSSs regardless of where group membership information is kept in the systems. The EFSS themselves are taken as the logical source of group membership information, regardless whether they actually are the sole source of truth, or whether they are just controlled by other components in the infrastructure. This approach thus **by no means conflicts** with expected tighter integration of user group management in the future.

Out of the list at the beginning of this section, we have dealt with establishing trust among EFSS instances by means of federating them in a scalable way, and group membership by providing means for the users to invite other users and groups to use various resources, be it file sharing, transfer, or access to remote applications. Sponsored accounts, identity consolidation, and providing service accounts are left beyond scope of the Science Mesh. Such functionalities can be supported by the local identity and group management systems or by the EFSSs themselves.

4.1.2 Other Design Considerations

Non-functional requirements (relevant after taking the aforementioned design decisions) include:

1. **Minimal** sets of **personal information** should be communicated among the systems in the infrastructure, following privacy-by-design/GDPR principles. Should there be any optional functionality (e.g. caching user records for future use), storing that additional personal data should be **based on user consent**.
2. Each user may be **a member of various groups**, the membership information may not be completely stored locally (just in the user's home EFSS).
3. Every **access control mechanism** that is based on rules must have a possibility to add a **list of individual exceptions** (both admissive and the contrary).
4. **Eligibility of users** to use individual EFSS instances depends solely on **local policy** established by the EFSS operator (and is usually derived from financing, national and/or institutional organisation structure and policy, etc.). There is absolutely **no intention to unify those policies** that are and shall remain defined by the EFSS instance administrator. Technical expression of those rules may be quite complex, and it is often centrally managed by national e-infrastructures.

4.1.3 Survey

The design decisions described previously are further backed by detailed information about sites that are expected to join the Science Mesh (project partners) and which demonstrates the complexity of user identity and user group management practices throughout the sites. We have collected information about user management practices in Science Mesh sites using a simple questionnaire. The results were collected from 9 infrastructures. We have obtained the following results.

Current system being provided
7 sites operate Owncloud, one seafile and one uses a proprietary solution. In addition, two sites operate parallel Nextcloud services internally**.**

A single site is directly connected to eduGAIN, 6 use their national academic identity federations, one

none of those. Out of the federated ones, two accept all identities from the federation, while others perform various levels of attribute filtering to establish eligibility of the users to access the services based on infrastructure policies. The filtering ranges from simple organisation affiliation to quite complex rules based on combination of attributes for various organisation types (e.g. accepting students and members of universities, staff of public libraries etc.).

Two sites support identity consolidation, i.e. they allow users to link their identities together and use any of them on a daily basis. One service creates LDAP accounts independent of the identity federation (mainly in order to enrich user attributes as well as to provide grace period for the users who are leaving the eligibility criteria), the remaining majority of sites permit logging into the service with the federated identity.

Four sites support OAuth2, four OIDC, and the remaining ones none of those.

Group membership
Regarding group handling, this area is most diverse of all. Some sites stick to group management provided by the EFSS solution, some operate their in-house extensions and/or local group management system integrations.

Not using eduGAIN
The answer to the question whether connecting the service to eduGAIN is possible was most of the times that it is possible but has limited use (at most as a source of guest identities) as only users covered by the current source of identities are eligible to access the service anyway. One site indicates that eduGAIN is not supported and one is already an eduGAIN service provider. All national identity federations are taking part in eduGAIN, though.

Future plans
We have also surveyed short- to mid-term plans of possible changes of the EFSS solution. Most sites intend to stick to their current systems (just following newly released versions), a single site considers switching to another fork of the EFSS they operate.

4.2 Monitoring, Accounting, QoS, Security

4.2.1 QoS and Monitoring

To ensure overall high quality of service within the Science Mesh, all sites and their services must be constantly monitored and checked for their health status. Additionally, metrics, like the total number of users, must be gathered and presented in a consumable way to get an overview of the mesh in its entirety.

The mesh metadata - i.e., all site information and their respective services - are stored in a central GOCdb instance. The GOCdb is a web frontend for storing and managing grid topology information in a streamlined way; it is part of the EOSC-hub and is maintained by EGI. Currently, an on-premise version is used. In section 6 we explain why joining the EOSC as a e-infrastructure requires us to have our own registry of resources, so for that reason we should operate one ourselves. The following screenshot shows the entry page of the GOCDB:

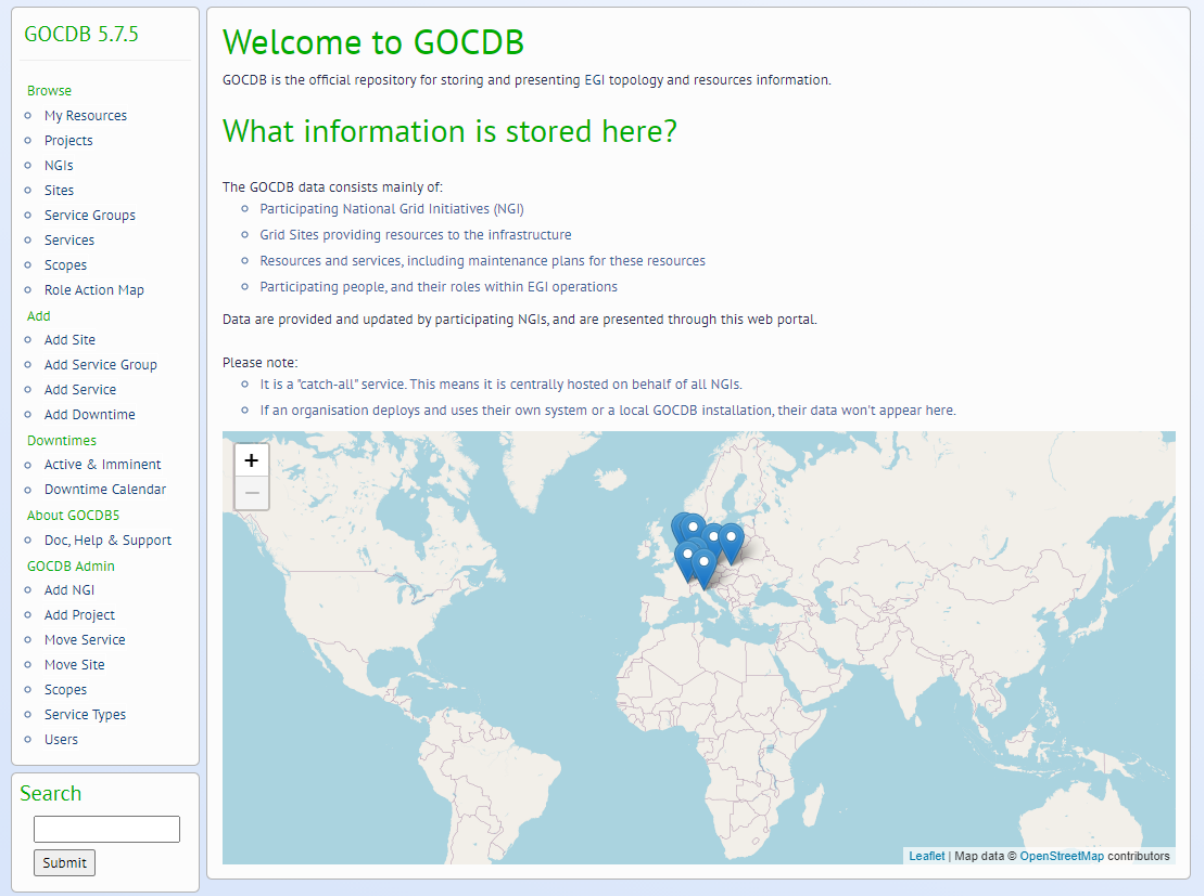


Figure 1 GOCdb screenshot

The GOCdb is part of the so-called ***central component*** which offers global services necessary for the operation of the Science Mesh as a whole. Other services include a Prometheus instance which are used to gather and present metrics and health information of all sites and services of the mesh, as well as a global Reva instance. Another major service of the central component is the Blackbox Exporter which is used for performing health checks on the various site services.

To connect all these different components, a service called *Mentix* (short for Mesh Entity Exchanger) was developed. Mentix is responsible for gathering site and service metadata from the GOCDB and exporting it to the other services of the central component. This includes offering the site information via HTTP endpoints - which can then be consumed by other services which require information about the Science Mesh topology -, as well as writing out target information used by Prometheus. Mentix can thus be seen as the bridge between the mesh topology information provider (i.e., GOCDB) and all services that require this information. An overview of the general workflow of Mentix can be seen in the following screenshot:

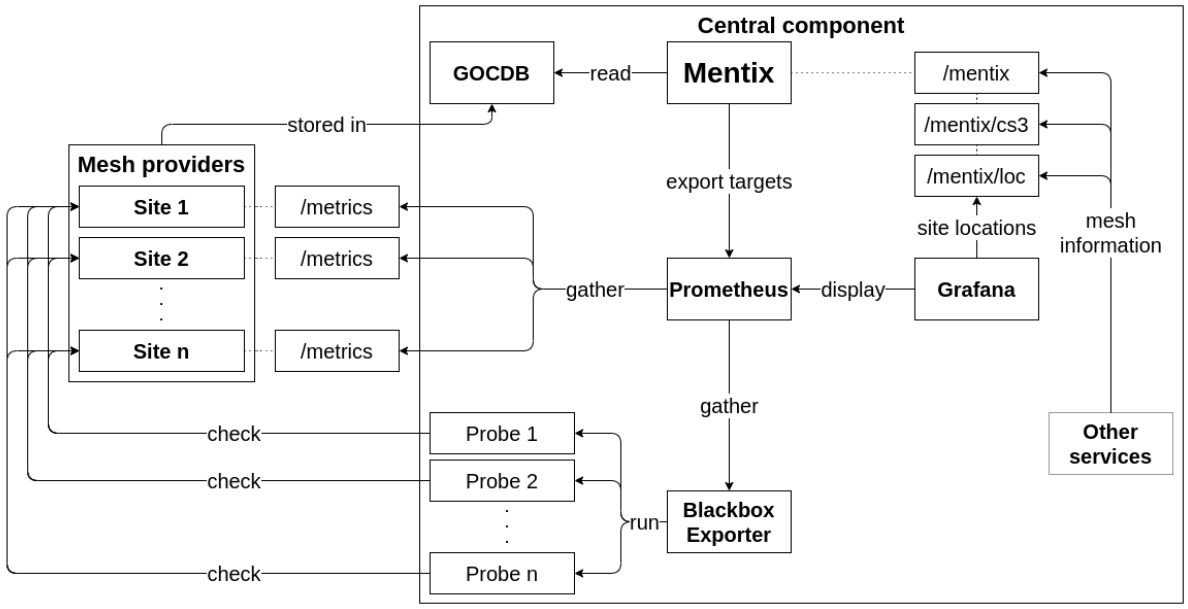


Figure 2 Architecture of site monitoring infrastructure

A major task of the central component is to gather metrics and health information from all sites and their services. Initially, ARGO[7], a monitoring system written and maintained by EGI, was considered for this. The reason for that being that we wanted to leverage already existing site monitoring software and not develop such software ourselves. We have investigated ARGO carefully and we found that the functionality offered by ARGO is far too rich for our purposes and a much simpler solution which is easier to setup would be more suitable for our purposes, at least for now and in order to get us started quickly.

A central Prometheus instance is now used to perform this critical task. Since Prometheus is offered as an open source on-premise solution, it supports all features necessary for the Science Mesh project and can be fully automated. It has demonstrated to completely fit the Project's needs in this area.

As can be seen in the above diagram, the central Prometheus instance is automatically configured by Mentix: whenever the Mesh's topology has been modified, all targets which are to be monitored are updated as well. The gathered metrics currently include the total number of users and groups in the underlying EFSS system, the total amount of used storage and various debugging information. The following screenshot shows an example of the data collected by Prometheus:

[7] https://argo.egi.eu/

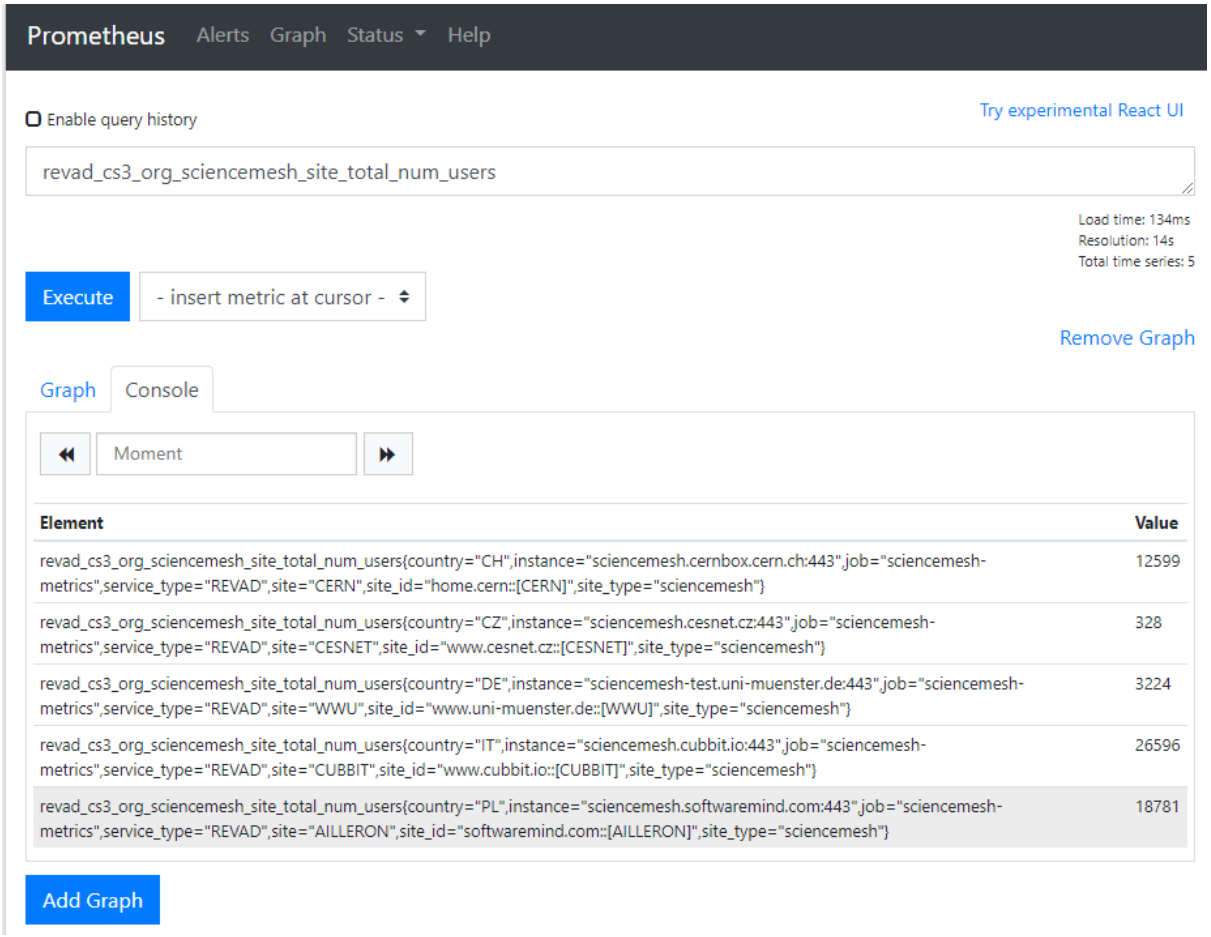
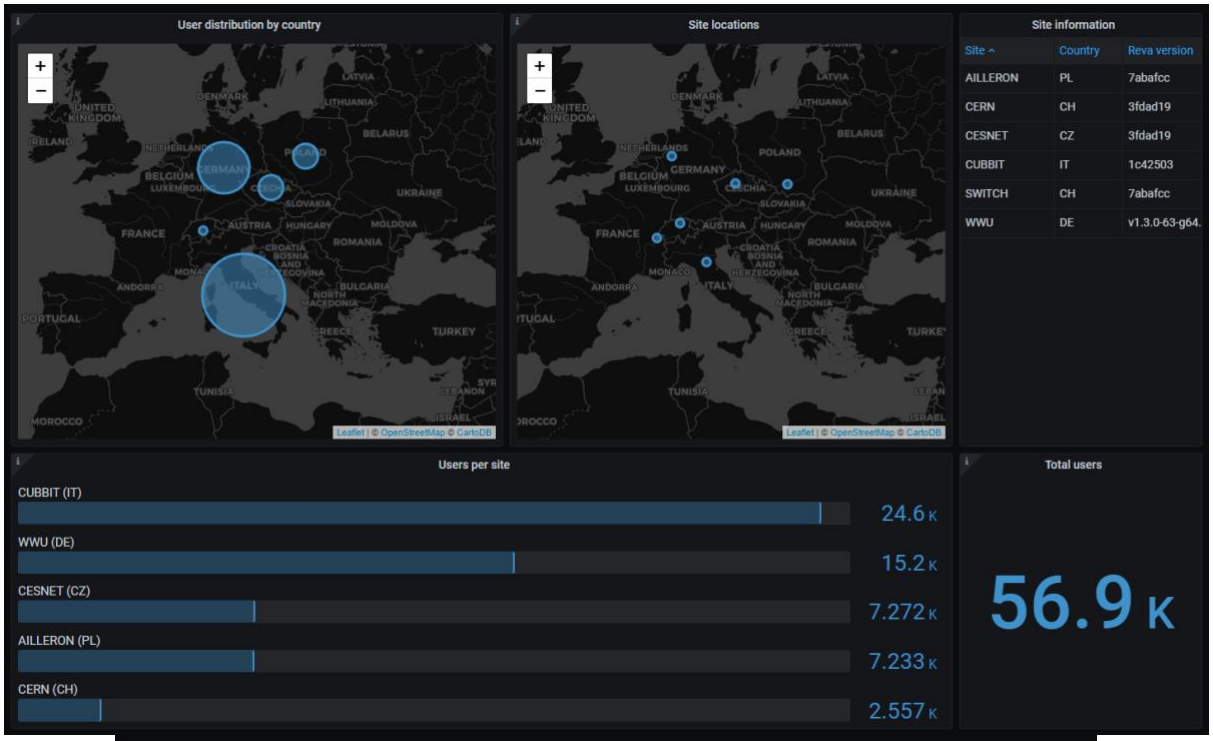


Figure 3 Screenshot of Prometheus

Prometheus is also used for collecting health information about all services running on each site. This is achieved through a service called Blackbox Exporter, which also runs in the central component and is, just like Prometheus, automatically configured by Mentix. The Blackbox Exporter periodically runs various so-called *probes* for every service, which check them for proper functionality and expose the results in the form of metrics consumable by Prometheus. This information is in turn used to calculate an overall site status which is a combination of the individual probes' results, as well as an *availability* rating (i.e., the percentage of time that a system has been in a healthy state).

While that is not yet implemented, Prometheus will also be used to notify administrative users about any service issues reported through the health check probes. This also includes alerts about sites going from a healthy status into either warning or even critical state, ensuring that administrators can react swiftly to any arising problems. Furthermore, a Prometheus instance will also be deployed locally on each site in the future, which will then gather all metrics from every service running on it. In return, the central Prometheus instance can gather these bundled metrics, greatly reducing the overall load on the central instance.

All this rather technical information is presented to the user through various Grafana dashboards. In the first of the following screenshots, a collection of different metrics can be seen. The second screenshot shows a simple site health status overview.



Health status

Site	Country	Status
AILLERON	PL	Healthy
CERN	CH	Healthy
CESNET	CZ	Healthy
CUBBIT	IT	Healthy
SURFSARA	NL	Critical
SWITCH	CH	Warning
WWU	DE	Healthy

Figure 4 Overview of Grafana dashboard

4.2.2 Security

Task 2.5 of the Project is currently ongoing and will provide an assessment of security aspects related to federating EFSS services in the Science Mesh. The first year has been devoted to researching documents related to these aspects writing in the context of other federated e-infrastructures. In

particular, the activities going on within the WISE[8] community have been investigated.

4.2.3 Accounting

A rather crude accounting system has been developed. Sites may operate different EFSS solutions so it is not feasible to come up with a generic solution for all EFSS systems. Therefore, we have chosen the approach to setup an HTTP-based service, using the IOP middleware that reads data from a file which in turn needs to be put in place by a script running at a site. The HTTP service exposes the accounting data to the Prometheus service, which queries for it at regular intervals.

Currently, only the number of users, groups and the amount of storage used are published but this can be extended with more metrics.

4.3 Science Mesh Governance

Besides the technical aspects of Science Mesh's establishment and operations, administrative and governance procedures must also be defined. The goal is to make the governance structure as simple as possible and to limit bureaucracy to the necessary minimum. There must nevertheless be a mechanism to reach agreement in the community (while expecting not all Science Mesh sites are necessarily active all the time in the decision processes), mechanisms to "ingest" new sites as well as remove them from the infrastructure, either of their own will, or in the unfortunate case they repeatedly significantly violate the Mesh's policies. The sites will be requested to refer to Privacy Policy of the Science Mesh in their Terms and Conditions, and to refer to a general description of personal data handling in the mesh.
The governance structure is currently at a proposal[9] state and has been disseminated for Project-wide discussion. The first reactions appeared to be positive, but it should be kept in mind that they refer to the Mesh in its current state and there may be quite significant changes in the future, nonetheless.
We expect the Science Mesh to be defined in a Constitution describing the purpose of the infrastructure, governing bodies, requirements for the sites to join the mesh, joining and leaving the Mesh, resolving violations, dispute resolution procedures and procedures to update the basic Science Mesh documents. A Site shall apply to join the mesh by signing a Declaration describing basic Site obligations, and by satisfying technical requirements (demonstrating compatibility).
Proposed governing bodies are:

- **Science Mesh Executive Board (SMEB)**, which will represent organisations financing the operation of the central components of the Science Mesh. The SMEB will be able to ratify or veto the most important decisions of the consortium.
- **Science Mesh Steering Group (SMSG)**, which will consist of Sites' representatives and will take important decisions (based on a simple majority of active members).
- **Operational Team**, which will operate the central infrastructure on a day-to-day basis, review new sites and provide support.

4.4 New Sites

To ensure operational excellence of the Science Mesh, each participating site will be required to fulfil certain technical requirements which shall ensure smooth operation and high availability of all of its services. An early draft of these requirements can be found in the Operational Excellence and New

[8] https://wiki.geant.org/display/WISE/WISE+Home
[9] https://surfdrive.surf.nl/files/index.php/s/uIMPYMFXj91ECWj

Sites draft[10]. This document, at its current stage, aims at gathering initial ideas for what will eventually become an OLA (Operational Level Agreement) and/or SLA (Service Level Agreement) each site will have to comply with. All requirements are meant to be as transparent and unrestrictive as possible while being as stringent as necessary. The general aim is to make becoming (and staying) part of the Science Mesh as easy as possible without risking a drop in its overall QoS.

To join (and to stay in) the Science Mesh, a site will be required to run various essential services (which are yet to be determined at this point of the Project) that are necessary to operate within the mesh. While each site is responsible for its own administration, it must provide technical contact information which can be used by the central administrative body to reach out to the site in case of technical questions or problems. Furthermore, every site will need to offer extensive support in the form of a helpdesk which end users can use to seek for help.
Proper operation of the Science Mesh will be ensured by constantly monitoring and checking every site and its services externally; this is performed through the central component as described in section 4.2. This also means that a site may only join the Science Mesh if all services are running correctly, which will be verified upfront by the central administrative body during a test phase before a site is added to the Science Mesh.

5 IOP Deployments

In table 1 below we list the current IOP deployments. The deployments of AARNET are underway and those of DTU and JRC are being worked on.

Partner	*URL*	*GRPC Port*	*HTTP Port*
Ailleron	https://sciencemesh.softwaremind.com/iop	443	443
CERN	https://sciencemesh.cernbox.cern.ch/iop	443	443
CESNET	https://sciencemesh.cesnet.cz/iop	443	443
Cubbit	https://sciencemesh.cubbit.io/		443
SURF	https://app.cs3mesh-iop.k8s.surfsara.nl/iop	443	443
SWITCH	https://sciencemesh-test.switch.ch/iop	443	443
WWU	https://sciencemesh-test.uni-muenster.de/api	9600	443

Table 1 The current IOP deployments

[10] https://surfdrive.surf.nl/files/index.php/s/Yg6Qn1zskPO6UKH

6 EOSC integration

6.1 Shaping EOSC

Within the course of the CS3MESH4EOSC project and starting from an already existing infrastructure based on EFSS systems focussed on data sharing, a new and more enhanced infrastructure called the Science Mesh will emerge which will allow not only the sharing of data but applications as well.

According to the EOSC vision, the European Open Science Cloud will offer European researchers a virtual environment with open and seamless services for storage, management, analysis and re-use of research data across borders and scientific disciplines by federating existing scientific data infrastructures which are currently dispersed across disciplines and EU member states. It is therefore obvious that the Science Mesh will significantly contribute to turning this vision into reality. For this reason, integrating the Science Mesh into the EOSC is a task of its own within the CS3MESH4EOSC project (T2.4).

It was stated in the proposal that the CS3MESH4EOSC project would seek joining the EOSC-hub infrastructure. Since the CS3MESH4EOSC project is currently set to outlive the EOSC-hub project which ends in 2020, we found it more appropriate to seek integration with EOSC at a broader level, while never excluding EOSChub or any other future initiative within the same field.

In the meantime many discussions have taken place on what the EOSC is supposed to look like and things seems to be still in a “state of flux”. This means that, when it comes to being part of EOSC, the CS3MESH4EOSC project will need to monitor future developments closely.

It is beyond the scope of this document to give a complete overview of the work that has been done so far by the various working groups and boards on shaping the EOSC, but it is good to give a small recount of the EOSC's structure that is currently envisioned.

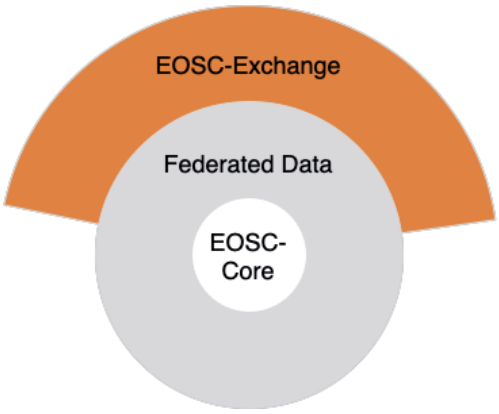


Figure 5 Schematic overview of EOSC

Figure 5 shows EOSC as it is currently envisioned: EOSC-Exchange constitutes the user-facing services of EOSC, allowing users to exploit FAIR data. In addition, there is EOSC core offering central services that are required to operate the e-infrastructure. Among these services there are accounting and monitoring, AAI, a registry of services, a helpdesk and policies for onboarding new service providers.

At the time of writing it was envisaged that EOSC would not consist of one monolithic core but a collection of different cores (e-infrastructures) that would be interoperable in some way.

Several working groups are currently actively working to shape this vision. They are working on the architecture, sustainability, mapping the landscape of existing e-infrastructures that are supposed to be part of the EOSC federation and other topics. More information on this can be found on EOSC Secretariat webpage[11]. One of these working groups is the *Rules of Participation Working Group* whose work will be very relevant for CS3MESH4EOSC.

6.2 Rules of Participation

The EOSC possesses a set of "rules of participation" (RoP) put together by the *RoP Working Group* that resource providers have to fulfil in order to be eligible to be included in one of the EOSC service catalogues. These rules form the foundation of policies, procedures and processes that assures the openness, quality and trust in the services and practices offered by participating in EOSC.

These rules of participation have evolved over the last two years in an iterative process. At the time of writing the current version is v0.5[12]. which was released on October 20th 2020. The final version v1.0 is expected to arrive in December 2020.

The past few months we have examined version v0.2[13] (January 29th 2020) and tried to assess whether the Science Mesh as envisioned by CS3MESH4EOSC would be eligible to enter the EOSC and what requirements the it would have to fulfil in order to do this. In the meantime, there have been some changes between v0.2 and v0.5, which need to be examined more closely in order to have an update on the estimated impact. Since this section is about the work done so far, we will describe the analysis of the rules of participation stated in v0.2 of the document.

The RoP consists of the following sections: Ground Rules, Data, Services and EOSC Operators. Below we will describe these sets of rules.

Ground rules

> *G1 EOSC is open to all.*
> *G2 EOSC resources are registered in an EOSC catalogue*

This means that EOSC is as open as possible and as closed as necessary. Resource providers may impose additional constraints as long as they comply with the rules regarding data and services.

Data

> *D1 Data resources exposed through EOSC are free at the point of access.*
> *D2 Data producers adhere to the principles of proper research conduct*
> *D3 Data producers determine the terms of use of data resources.*
> *D4 Data providers respect the principle of FAIR data*
> *D5 Data users adhere to the terms of data resources*

[11] https://www.eoscsecretariat.eu/eosc-working-groups
[12] https://www.eoscsecretariat.eu/sites/default/files/draft_eosc_rop_version_0.5_20-10-2020.pdf
[13] https://repository.eoscsecretariat.eu/index.php/s/QWd7tZ7xSWJsesn#pdfviewer

D6 Data users reference the source

Since the CS3MESH4EOSC project and its participants are not the owners of the data, this set of requirements does not directly affect them. However, if data owners are expected to adhere to the principles of FAIR data then the CS3MESH4EOSC project should provide them the means to do so. This is provided by task 4.2.

Services

S1 Services exposed through EOSC are free at the point of access
S2 Service providers adhere to the principles of proper research conduct
S3 Service providers determine and publish the conditions of use of their services
S4 Services align with the EOSC services architecture
S5 Service users adhere to the terms of use of the services they consume
S6 Service users reference the resource

Rule S1 does not necessarily mean that everything should be free. Service providers may ask for compensation for the usage of the resources they provide.

It remains to be seen as how the rules S2 and S6 are to be implemented within the CS3MESH4EOSC project, since this seems to apply to services provided by academic institutions involved with European e-Infrastructure projects focussed on scientific communities.

Rule S4 is where our attention should be focussed on. Since the work done by the *Architecture Working Group* involves AAI and a PID infrastructure.

Rules for EOSC operators

Op1 Registry of data and service catalogues
Op2 Onboarding data and services
Op3 Monitoring and Accounting
Op4 Authentication and Authorisation
Op5 Search function and other global functions
Op6 API and value-add providers

Most of these requirements are covered by CS3MESH4EOSC right out of the box. The RoP states more specifically that EOSC operators should support procedures that enable authentication and authorisation based on academic credentials for federated AAI.

The EOSC AAI is envisioned to be based on the AARC BluePrint Architecture. The idea behind this is that, in order to be able to federate resources, AAI has to be federated as well. Trust comes from communities and community AAI forms the basis of AAI within EOSC. Users are able to access resources through community AAI using their institutional credentials used in national identity federations within eduGAIN, or a community managed identity provider, or some other identity providers allowed by the community. As we explained in section 4.1, our approach, while taking some shortcuts to be able to deliver the infrastructure quickly, is fully compatible with this vision of EOSC AAI.

6.3 How to join EOSC

There are basically two ways to join EOSC. One is as a **resource provider** and the other as an **e-infrastructure**. At the time of writing, it is possible to join EOSC through projects like EOSC-hub or EOSC-Enhance, both having their requirements[14] for allowing a service provider to become part of EOSC. These requirements are nevertheless quite different from what the *RoP working group* has put forward which indicates that things are still very much in a state of flux. It follows that these developments must be closely monitored and the project will have to make adjustments as necessary. In any case it is conceivable that all EFSS sites join EOSC as service providers separately.

Joining as an e-infrastructure is the other route to EOSC. Since the CS3MESH4EOSC project already contains a number of central services that would be part of EOSC core, it seems at the moment that it is appropriate for CS3MESH4EOSC to join EOSC as a separate e-infrastructure.

7 Conclusion and Future Work

This report summarises the work of **Work Package 2** in the **first year** of the Project. The overall architecture of the Science Mesh was defined. Both **user-facing** as well as **participating site-facing** procedures took shape. The former is demonstrated in the form of use cases and procedures to achieve sharing resources in the mesh, be it files and folders, file transfers, and/or access to applications. The most significant use case was illustrated in the form of a demo.

The **internal structure** of the Science Mesh ranges from technical ideas on how to provide **monitoring**, **accounting**, **help desk**, the organisation **of user support** as well **as support for sites who are joining** or have joined the Science Mesh and other necessary functionalities, to the first proposal of Science Mesh **governance structure** and **formal establishment**, including procedures for the sites to **join and to cooperate on mesh development**.

This is obviously **not to say** that the effort is **finished**. The AAI scenarios are being implemented and, at the time of writing, are available in the form of command-line tools. In order for them to be useful to regular users, graphical user interfaces and integration with the deployed EFSSs will be necessary, which is the ensuing task in the development of the **Reva IOP system**.

In order to operate the infrastructure, ensuring its **operational health** is a necessity. **Prometheus** will be used to notify site administrators about any **service issues** reported through the **health check** probes, ensuring that administrators can **react swiftly to any arising problems**. Furthermore, a hierarchy of **Prometheus instances** will also be deployed from site-local ones to a central instance gathering high level information to **ensure scalability**.

It is essential that the security of the system be flawless. All partners joining the Science Mesh must be confident that they can trust its components to be secure and not expose their origin system to new threats. To ensure that all the newly developed parts of the Project are state of the art security-wise, a penetration test is planned for the first half of 2021. The results and findings of that exercise will be detailed in deliverable D2.4 for M24. The penetration test will look at aspects like Privilege Escalation, SQL Injection, Error Handling, etc.

With respect to the security aspects of a federated EFSS infrastructure, an assessment will be performed and reported on. At the midterm review (M18) we expect to have a draft document about

[14] https://wiki.eosc-hub.eu/display/EOSC/Criteria+for+possible+inclusion+in+the+EOSC+Service+Portfolio

it. In addition, security requirements of related to joining the Science Mesh will be investigated.

As far as integration with EOSC goes, we expect to produce a plan in the coming months on how the Science Mesh is going to join the EOSC, and which aspects will require attention. For that, a number of items will have to be clarified, and greater investigative effort is needed. For example, the current setup as described in this document and as shown in the demo deals with the scenario where existing users of EFSS services which joined the Science Mesh can share resources amongst themselves. Another question that needs to be answered is how users who do not yet have access to EFSS services can have access to this new infrastructure. Nevertheless, we are confident that the Science Mesh with its ever growing several hundred thousand users will form a major contribution to EOSC.