D5.6 Final report on maintenance and integration of federation and collaboration services

Lead Partner:	KIT
Version:	1
Status:	FINAL
Dissemination Level:	Public
Document Link:	https://documents.egi.eu/document/3636

Deliverable Abstract
The document outlines the final report on maintenance and integration of federation, access enabling and collaboration services, one of the key components of the EOSC Federating Core, a fundamental asset that EOSC-hub provides to EOSC. It provides a technical description of enhancements for the EOSC-Hub services in Work Package 5 (WP5), results of integration activities, and collaboration work with other initiatives made during the third year of the project. In addition, it provides changes or enhancements made during the third year of the project in the form of release notes presented in a uniform format. The report identifies the integration gaps and elaborates the recommendations of future EOSC initiatives.

COPYRIGHT NOTICE

DELIVERY SLIP

Date	Name	Partner/Activity	Date
From:	Nicolas Liampotis	GRNET	29.03.2021
	Kostas Koumantaros	GRNET	
	Themis Zamani	GRNET	
	Roksana Rozanska	CYFRONET	
	Cyril L'orphelin	IN2P3	
	Pavel Weber	KIT	
	Marcus Hardt	KIT	
	Alexandros Nakos	IASA	
	Ivan Diaz Alvarez	CESGA	
	Jens Jensen	STFC	
	Greg Corbett	STFC	
	Raphael Ritz	MPG	
	Daniel Kouril	CESNET	
	Christos Kanellopoulos	GEANT	
	Sander Apweiler	Julich	
	Mischa Sallé	Nikhef	
	Slavik Licehammer	CESNET	
	Enrico Vianello	INFN	
Moderated by:	Malgorzata Krakowian	EGI Foundation/WP1	
Reviewed by:	John Kennedy	MPCDF	
	Paolo Manghi	ISTI-CNR	
Approved by:	AMB		

DOCUMENT LOG

Issue	Date	Comment	Author
v.0.1	22.10.2020	Table of Content ready	Pavel Weber
v.0.2	28.10.2020	The first draft with sections ready	WP5 service/tool owners
v.0.3	28.12.2020	All contributions are provided	Pavel Weber
v.0.4	05.01.2021	Added executive summer	Pavel Weber
v.0.5	28.02.2021	Ready for review	Pavel Weber
v.0.6	13.03.2021	Review	John Kennedy Paolo Manghi
v.1	29.03.2021	Final	Pavel Weber

TERMINOLOGY

Terminology/Acronym	Definition
AAI	Authorization and Authentication Infrastructure
AARC	Authentication and Authorisation for Research and Collaboration
AC	Attribute Certificate
AppDB	Applications Database
AppDB IS	AppDB Information Service
AppDB VMOps	AppDB VM Operations
AUP	Acceptable Use Policies
BDII	Berkeley Database Information Index
CA	Certification Authority
CDI	Collaborative Data Infrastructure
CMDB	Configuration Management Database
DPMT	Data Project Management Tool
EGI	European Grid Infrastructure
EOSC	European Open Science Cloud
EUDAT	European Data Infrastructure
GDPR	EU General Data Protection Regulation
GGUS	Global Grid User Support
GOCDB	Grid Operations Configuration Management Database
HA	High Availability
IAM	Identity and Access Management System
IdP	Identity Provider
IGTF	Interoperable Global Trust Federation
LB	Load Balancing
LoA	Level of Assurance
OIDC	OpenID Connect
OLA	Operational Level Agreement
PKIX	Public-Key Infrastructure (X.509)
SDT	Service Description Template
SLA	Service Level Agreement
StaR	Storage Accounting Record
SOCRM	Service Order and Customer Relationship Management
PID	Persistent Identifier
SP	Service Provider

SMS	Service Management System
SAML	Security Assertion Markup Language
SSM	Secure STOMP Messenger
TRL	Technology Readiness Level
VM	Virtual Machine
VO	Virtual Organisation
VOMS	Virtual Organization Membership Service

Contents

Executive summary

The major federation and collaboration services in WP5 are considered as candidates to form the main building blocks of the EOSC-Core. Offering a set of capabilities such as resource discovery, access, ordering, monitoring, accounting, user support etc. they contribute to the core functions of EOSC, its stable and distributed operation as well as facilitate the onboarding and integration of thematic and researcher-facing resources delivered by various communities in the EOSC. The federation and collaboration services are the components of the EOSC-hub Key Exploitable Results (KER) "Internal Services Provided in the Hub Portfolio" and "EOSC Portal and Marketplace".

During the last reporting period of EOSC-hub project the work carried out in Work Package 5 was focused on further implementation of the roadmap established during the initial period, addressing the recommendations from the first project review and fulfilment of the requirements of other EOSC-hub work packages as well as requirements of research communities associated with EOSC-hub. Work Package 5 followed the work and recommendations regarding the evolution of the EOSC architecture delivered by EOSC Sustainability Working Group and EOSC Architecture Working Group shaping the initial subset of federation services which should contribute to the beta version of the EOSC-Core.

Significant focus was given to the improvement of interoperability and integration of the federation services with EOSC Portal. This work in the last period has been performed in the scope of the cross-project EOSC Portal Collaboration Agreement between the EOSC-hub, OpenAIRE-Advance and EOSC Enhance projects. One of the major outcomes of this endeavour was the full integration of the EOSC-hub Marketplace in the EOSC Portal, establishing a single unified catalogue.

Work Package 5 cooperated with multiple EOSC-hub work packages actively supporting the integration work which was carried on for common and thematic services in WP6 especially for AAI. In addition, WP5 significantly contributed to the establishment of the EOSC-hub Configuration Management Plan governed by WP4 and supported smooth running of several major processes established in the scope of EOSC-hub Service Management System, like EOSC-hub order management process, incident, and request management process, change management.

The federation and collaboration services have undergone many enhancements, improvements of interoperability and interfaces. The major achievements per WP5 tasks can be summarised as follows:

- **Identification, Authentication, Authorisation and Attribute Management:** The activities carried out within this task focused on the delivery of an integrated EOSC-hub AAI, enabling Service Providers to offer their resources to research communities and individual researchers, allowing users to use their institutional and community-based digital identities. The task has demonstrated technical ability of research communities to access resources using different Community AAIs (B2ACCESS, Check-in, eduTEAMS, INDIGO IAM). The harmonisation of the user and community information as well as the alignment of the multiple policies like Acceptable Use Policies (AUPs) across different e-infrastructures has been achieved. The white labelled AAI service (EOSC Portal AAI) for enabling individual researchers and research communities to access the EOSC Portal has been deployed. Major

Community AAI and Infrastructure Proxy services have been interconnected. The workflows for users to access the resources have been significantly simplified.

● **Marketplace and Order Management Tools:** During the course of the project a simple Marketplace service with a limited catalogue of the services has undergone a remarkable evolution to the powerful interoperable platform with multiple user-facing functionalities like discovery, resource filtering, order management. It has been integrated with multiple services and systems like EOSC Portal, EOSC Resource Registry, Provider Component and Service Order Management Back Office and now supports complex workflows and business processes like distributed order management as well as the onboarding of new EOSC resources. The finalized development of a White label Marketplace solution provides the possibility for communities and other interested parties to create their own branded Marketplaces based on the EOSC-hub Marketplace solution and integrate them with the EOSC Marketplace. This approach provides significant benefits for small communities with limited resources to develop their own catalogues and Marketplaces and facilities the building of EOSC as a federated system.

● **Integrated Business and Operations Support Systems:** The activities carried out in this task were focused on several services which support the daily operations of the EOSC. A remarkable amount of progress has been made within the Operations Portal. A functional component of Operations Portal - Service Order Management Back Office (SOMBO) has been developed from scratch and successfully deployed in production as a part of the Order Management System according to developed in the course of the project order management concept and established roadmap for Order Management System. SOMBO has been integrated with the Marketplace to receive, manage, and dispatch orders to multiple service providers. A new module has been put in place to collect metrics about orders in order to generate reports for the European Commission. A new instance of the Operations Portal in the EOSC scope has been deployed. The interoperability of the other services such as GOCDB, DPMT, DMP in the task has also been enhanced. A scalable service data model and configuration plan have been developed in cooperation with WP4 to facilitate the foundation of the EOSC Configuration Management System in the future EOSC projects.

● **Monitoring, Accounting, Messaging and Security Tools:** A unified web-portal that combines services from a number of different providers/infrastructures involved in EOSC has been developed and taken into production. The ARGO Monitoring has been significantly enhanced and managed to achieve its goal of establishing a One Stop Shop that simplifies and automates the operation and configuration of ARGO components. The enhancement of the ARGO Web API drastically simplified the integration with external and internal sources of information and allowed it to handle a plethora of new metrics, profiles and topologies coming from different sources and evolve to act as an aggregator of Monitoring data coming from external sources. The dynamic development of ARGO Messaging Service (AMS) led AMS to play the role of the transport layer for the secure exchange of information between the services (e.g., Accounting, Monitoring, AAI) and will allow AMS to play an important role in EOSC-Core as the transport layer between EOSC-Core components and EOSC-Exchange providers that are willing to push Monitoring and Accounting data to EOSC-Core.

- **Helpdesk Services and Tools:** A unified EOSC-hub Helpdesk has been established together with a multi-level structure of the support units. The helpdesk system has been fully integrated with EGI and EUDAT helpdesk systems as well as with EOSC Portal. The helpdesk has successfully supported the incident and request management process during the project runtime.

- **Application Store, Software Repositories, and other Collaboration Tools:** The focus of the activities for this task have been on improving the quality of the services provided to the community. The Applications Database has delivered its initial goals of an enhanced service, without any deviations from the roadmap, in particular, some notable major achievements are the deployment of two new dashboards which enhance the quality of offered VM services, the migration to the GLUE2.1 cloud information schema which provides better support for VM operations on cloud sites, and the introduction of a new information system which provides a consolidated view of the FedCloud infrastructure, both to AppDB components and to external services alike, though it RESTful API.

A detailed list of enhancements and summary with assessment of the progress is given in the dedicated summary section for each task in this document. In the summary section of this document, we address the major challenges we faced during the course of the project and provide an outlook for the future.

1 Introduction

This document provides a final report on maintenance and integration of federation and collaboration services in the EOSC-hub project. It summarises the activities of Work Package 5 (WP5) for the third reporting period according to the roadmap reported in the previous deliverable D5.5 and also provides the summary and assessment of overall progress and achievements for each task in WP5 for the whole project.

The structure of the document follows the structure of WP5 which is itself organised in 6 tasks. Chapter 2 provides a short description of the concrete implementation of the subset of federation services considered as candidates to the EOSC-Core. Each task is described in the corresponding chapter which provides a summary of the major enhancements for each service followed by results of integration activities followed by the task summary for the whole project time. The general summary for WP5 is given at the end of the document followed by the Appendix with a list of the info cards for all services in WP5 including the changelog.

2 EOSC-hub contribution to the EOSC Core establishment

The EOSC-hub federation and collaboration services, that have been further developed and integrated by Work Package 5, will contribute with their capabilities to the core functions of the EOSC-Core.

Following the EC recommendations to focus on the integration of front-end and back-end federation services with EOSC Portal the initial integration roadmap was modified accordingly and pursued an enhancement of the federation services and their tight integration with the EOSC Portal in order to support the EOSC Portal core functionality.

This chapter presents a subset of the federation services and their instances running in production which have been deployed during the second half of the project. This subset as shown in Table 1 contributes to the beta version or initial implementation of the EOSC-Core.

Table 2-1 Subset of the service candidates to the EOSC-Core

Service	Instance URL	Description
EOSC Portal AAI	https://aai.eosc-portal.eu/proxy	The EOSC Portal AAI enables researchers in Europe to use their Research Community AAI or academic/social account of choice including eduGAIN for access and authorisation to the services and resources offered through the EOSC Portal.
EOSC Helpdesk	https://helpdesk.eosc-portal.eu/	EOSC Helpdesk provides a unified interface for the users of all the different infrastructures integrated on EOSC, facilitating their access to a support unit, and providing a unified system to store, classify and escalate the incidents and problems.
EOSC Marketplace	https://marketplace.eosc-portal.eu	The Marketplace (MP) is a user-facing platform where productional EOSC services can be promoted, discovered, ordered, and accessed. The MP is a component of the EOSC Portal and is integrated

		with the EOSC Portal Provider component.
EOSC Operations Portal	**https://opsportal.eosc-portal.eu**	The service provides Service Order Management Back Office (SOMBO) for management of EOSC orders submitted via EOSC Marketplace
EOSC Monitoring	**https://argo.eosc-portal.eu**	A unified web-portal that combines services from a number of different providers/infrastructures involved in EOSC.
EOSC Configuration Repository	**https://gocdb.eosc-portal.eu**	EOSC registry to record information about the topology of an EOSC infrastructure. It provides the information about EOSC-Core services, including their providers, service-endpoints, information about downtimes, contact information and roles of users responsible for operations at different levels.

By consolidating the services presented in Table 2-1 under the EOSC Portal domain, we have started to shape the EOSC-Core as a subset of the instances of federation services which are tightly integrated with EOSC Portal. These services enhance the EOSC Portal functionality and provide support at the operational level.

3 Identification, Authentication, Authorisation and Attribute Management

3.1 Overview

The EOSC-hub AAI is key to enabling access to research data and services in a secure and user-friendly way. During the last reporting period, the task placed focus on finalising the integration of the AAI services delivered by EGI, EUDAT, GÉANT and Indigo IAM. This integration involved further harmonisation across the AAI services, both at the technical and the policy level, including the alignment of user attributes and the adoption of guidelines (see also Section 3.10.3 in D5.3 [R8]). The result was a significant improvement of the user experience and simplified access to the underlying services and resources using community identities which act as the interface between individual users and the resources.

The remainder of this section elaborates on the enhancements made to the EOSC-hub AAI services and the activities required to achieve integration.

3.2 B2ACCESS

A detailed description of the B2ACCESS service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

3.2.1 Maintenance activities

In the reporting period the underlying software, Unity, was updated multiple times and is running on latest version 3.3.4. Other used packages were kept up to date, too. During the Unity update a short downtime (a few minutes) was needed to switch to the new release and update the internal database. Besides these planned and announced downtime, no further downtimes were required. Furthermore, the certificate infrastructure of B2ACCESS was renewed including a new certificate authority which was also integrated into the federation as a trusted certificate authority.

3.2.2 Summary of service enhancements

In the last year, several small enhancements for the users have been made. The acceptable use policy (AUP)/terms of use (ToU) were updated to the harmonized policy with other infrastructures. This allows skipping the additional consent of users, authenticating with Check-in or eduTeams, which uses the same policy. INDIGO-IAM has been added as an additional identity provider so that users with an account in INDIGO-IAM can use services connected to B2ACCESS.

3.2.3 Future plans

In future the validation of OAuth2 token, granted by other infrastructure like Check-in or eduTEAMS, should be validated through B2ACCESS to grant access to services which are connected to B2ACCESS. This offers more flexibility and easier service usage across different infrastructures for the users. Furthermore, the underlying software stack will be updated.

3.3 Check-in

A detailed description of the Check-in service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

3.3.1 Maintenance activities

The production operation of the EGI Check-in service involved technological upgrades of the underlying framework and libraries in order to take advantage of new features and robustness, as well as continuous optimisation of the architecture and automation of new tasks to ensure the uninterrupted and performant operation. In this context, the development and production instance of Check-in was moved to GRNET's ~okeanos-knossos data centre to improve performance. The migration lasted 30 minutes, during which the user and group enrollment functionality was not available. Internal (fabric) monitoring was also improved.

3.3.2 Summary of service enhancements

3.3.2.1 Enhancements to user & VO enrollments flows

The different enrollment flows provided by Check-in were improved as follows:

1. User enrollment flow:

    a. When a new user is trying to access a service through Check-in, they get automatically redirected to the Check-in user membership registry (COmanage Registry) to sign up. After completing the signup process (acceptance of AUP and email validation) the user ends up in COmanage instead of getting redirected to the original service they were trying to access. This change enabled the automatic redirection of the newly registered user to the original service they were trying to access.

    b. Email verification can be skipped during user enrollment with Check-in if the user's authenticating IdP releases an already verified email address. This can be determined based on the availability of specific attributes:

        i. The `voPersonVerifiedEmail` attribute in the case of SAML IdPs

        ii. The `voperson_verified_email` claim in the case of OIDC IdPs

        iii. The `email` claim combined with an `email_verified` claim set to `true` in the case of OIDC IdPs

    It is also possible to configure Check-in to explicitly skip email verification for specific IdPs. This is intended to be used for those IdPs that are known to require email verification but lack support for expressing the verification status of the released mail information through one of the aforementioned attributes (e.g., EGI SSO IdP).

    c. Fixed a bug that prevented users from signing up with Check-in when their attribute assertion contained multiple certificate subject DNs.

2. <u>VO enrollment flow:</u> Users need to be registered with Check-in before being able to request membership/join VOs. Instead of showing a generic error about not being able to submit the request, the VO enrollment flows were improved to provide a more helpful message that also includes a link to the signup flow.

3. <u>User/VO enrollments flows:</u>

    a. When users request to sign up or join VOs they are presented with a circular progress indicator to visualise the status of their enrollment request.

    b. Menu options that were confusing for some users were removed and the look and feel of the different enrollment steps was improved.

    c. Added an EnrollerPlugin [R1] to allow Check-in to check certain attributes of a registered user's identity before allowing that user to request membership in a VO.

4. <u>Explicit identity linking flow:</u> This flow requires the user to authenticate first with any of the identities already linked to their Check-in identity, and then to re-authenticate using the login credentials of the additional identity they want to connect. Previously the flow required the user to follow a link in an email sent after requesting to link a new identity. This step was removed to make the linking process simpler and faster.

3.3.2.2 Enhancements to GOCDB role plugin

Check-in supports retrieving role information from the GOCDB. This change improved the GOCDB integration as follows:

1. The Check-in GOCDB plugin was configured to fail over to a backup instance of the GOCDB if the primary one fails to return role information

2. If role information cannot be retrieved from the GOCDB, Check-in warns the user that role information is not currently available and provides an option to continue accessing the service without role information.

3.3.2.3 Enhancements to Community AAI flows

This change simplified access to resources protected by Check-in for users authenticating via Community AAIs (e.g., eduTEAMS, DEEP Hybrid DataCloud). Specifically:

● Users are not required to validate their email address given that this is already verified by the Community AAI

● Check-in passes onto the infrastructure services the user's Community identifier as released by the Community AAI

● Check-in can be configured to map the entitlement values released by the Community AAI to entitlement values that can be used by downstream services for authorising access to their resources

3.3.2.4   Enhancements to usage statistics viewer

A usage statistics viewer plugin [R2] was added to the Check-in user membership registry (COmanage Registry). The plugin provides anonymised usage statistics for:

- Total number of user logins

- Number of user logins from identity providers

- Number of service providers accessed through Check-in

- Number of user registrations

- Number of VOs registered in COmanage Registry, including information about the time of creation and the number of active and suspended users per VO

The plugin allows specifying a group of privileged users with access to detailed statistics and export capabilities (CSV, PDF format supported) for customisable time ranges on daily, weekly, monthly, or yearly basis.

3.3.2.5   Support for service identities

Check-in added support for service identities. Service identities can be used for accessing resources by non-personal identities, i.e., applications/resources (e.g., portals). At the same time, service identities can be associated to one or more personal identities, i.e., the service identity owners.

3.3.2.6   Enhancements to RCauth linking plugin

The RCauth certificate linking plugin for the Check-in Membership Registry was refactored so that it does not rely on modifications to the core code of the COmanage Registry.

3.3.2.7   Enhancement to VOMS (De)Provisioning plugin

The VomsProvisioner plugin [R3] was upgraded to support the SOAP and REST VOMS Admin API v3.7.0. In addition, the new version of the VomsProvisioner plugin provides enhancements to the Plugin's Admin Configuration UI. The new features are:

- Management of the Private Key and Certificate of the Robot user, which will handle the communication with VOMS

- Checking of the Robot User's Certificate validity

- Bulk import of VOMS servers using JSON format

3.3.2.8   Enhancements to SAML SP/OIDC Provider interface

1. Check-in added support for expressing the email verification status in the SAML attribute assertions and the OIDC claims released to relying parties. Specifically:

   ○ In SAML, the verified email address(es) is made available through the multi-valued `voPersonVerifiedEmail` attribute

   ○ In OIDC, the verified email address(es) is made available through the multi-valued `voperson_verified_email` claim. Also, the `email_verified` claim is set to

true/false depending on the verification status of the address in the single-valued email claim

2. The cert_entitlement scope and claim were added to Check-in's OIDC Provider interface. The cert_entitlement claim has been introduced to satisfy DIRAC's requirement for a claim that provides information about the user's certificate subject(s) and the associated VO(s). The structure of the cert_entitlement claim is a JSON array containing multiple objects. Each object contains

   ○ `cert_subject_dn` - single string value expressing the user's certificate subject DN

   ○ `cert_iss` - single string value expressing the user's certificate issuer DN

   ○ `eduperson_entitlement` - one or more string values formatted as URNs expressing the user's VO/group membership information

3.3.2.9   Enhancements to user profile page

The user profile page in the Check-in membership registry (COmanage Registry) was improved as follows:

1. Support for renewing acceptance of updated AUPs: Users have the choice to renew the acceptance of updated AUPs from their user profile at https://aai.egi.eu/registry under "Review Acceptable Use Policy". For each AUP listed on this page (infrastructure-wide or VO-specific), if there is a new version, its status has a message informing the user about the update. The user can agree to the updated AUP by clicking the "Review Acceptable Use Policy" button.

2. Customisable action list for linked identities panel: A new "Actions" drop-down button was introduced on the top right corner of the organisational identities panel at https://aai.egi.eu/registry. The "Actions" button includes two actions:

   a. Link new identity

   b. Link RCauth certificate

   The list of actions is configurable so it can be extended to include additional options relevant to the user's linked identities.

3.3.3   Future plans

3.3.3.1   Proxy certificate retrieval through SSH key information managed in Membership Registry

Users can use SSH key authentication in order to retrieve proxy certificates from the MasterPortal (see Section 3.8). Currently, this requires users to upload the ssh public key via a dedicated self-service portal [R4]. This activity will enable proxy certificate retrieval from the MasterPortal using the SSH keys uploaded by users through the Check-in Membership Registry (COmanage Registry) user profile management page [R5]. It should be noted that the MasterPortal will take into account the SSH keys from both the existing dedicated portal and the COmanage Registry.

3.3.3.2   *Improve identity linking user experience and interface*

Identity linking (also known as account linking) refers to the process of connecting the user's identity generated by Check-in with their external identities, i.e., identities created and assigned by institutional or social media IdPs. The identity linking process allows the user to access resources with a unique identity regardless of their external identity, used for authentication. Check-in currently supports *explicit* linking (see also Section 3.3.2.1), which enables users to request that an additional identity be linked to their existing Check-in identity. This enhancement includes the following activities:

- The linked identities panel shown in the Check-in user management profile page will be improved by including the friendly name & logo of the user's linked identity providers.

- Check-in will add support for *implicit* linking, which will be triggered when one attribute, or a combination of attributes, of one identity correlates to one or more attributes of another identity that is already associated with a registered user. Implicit linking can prevent an individual from accidentally registering distinct Check-in identities. The identity linking considers a proper combination of assurance information, so even if the user's identities have been linked (either explicitly or implicitly), the assurance is determined based on the identity the user is authenticating with (effective identity). This is based on guidelines document AARC-G031 [R7]

3.4   eduTEAMS

A description of the eduTEAMS service is given in D5.3 [R8]. The release notes for the reporting period are provided in Appendix A.

3.4.1   Summary of maintenance and service enhancements

During the reporting period, the eduTEAMS service was upgraded from version 3 to version 4. In addition, regular maintenance of the underlying infrastructure and software components has taken place.

The Proxy component was enhanced with improved support for OIDC flows, including support for remote token introspection. The enhancements allow the full configurability of the claims that are included in id_tokens and access tokens issued by the eduTEAMS OISC Provider. Managing OIDC clients has become easier, as now OIDC clients can be registered by authorized users using OIDC client registration. The support for the subject-id and pairwise-id identifiers has been improved.

The Discovery Service (DS) has been revamped and now it issues a modular architecture separating the user facing frontend from the backend query engine. The new version of the DS has significantly less memory requirements and offers visible improvement in performance. In addition to the DS, the MetaData Query (MDQ) service has been updated. The new implementation decouples the MDQ service, from the DS UI and the DS backend query engine.

The eduTEAMS Account Registry (AR), has improved support for multi-tenant environments and now supports platform-initiated identity linking. New users can now modify their application before being approved. The email validation flow has been enhanced and now the AR can require email

validation when the email of the users is coming from unverified sources and bypass it when it is coming from Identity Providers which provided emails that can be considered validated.

The eduTEAMS User Inform component has a redesigned user interface that improves the usability and overall user experience. Attributes are presented to the user with explanatory names helping the user to better understand, which data is made available to each requesting service. The user interface was updated to provide more information about the requesting services with support for both SAML Service Providers and OIDC clients. In addition to the UI improvements, the backend code received a number of developments improving the overall reliability and performance of the component.

Finally, the eduTEAMS Infrastructure has been significantly improved with support for rolling reboots, improved change management processes and improved in the CI/CD and automated testing for new releases.

3.4.2   Future plans

The following plans are listed in the service roadmap:

- Support for dynamic filtering of Identity Provider in the DS based on the metadata of the IdPs and attribute release test flow.

- Improvements in the registration flow of new users

- REST API for the eduTEAMS AR

- New Policy Enforcement Point that will allow enhanced support for enforcing access policies centrally

- Support for AARC-G057 "Inferring and constructing voPersonExternalAffiliation"

- Support for AARC-G061 "A specification for IdP hinting"

3.5 INDIGO-IAM

A description of the INDIGO IAM service is given in D5.3 [R8]. The release notes for the reporting period are provided in Appendix A.

3.5.1   Summary of maintenance activities and service enhancements

3.5.1.1   *Improved token exchange flexibility*

Token exchange flexibility has been improved with the development of token exchange policies which allow having fine-grained control on the token exchange process. In order to do this, the support to configure IAM to include the scopes linked to an access token in the access token has been introduced using the "scopes" claim as standardised by the OAuth token exchange spec [R9]. Then, on external SAML and OpenID Connect authentications, administrators can define validation rules based on the presence of a given claim (e.g., an entitlement) or attribute in order to allow/limit access to the IAM-managed organisation.

3.5.1.2 Multiple token profiles

The same IAM instance can expose authentication and authorization information according to different token profiles. The token profile used is a set of rules defined at the client application level. A profile may also influence how scopes are used to request inclusion of information in tokens.

3.5.1.3 Login hint on authorisation requests

The support for login hint on authorisation requests is a feature that allows a relying party to specify a preference on which external SAML IdP should be used for authentication.

3.5.1.4 Improved security of actuator endpoints.

The actuator endpoints provide sensible information then, in order to avoid that all IAM administrators have full access, dedicated credentials have been introduced to limit the access to service deployers/developers only. This user (or set of users), specified in the configuration, can be completely orthogonal to IAM users/administrators.

3.5.1.5 Improved management of Acceptable Usage Policies

A migration from the AUP text stored in the database to a URL pointing to a document has been introduced. Such approach would give deployers more flexibility on the AUP document format and presentation.

3.5.1.6 Improved account lifecycle management

The support for account end of life management has been introduced. When accounts have expired, they are suspended or removed (potentially after a grace period) depending on the configuration.

3.5.1.7 Support for the AARC-G002

One of the enhancements was the introduction of support for the AARC-G002 [R10] profile for group membership information.

3.5.1.8 Custom local SAML metadata configuration

This enhancement of a custom SAML metadata configuration means the ability to provide an external metadata file.

3.5.1.9 Disable local authentication

The possibility to disable local authentication (via username/password credentials) and only rely on brokered authentication has been added.

3.5.1.10 Static local resources

One of the enhancements was the support for serving static local resources. This can be useful to support locally hosted logo images, SAML metadata documents, privacy policy documents, etc.

3.5.1.11 Registration through an external IdP

IAM can be configured to require authentication through an external identity provider at registration time.

3.5.1.12 Disable user profile editing

The possibility to disable user profile information editing has been introduced since in some scenarios (e.g., for LHC VOs, where information is populated from an external database) users should not be allowed to edit their personal information.

3.5.1.13 Organisation logo customisation

Deployers can customise the organisation logo size presented in login and other pages.

3.5.2 Future plans

For the future IAM release some dashboard and API fix/improvements already planned, for example:

- IAM should allow customisation of the position of login page UI elements.

- Allow users to change registration name, email, etc when registering via SAML.

- The registration approval page should show more information about user authentication.

- The token management API should not expose token values to privileged users.

- Harmonization activities:

  - Alignment of user attributes following the REFEDS-R&S attribute bundle [R11].

  - Alignment of resource capabilities information according to AARC-G027[R12].

  - Alignment of affiliation information according to AARC-G025[R13].

The main development task is the transition to Keycloak as the main IAM authentication engine. Keycloa is a flexible and popular open-source solution by Redhat for centralized authentication and authorization. Integrating Keycloak in IAM will provide enhanced capabilities and improved sustainability

3.6 Perun

A detailed description of the Perun service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

3.6.1 Maintenance activities

Perun has been regularly updated and new versions were deployed in a regular manner.

Configuration of individual service components was reviewed and simplified, which is a first step to move towards infrastructure as a code.

3.6.2 Summary of service enhancements

The integration of Perun with eduTEAMS and Check-in services was improved during the reporting period. Different VO registrations flows were tested with Check-in service. Prototype of a new API interface for IdP/SP Proxy integration was deployed and it is being tested within eduTEAMS.

Perun is also supporting resource capabilities expression [R12] in entitlement attributes. VO managers can manage access to individual resources in the VO management. This configuration can be consequently passed to IdP/SP proxy which will release it in the entitlement attribute.

The provision engine was optimised with both performance and memory consumption being improved. The next step will be to update individual connectors for services to benefit from the engine upgrade.

A new user interface is in pre-production state and it is being tested with selected representatives of users.

3.6.3 Future plans

We plan to continue to collaborate with user communities to gather their requirements and improve the service based on that. Moreover, Perun will be improving implementation of standards and recommendations to strengthen compatibility with other AAI components and tools.

3.7 WaTTS

A detailed description of the WaTTS service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

3.7.1 Maintenance activities

The WaTTS servers have been kept up-to-date by regular updates and required reboots.

The WaTTS server software received updates with regard to security fixes in the TLS support:

- Old ciphers that are now believed to be weak were removed.
- Support for TLS 1.0 was removed.

We followed the assessment of [R14] to maintain the "A-rating" of the watts servers, which is checked on a regular basis.

The high-availability features were implemented in software. However, it was not possible for KIT to install the existing hardware in the secured environment that fulfils the Credential Store requirements. The reason for this is that the KIT department tasked with providing the hosting environment delivered one year and five months behind schedule. According to new schedule the high-availability installation of the WaTTS service will be provided by the end of May.

3.7.2 Summary of service enhancements

For supporting the WORSICA VO, the VOMS plugin was extended. The WaTTS VOMS plugin allows the request of a VOMS certificate from a VOMS server, right after the actual End Entity Certificate was requested. The extension of the VOMS plugin adds an authorisation and a provisioning step before requesting the VOMS certificate. For this, we check the upstream authorisation (i.e., the

OIDC Attributes from the Community AAI) to determine if the user is considered to be a member of a VO. If authorised the plugin ensures that the user's membership is also reflected in the connected VOMS server. After that VOMS-Proxies issued by that VOMS server for the given user will certify membership in the VO. This should enable VOs to only manage their memberships in their Community AAI. A VOMS-admin server is not required any longer.

3.7.3   Future plans

We will continue to support interested communities with their needs for credential translation and continue to keep the service available.

We still plan on taking advantage of the fault tolerant setup once the hardware is installed in the secure hosting environment.

3.8 MasterPortal

The MasterPortal consists of several components: a webserver in front of TomCat, a database, and a backend myproxy-server for caching long-lived proxy certificates. In addition, it runs an ssh-keys server for retrieving short lived proxies. There are four different TomCat servlets: 1) a client to the RCauth Online CA, responsible for retrieving end-entity certificates and storing long-lived derived proxy certificates in the backend myproxy-server, 2) a server to MasterPortal clients (VO portals or Science Gateways) responsible for retrieving and returning short-lived proxy certificates from the backend myproxy-server, 3) a (demo) VO portal for users to obtain proxy certificates, 4) an ssh-key upload portal. A more detailed description of the MasterPortal service is given in D5.1 [R6].

3.8.1   Maintenance activities

One bug has been fixed in the MasterPortal code. The `/getproxy` endpoint would return a server error in case it could not retrieve a new end entity certificate (EEC) from the Delegation Server (DS). However, that also happened when the latter's access token was no longer valid (expired or already used for retrieving an EEC). It now returns instead a 403 with invalid request error and clarifying description which can be used for matching.

3.8.2   Summary of service enhancements

The /getproxy endpoint of the MasterPortal has been enhanced in several ways. First of all, error handling is improved and in case of error it now returns a JSON with an error and error_description to its client.
Furthermore this endpoint is extended to support getting the myproxy timeleft information (for the long-lived proxy stored in the backend myproxy-server) by specifying a new GET or POST parameter "info". Such an info request returns a JSON containing the username, timeleft, tolerance, max_proxy_lifetime and default_proxy_lifetime of the cached credential:

- ○ timeleft - remaining time in seconds for the long-lived proxy.

- ○ tolerance - also in seconds, used (in combination with max_proxy_lifetime) to determine the longest proxy lifetime that can be requested in a /getproxy call: max_proxy_lifetime - tolerance.

- ○ max_proxy_lifetime - typically the lifetime in seconds of the long-lived proxy, i.e., 950400 (which is 11 days). Used in combination with tolerance to determine the longest proxy lifetime that can be requested in a /getproxy call.
- ○ default_proxy_lifetime - default lifetime in seconds of returned short-lived proxy, typically 43200 (i.e., 12 hours).

NOTE: if a /getproxy request - including the new info request - has a proxy lifetime (or its default default_proxy_lifetime) which is longer than timeleft, a new long-lived proxy needs to be stored in the MasterPortal, which means that a flow for obtaining a new EEC from the Delegation Server will be initiated in such a case.

3.8.3 Future plans

The MasterPortal - like the Delegation Server of the RCauth Online CA (see next section) - will need to run in a high-availability mode. Since the technological implementation of this setup will be almost identical to that of the Delegation Server, we await the results of that task first.

3.9 RCauth - Online CA

The RCauth online Certification Authority (CA) service [https://rcauth.eu/] consists of three main service components: a database, a signing server, and a delegation server through which the clients interact with the CA. Optionally, the delegation server can be supplemented with a WAYF (Where Are You From) front end. A more detailed description of the RCauth - Online CA service is given in D5.1 [R6]

The major contribution of EOSC-hub is to replicate RCauth instances, so the service is run across three sites rather than one, by replicating the original CA at Nikhef to also run at STFC and GRNET. However, due to the high complexity of the task and the COVID lockdowns (and loss of staff), there have been delays in finalising the work.

3.9.1 Maintenance activities

Many parts of the RCauth tasks have been highly technical - cross-site synchronised databases, secure data replication, secure key exchanges, secure communications. As many parts of RCauth are quite complex, maintenance of these services can also involve some complexity. The regular maintenance activities include:

- ● Operations - we have weekly operations calls, to track activities that require coordination between the sites. Experience has shown that the regular Task 5.1 calls alone were not sufficient; we need a weekly one-hour call dedicated to RCauth (once the service is fully in production, fortnightly calls should be sufficient).
- ● Routine maintenance of systems - databases, VPN (virtual private network), etc.
- ● Ensuring the EUGridPMA is kept informed of the progress.

The need to keep EUGridPMA informed arises because the IGTF accreditation of RCauth is established through the EUGridPMA. Cloning a CA between different organisations has never been

done in IGTF, so it is important that the PMA has approved any RCauth plans prior to their implementation and is kept informed of the experiences and lessons learned from the process.

In addition, the EUGridPMA meetings have recently incorporated activities like GÉANT EnCo, the WISE community, and AARC communities, which has broadened the audience with even more security, technical, policy and operations people. This is useful not just to get wider feedback on the technical or policy issues, but also helps us communicate our innovations and experiences to a wider community.

3.9.2 Summary of service enhancements

In the final reporting period, the following improvements have been made:

- The database is now fully replicated across all three sites. Replication is done over a VPN secured with X.509 certificates.
    - Our initial plan was to use primary/replica replication, with each site acting as primary (formerly called 'master'). However, this turned out to work only in a two-way replication, so we had to migrate to a Galera cluster setup.
    - This setup has been tested carefully, even with an unrealistic client usage pattern where parts of the interaction goes to one site and parts to another - the databases synchronise quickly enough to make the delegation server(s) interactions fully transparent.
- The VPN network which is necessary to synchronise the databases with each other has been reconfigured to a High Availability (HA) setup where each site hosts a server and is a client for another site. Thus, each site is connected to the other two, and if a link goes down between two sites, the data will route through the third as a backup link. This, too, has gone through extensive testing.
- The key exchange process, critical to the secure distribution of the actual RCauth private key, has been demonstrated with a less critical key. The distribution of the production key was delayed by COVID lockdown of the data centres.
- One site is currently running a HA Proxy fronting the delegation servers.

3.9.3 Future plans

A few key tasks need to be finished to have achieved the goal of having a full HA RCauth deployment across the three different sites.

1. The production key needs to be replicated from Nikhef to GRNET and STFC, which in turn depends on all three sites having access to their respective machine rooms hosting the servers and the specialised hardware.
2. All sites need to run HA Proxy servers for the delegation servers - at present we have one; we need three, which need a round-robin DNS or similar.
3. CRL issuance and synchronisation. A plan exists for this, but it needs implementing.

These tasks are all fairly complex, and most are in progress, but they need to be fully completed before we have achieved the HA RCauth setup. These are all expected to be completed by the end of March 2021.

Other important future tasks include:

- Present work to the EUGridPMA in order to retain the IGTF accreditation. This was intended for the February 2021 PMA but was postponed to the Spring PMA (date tbd). The intention is to present a self-audit (as required by PMA policy) of the full production HA RCauth CA.

- Additional mostly site-specific production setups - backups, monitoring, documentation, etc.

- Setting up WAYFs - although it is a less complicated task, it has been agreed that WAYFs are part of the RCauth service and should also be HA. It is probably not necessary to have (redundant) HA Proxies; a DNS-based solution would probably be sufficient.

In general, it is clear that replicating RCauth was more complex than anyone had expected (and of course, COVID caused additional delays). As this is the first time a CA has been replicated in IGTF, quite a few innovations have been brought together, improving on the state of the art of operating an online CA. As a part of EOSC's commitment to open science, it is important we share what we have learnt.

3.10 Integration activities

3.10.1 Summary of integration activities

3.10.1.1 Integration of EOSC-hub AAI services

This section provides information about the integration between the EOSC-hub AAI services. Following the AARC Blueprint architecture [R16], each EOSC-hub AAI service can act as a Community AAI enabling the use and management of community identities for accessing resources, while, at the same, it can act as an Infrastructure Proxy enabling access to resources through other Community AAIs. To this end, each row in Table 3-4 shows the integration of a particular EOSC-hub AAI service in its capacity as a Community AAI with the rest of the EOSC-hub services as Infrastructure Proxies. For instance, the first row provides information about the integration of B2ACCESS as a Community AAI with Check-in, eduTEAMS and INDIGO-IAM as Infrastructure Proxies. The "✕" symbol is used to denote integrations enabled in production, while "╱" denotes integrations that have been tested in development/staging environments. The latter will be moved to production once there is at least one-use case for a community to access resources involving the specific Community AAI and Infrastructure Proxy(ies). This is expected within the EOSC-Future project following the proposed EOSC AAI Federation model [R17] which will provide for a more scalable approach for connecting AAI services (see also Section 3.11).

Table 3-4 Integration of EOSC-hub AAI services

	B2ACCESS	Check-in	eduTEAMS	INDIGO-IAM
B2ACCESS		✕	╱	✕
Check-in	✕		╱	✕
eduTEAMS	✕	✕		╱
INDIGO-IAM	✕	✕	╱	

3.10.1.2   Improved integration of EOSC Portal AAI with ORCID

The ORCID login connector used for the EOSC Portal AAI was reconfigured to use ORCID member API credentials instead of the ORCID public API. Using the ORCID member API allows reading limited-access information on an ORCID record in addition to any information made publicly available by that record's holder. More specifically, member API credentials enable requesting permission from users to read limited-access data on their ORCID records through 3-legged OAuth authorization [R18]. It should be noted that this limited-access permission is valid only for that individual's ORCID record.

One of the benefits of using the member API credentials is that the email address of the authenticating user can be made available (after the user's explicit authorisation) to the services connected to the EOSC Portal AAI, even if the email address visibility is restricted to trusted third parties only. Previously, ORCID login through the EOSC Portal AAI required users to set their email information as public.

3.10.1.3 Keycloak extensions

Keycloak [R19] is a flexible and popular open-source identity and access management solution. During the reporting period, the task developed extensions and customisations tailored to scientific collaboration use cases. Specifically, the goal was to facilitate the integration of Keycloak with SAML identity providers and OpenID Connect relying parties in an identity federation context. This was achieved by implementing the following features:

- **SAML federation support:** Keycloak already supports SAML authentication, primarily in the context of single SAML IdP enterprise use cases. This was extended to support identity federations. Specifically, an operator can configure Keycloak to register IdPs through a SAML metadata aggregate instead of many individual IdP metadata sources. Keycloak's login page was also extended with search capabilities to properly support the discovery of a large number of authentication providers that typically become available through a federation.

- **OIDC federation support:** The objective of this task was to extend Keycloak to support the OpenID connect federation standard and demonstrate interoperability with other

components/libraries currently in development in the context of the OpenID foundation. The implementation focused on enabling Keycloak as an OpenID Provider that can interact with federated relying parties through explicit client registration. The implementation of the automatic client registration is still work in progress since it requires feedback from the Keycloak development team.

The Keycloak extensions activity followed an open development process. All the code repositories are hosted on GitHub [R20]. It should be noted that parts of these extensions have been submitted as pull requests (PRs) against the upstream Keycloak source code repository [R21] (two PRs have been approved, while the rest are under review by the Keycloak development team). Some contributions also required submitting PRs in the Keycloak Documentation repository [R22].

3.10.2 Identified integration gaps

The EOSC-hub AAI services have been integrated following the AARC Blueprint Architecture [R16] in order to streamline researchers' access to services and resources across infrastructures. Specifically, each EOSC-hub AAI service can act as a Community AAI enabling the use and management of community identities for accessing resources, while, at the same, it can act as an Infrastructure Proxy enabling access to resources through other Community AAIs. This enables researchers to use their community identity for accessing resources without the need for multiple registration processes, which was one of the key usability issues identified in D5.3 [R8]. However, the following issues remain open:

- **Multiple IdP discovery steps:** Users typically need to go through multiple IdP discovery steps; for example, they may need to first select their Community AAI and then select the IdP of their Home Organisation. During this process, users do not need to re-enter their login credentials, however the IdP selection can be frustrating in some cases. The adoption of the upcoming IdP "hinting" specifications by the AARC community will significantly simplify the discovery process by either narrowing down the number of possible IdPs to choose from or by allowing to bypass the actual selection process.

- **Token-based multi-infrastructure workflows:** The current EOSC-hub AAI architecture works very well when the user is consuming services directly. However, there are use cases requiring a service agent to be able to act autonomously, on behalf of the user, consuming services, and resources. If the services consumed by the agent are behind the same proxy, the current architecture works. For those cases, though, where an agent running on Service A needs to access resources on Service B, which might be connected by a different proxy, then there is no straight-forward solution at the moment. So, currently, services need to trust the same proxy to support those use cases. A solution for dynamically establishing trust in a distributed environment will be provided by the OpenID Connect Federation specification v1.0 (draft) [R23]. The AARC community is investigating an extension of the OAuth2 Token Introspection specification as an interim solution until the OIDC Federation Specification is widely available.

3.11 Summary and outlook

The activities carried out within this task focused on the delivery of an integrated EOSC-hub AAI for enabling Service Providers to offer their resources to research communities and individual researchers, allowing users to use their institutional and community-based digital identities.

The main achievements of this task are:

- Harmonisation of user and community information expression, including alignment of user attributes and adoption of guidelines for expressing authorisation attributes such as groups and roles

- Adoption of recommendations and best practices for implementing a scalable policy and operational framework for the integrated EOCS-hub AAI, including the alignment of operational security and incident response policies meeting the Sirtfi [R24] requirements, as well as the alignment of Acceptable Use Policies (AUPs) following the WISE Baseline AUP model [R25]

- Demonstrated technical ability for research communities using different Community AAIs (B2ACCESS, Check-in, eduTEAMS, INDIGO IAM) to access resources

- Adoption of standards and open technologies, such as SAML 2.0, OpenID Connect, OAuth 2.0 and X.509v3 that enables sustainable interoperability with the AAIs components of other research- and e-Infrastructures

- Improvement of the white labelled online certification authority (RCauth) and token translation service

- Delivery of a white labelled AAI service (EOSC Portal AAI) for enabling individual researchers and research communities to access the EOSC Portal

- Provided coordinated consultancy, support, and guidelines on how to connect Service Providers and Identity Providers to the EOSC-hub AAI

It should be noted that the integration between the EOSC-hub AAI services required the establishment of trust between these services in a peer-to-peer manner (e.g., through the exchange of SAML metadata or static OIDC client registration mechanisms). Despite the adoption of common technical and policy guidelines, the establishment of M:N relationships can pose scalability issues, particularly with the growing number of Community AAI and Infrastructure Proxy services that need to be interconnected for enabling access to resources across infrastructures within the wider EOSC environment. To address scalability concerns, the AAI task force of the EOSC architecture working group has proposed the establishment of an EOSC AAI Federation [R17], thus alleviating the need for bilateral relationships. Further improvements have already been planned in the context of EOSC-Future and other EOSC-related projects aiming to fill the identified gaps described in Section 3.10.2. The evolution of the architecture, as well as the alignment activities on the technical and policy level will be continued in collaboration with the AARC Engagement Group for Infrastructures (AEGIS) [R26] and the pertinent EOSC Task Forces addressing its authentication and authorization infrastructure.

4 Marketplace and Order Management tools

4.1 Overview

The EOSC-hub Order Management System (OMS) comprises several central components including the EOSC Portal Catalogue and Marketplace (MP), Service Portfolio Management Tool (SPMT), Service Order Management Back Office (SOMBO) and integrates many other systems to facilitate promotion, discovery, access and ordering of the productional EOSC-hub resources in the distributed EOSC environment.

The EOSC-hub OMS allows customers to access the central EOSC Portal Catalogue and Marketplace, discover resources using the intelligent search and filtering mechanism, place an order and track it until its fulfilment and the resource delivery. The EOSC-hub OMS can integrate Order Management Systems, which are operated by other e-infrastructures.

4.2 Marketplace

A detailed description of the initial version of EOSC Portal Catalogue and Marketplace resources is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

4.2.1 Maintenance activities

In order to ensure high availability and the platform's smooth operation, regular software updates including gems' updates are being performed.

4.2.2 Summary of service enhancements

During the last months, the Marketplace has been evolving in various dimensions, but the most significant enhancement was the **Whitelabel** solution [R27]. This solution provides a possibility for communities and other interested parties to create their own branded Marketplace based on the EOSC-hub Marketplace solution. This solution has many advantages both from provider and user perspectives. It allows providers to manage their services and service offers in their own dedicated Marketplace and on the other hand it allows users to enjoy the benefits of using the main features of Marketplace such as: service search and service browsing, service order, contact with service support. There were many technical tasks conducted to achieve this solution: code refactoring, webpack reorganisation, reorganisation of development process, defining the process to build and deploy, establishment of a common framework to create branded Marketplace instances. Each Marketplace has its own customizable branding elements (logotypes, footers, colors) and it allows the owners to customize the main view of their Marketplace according to their preferences. Moreover, all instances share the same application code, the source code is built by an automatic builder for a specific branding purpose. From the owner perspective each Marketplace can have a separate codebase - its own home page, links to contact information, "about us" sections etc. Other changes of the GUI can be applicable in the source code. This solution has great potential and could be developed in many directions in the future.

Another big achievement was done with respect to service **offer attributes**. Service offers, with a possibility to adequately configure their attributes while defining a given service offer by the

provider, present different versions (in terms of business or technical parameters) of the service offering in the user-facing service presentation page as shown in Figure 4-1. Service offers were added to the autocomplete and search tool, to make them easy to find. Moreover, the user interface for offers management was added to the Marketplace Backoffice to make offers easy to define in the Marketplace.
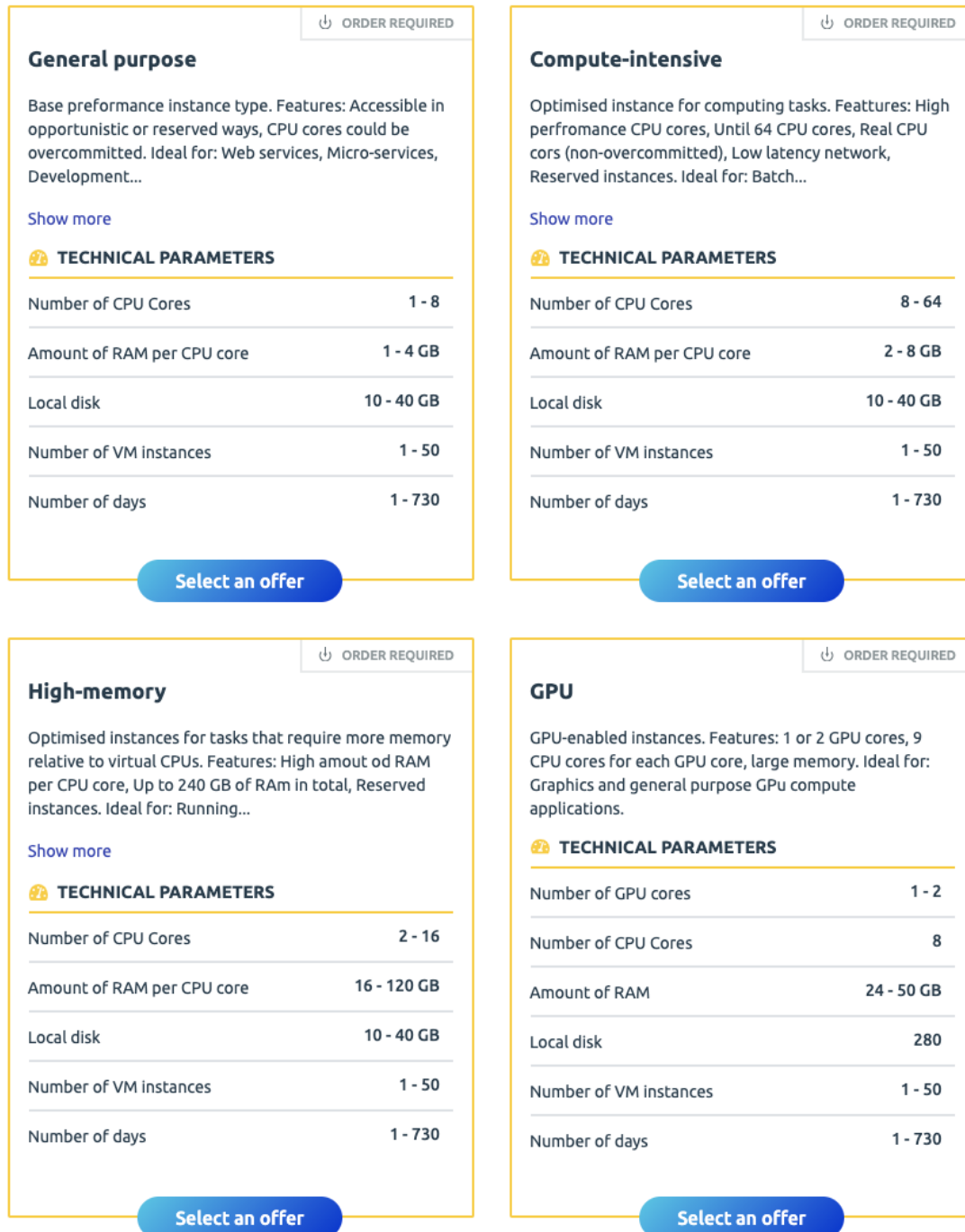

General purpose

⏻ ORDER REQUIRED

Base preformance instance type. Features: Accessible in opportunistic or reserved ways, CPU cores could be overcommitted. Ideal for: Web services, Micro-services, Development...

Show more

TECHNICAL PARAMETERS

Number of CPU Cores	1 - 8
Amount of RAM per CPU core	1 - 4 GB
Local disk	10 - 40 GB
Number of VM instances	1 - 50
Number of days	1 - 730

Select an offer

Compute-intensive

⏻ ORDER REQUIRED

Optimised instance for computing tasks. Feattures: High perfromance CPU cores, Until 64 CPU cores, Real CPU cors (non-overcommitted), Low latency network, Reserved instances. Ideal for: Batch...

Show more

TECHNICAL PARAMETERS

Number of CPU Cores	8 - 64
Amount of RAM per CPU core	2 - 8 GB
Local disk	10 - 40 GB
Number of VM instances	1 - 50
Number of days	1 - 730

Select an offer

High-memory

⏻ ORDER REQUIRED

Optimised instances for tasks that require more memory relative to virtual CPUs. Features: High amout od RAM per CPU core, Up to 240 GB of RAm in total, Reserved instances. Ideal for: Running...

Show more

TECHNICAL PARAMETERS

Number of CPU Cores	2 - 16
Amount of RAM per CPU core	16 - 120 GB
Local disk	10 - 40 GB
Number of VM instances	1 - 50
Number of days	1 - 730

Select an offer

GPU

⏻ ORDER REQUIRED

GPU-enabled instances. Features: 1 or 2 GPU cores, 9 CPU cores for each GPU core, large memory. Ideal for: Graphics and general purpose GPu compute applications.

TECHNICAL PARAMETERS

Number of GPU cores	1 - 2
Number of CPU Cores	8
Amount of RAM	24 - 50 GB
Local disk	280
Number of VM instances	1 - 50
Number of days	1 - 730

Select an offer

Fig.4-1 Marketplace Service Offers for EGI Cloud Container Compute Service.

The Marketplace API improvements were developed in the EOSC-hub project extension period. During this time, the following new MP API capabilities were achieved:

- Implementation of an API for provider offering/offering parameters which gives providers an additional way to manage their service offers (in addition to the service offerings dedicated UI). API documentation is provided [R28].

- Implementation of the order handling integration API, with a reference integration with EOSC-hub JIRA system which is a part of EOSC-hub Order Management System

Other enhancements made in this reporting period (including the project's extension) according to EOSC-hub project:

- Refactoring for each type of the service order type so the right information is displayed in the service Information page

- Sharing the relevant information about published services (service owners, contacts etc.) with the ARGO monitoring tool and SOMBO using a dedicated MP interface

- Configuring Google Analytics to share statistics with SOMBO

- Extending the scope of the statistics available in the MP Executive Panel.

In the meantime, a lot of work was done in the scope of the EOSC Enhance project. Delivered enhancements are described in the project's internal documentation and will become publicly available in D4.2 deliverable: UX model and verification (update).

4.2.3 Future plans

The EOSC-hub Marketplace is a powerful tool for users that allows easy access to a large number of resources for various research domains along with integrated data analytics tools. The development planned in connection with the EOSC-hub project has been completed, but the Marketplace is still a tool that can be developed in many directions and on many levels.

In addition, the development will be continued in the EOSC Enhance project which plans the following developments in the near future:

- Recommendation engine for the users which suggests possible services of interest based on: searches of 'similar' users (similarity considers users' profiles), EOSC services that the user has already searched/ ordered and other relevant users' actions taken in the Portal

- Improvements in the EOSC MP Projects which consider input gathered during user workshops and data-based analysis

4.3 Service Portfolio Management Tool (AGORA)

The Service Portfolio Management Tool - AGORA provides a full list of Providers and allows the user to manage service descriptions according to the service management guidelines of FitSM. Agora also allows exporting service descriptions to other tools and service catalogues, such as https://www.eosc-hub.eu/catalogue. A view of the Web interface of Agora that depicts its evolution to adopt profiled 3.0 is seen in Figure 4-2.

Fig.4-2 EOSC-hub Agora Web UI.

4.3.1 Maintenance activities

There is a standardized maintenance window every first Wednesday (Devel Instances) and Thursday (Production Instances) of each month. These maintenance windows are used for applying regular OS upgrades and stable releases. All necessary precautions (backing the data) are taken care of beforehand by the monitoring team.

One major part of maintenance activities are the updates / upgrades of the software / library dependencies that AGORA has. This follows a specific process where performance, features, and service stability are taken into consideration. When a reliable version of a software dependency is available, the development team deploys a new stand-alone instance to test the validity of all main features and decide on a list of changes required. When a stable version is implemented, it is deployed on the development instance for at least one month until it is deployed in the production service. AGORA follows an agile development process that includes mandatory tests for checking the functionality and the quality, correctness of the software. This process consists of automated unit tests and code quality checks, running via a CI tool (GitLab). Unit tests that test CRUD and domain logic functionality on all resource objects supported by the API.

4.3.2 Summary of service enhancements

During this period AGORA was further developed and evolved to become a Resource Portfolio Management Tool with Fully support for EOSC Profiles 3.0 [R29]. In order to add support for EOSC Profiles 3.0 the model from AGORA was developed from scratch as these profiles are not backward compatible with the Service Description Templates that were used by the previous versions. In more detail during this period, we provided a number of enhancements to respond to the requirements of EOSC-hub such as:

- Support for Resource Contact Information
- Support for Resource Geographical and Language Availability Information
- Support for Resource Classification Information
- Support for Resource Marketing Information
- Support for Resource Basic Information
- Support for Resource Classification Information
- Support for Resource Location Information
- Support for Resource Management Information
- Support for Resource Attribution Information
- Support for Resource Maturity Information
- Support for Provider Networks
- Support for Provider Maturity Information
- Support for Provider Contact Information
- Support for Provider Other Information
- Support for Provider Marketing Information
- Support for Provider Basic Information
- Support for Provider Classification
- Support for Provider Location
- Deployed new instances
- Created new public react based view that lists only published Providers
- Created a workflow to publish Resources and Providers to a Public View and/or to the EOSC Portal.

4.3.3 Future plans

Continue to evolve AGORA to follow the developments and the next releases of the EOSC Profiles and improve its usability and functionality.

4.4 Integration activities

4.4.1 Integration of Marketplace with EOSC Portal

4.4.1.1 Summary of integration activities

Main activities in the scope of integration with the EOSC Portal were conducted according to the EOSC Enhance project. A short summary of work that was undertaken follows:

There were 2 catalogue instances running in the EOSC Portal; eInfraCentral (EIC) Catalogue (based on EIC Service Description Template (SDT) v1.13) and EOSC-hub Marketplace (based on MP SDT

v1.3.0). Until November 2019, onboarding was enabled in both catalogues, thus leading to inconsistent resource information hosted in the two catalogues. Because of that, it was decided to aim EOSC Portal development at maintaining a single catalogue of resources, in which the User facing component integrates with EOSC Portal Registry API and they interact to add\remove\update resource information. So, now in the EOSC Portal we can distinguish 3 components:

- Content Component (EOSC Portal Website based on Drupal CMS)

- User Component (Marketplace)

- Provider Component (including EOSC Portal Resource Registry)

Communication between these two components is led through Java Message Service (JMS). Figure 4-3 shows the high-level architecture of EOSC Portal including its components and interactions.
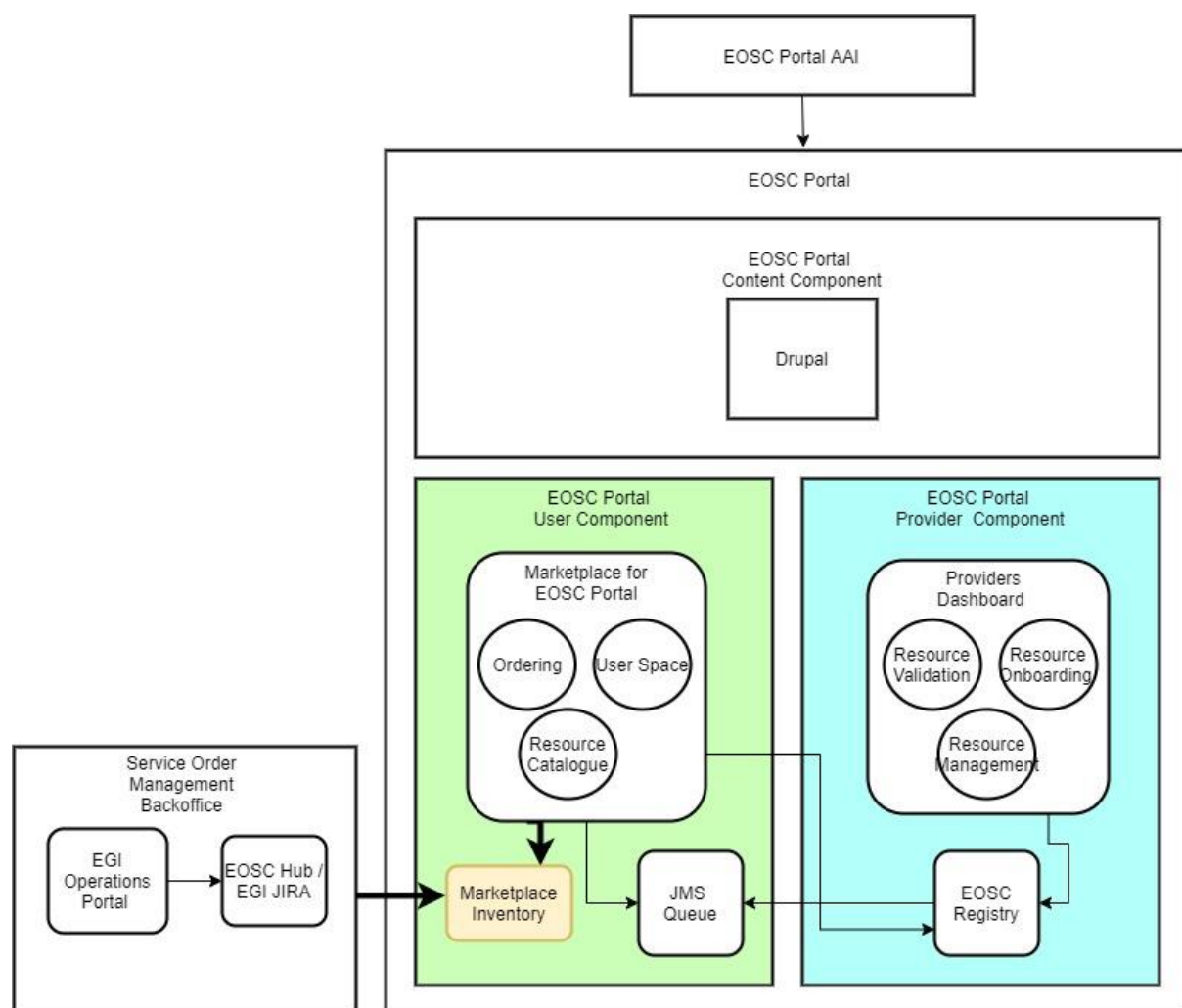

Fig.4-3 Current EOSC Portal high-level Architecture.

In the reporting period the main activities conducted lead to synchronizing the service catalogue and the fronted based on the new Resource Description Template (RDT) version [R29].

4.4.1.2   Identified integration gaps

The synchronization between catalogues is still fresh and needs to be monitored and in case of any inconsistency, and it is being constantly improved.

4.4.1.3   Future plans

EOSC Portal will be developed with an emphasis on the user experience element as part of the EOSC Enhance project. An appropriate methodology will be used to analyse the EOSC Portal user experience and then measure progress during the implementation.

The project will also develop profiles for different EOSC Portal objects. An upgrade of the service resource type to version 3.0 has been included in the Portal at the end of the EOSC-hub project (as a part of EOSC Enhance) and, during the EOSC-hub extension, the representation of EOSC providers has been introduced using the Provider Profile. The next significant step is the delivery and inclusion of the EOSC data source profile. It is a highly anticipated step in the whole EOSC community, and a great deal of consideration is devoted to both: developing the profile itself, implementing the profile and making the EOSC data resource available in the EOSC Portal. The EOSC Enhance project's current findings regarding integrated data discovery which are based on the conducted research and analysis can be found in the deliverable D4.3: Analysis of existing research data cataloguing efforts towards integrated discovery [R43].

EOSC Enhance project will also investigate and work on the profiles for EOSC research products to deliver the first full version of the profiles for the EOSC Portal resources.

4.5   Summary and outlook

At the very beginning of the project, the main emphasis according to the evolution of the Marketplace was put on the EOSC user-facing functionalities. The platform was designed to facilitate basic resource discovery and, most importantly, resource orders. The user was able to crawl the database, filter the results to depict the resource of interest and issue an order for it. Each of these features was implemented rather in its simplest form and required deeper business and operational analysis. After several months of the EOSC Marketplace being in production, Marketplace users and resource providers started to provide requirements and insights based on their experience in the Marketplace. This gave an additional input for the analysis required and allowed to start implementing the system's enhancements.

At the beginning of the project, the EOSC landscape also differed. The scope of MP team dev plans was largely connected with integrational activities with other federational tools developed in the EOSC-hub projects. The list consisted of:

- Integration with EOSC-hub AAI

- Integration with EOSC-hub SPMT

- Integration with EOSC-hub CMDB

- Integration with EOSC-hub ARGO monitoring system

- Integration with EOSC-hub Operation Portal

As a result of decisions taken on the EC level, especially a decision to develop the EOSC Portal where EOSC-hub Marketplace will play a user-facing role, the scope of planned activities has significantly changed. Because a large amount of development was needed for the delivery of the Portal, the integrational plans for CMDB and monitoring system were dropped and the integration with the ESOC-hub SPMT (as the tool meant for service management for providers; integrational efforts were started) was replaced with the integration with EOSC Resource Registry (former part of e-InfraCentral Catalogue). On the other hand, the EOSC-hub development of the Distributed Order Management Process, replaced the Marketplace - Operations Portal integration with a new integration depending on a JIRA system, being a middle-man between the first and the latter.

The delivery of the EOSC Portal resulted in the EOSC Enhance project (meant for a dedicated EOSC Portal development. The EOSC Enhance project was preceded by the EOSC-hub cooperation on the project level with OpenAIRE-Advance, EFIS, UoA and JNP. The cooperation was formally supported by the Collaboration Agreement [R30], which goals and dedicated development activities also influenced the current shape of the Marketplace, especially in the area of user experience improvements. The specification of these improvements is formed on the basis of a dedicated UX methodology established within the EOSC Enhance project.

In general, it is clear that this task was more complex than anyone had expected. During the course of the project, there were many expected (impact of the EOSC Enhance project) and unexpected (Covid) situations that had an impact on the final shape of our product. However, in the end, we managed to achieve all the goals intended in the project, and we are fully convinced that we provide a successful product which is an integrated platform that allows easy access to lots of resources for various research domains along with integrated data analytics tools.

Of course, it is clear that the possibilities of Marketplace enhancements are not finished while we can see here so much to achieve with its current potential. However, further work will be carried out in the scope of EOSC Enhance and EOSC Future project.

5 Integrated Business and Operations Support Systems

5.1 Overview

This section describes the maintenance and integration activities for several operational services like Operations Portal, GOCDB, DPMT, DMP and SVMON. Many enhancements have been introduced and described for each service in the corresponding section. In addition, we provide an overview of collaborative work which has been carried out together with WP4 on the establishment of the Configuration Plan for federation services, which should provide guidelines for further improvement of the configuration management system and interoperability of its components. The summary section concludes the work done in these tasks and provides an outlook for the future.

5.2 Operations Portal

A detailed description of the Operations Portal service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

5.2.1 Maintenance activities

The usual maintenance activities have been applied to ensure the performances of the application

- bugs fixes
- regular software updates

5.2.2 Summary of service enhancements

This last year several improvements and new features have been deployed:

1) **API**

    The Operations Portal is an historical tool and with the evolution of features we have proposed at different times different urls for programmatic access have been implemented. But these different urls have been proposed without any harmonization (format of url, access mode, formats), without access control (http access) and without a central place referring to all these urls.

    So, the decision was taken to replace all these urls by an API provided in one place with a token access policy (requires a validation) and detailed documentation.

    Consequently, a new page has been deployed for the management of this API. A token is mandatory to get access to the content of the different methods which you can request on this page (see Figure 5-1), test the different methods (see Figure 5-2), and get access to the documentation (see Figure 5-3).
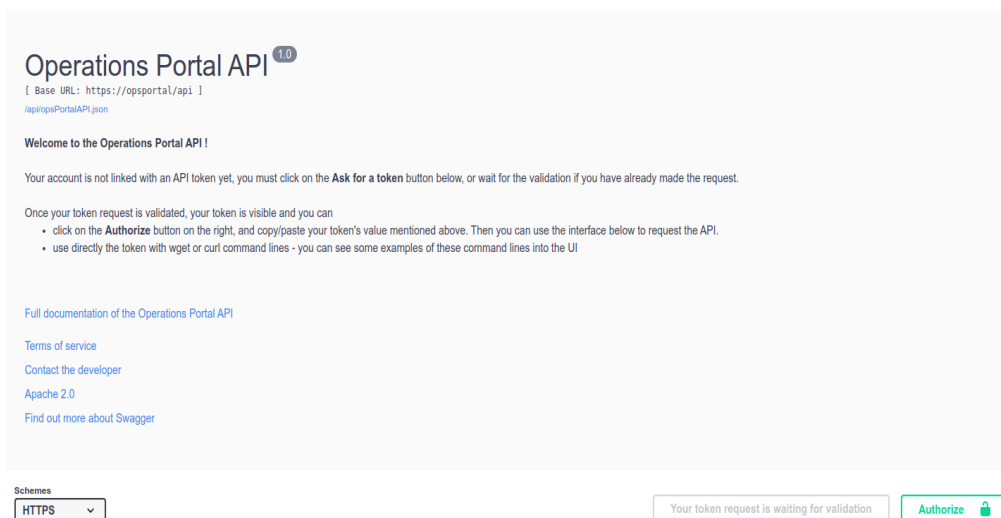
Fig.5-1 Landing page for the API: request a token.
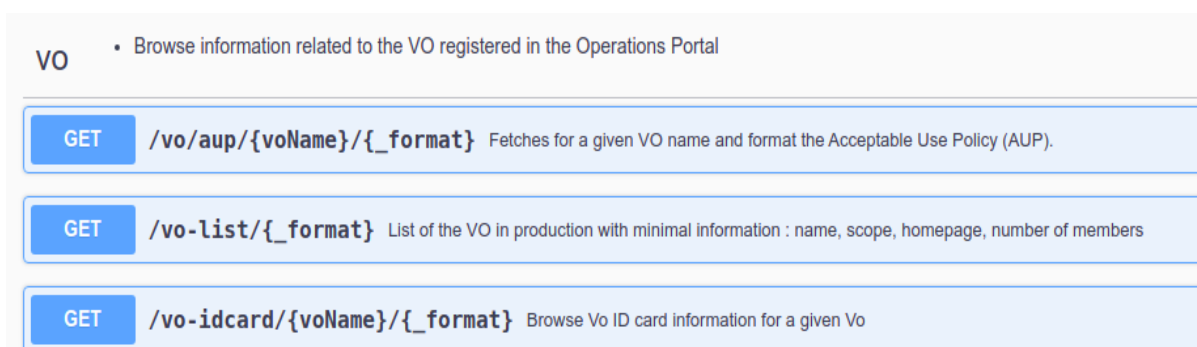
Fig.5-2 - Example of methods available in API.
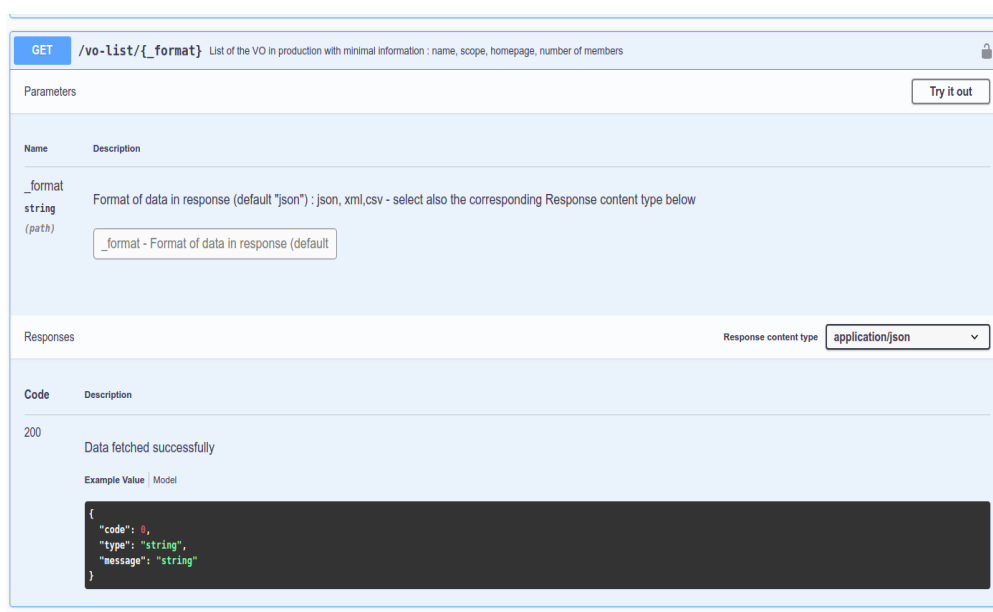
Fig.5-3 - User interface: access documentation and test methods.

2) **EOSC Metrics**

A new module has been put in place collecting metrics from Jira and Google Analytics in order to generate reports for the European Commission. Figure 5-4 shows high-level architecture of the metrics module. On the Figure 5-5 an example page of report for European Commission is shown.
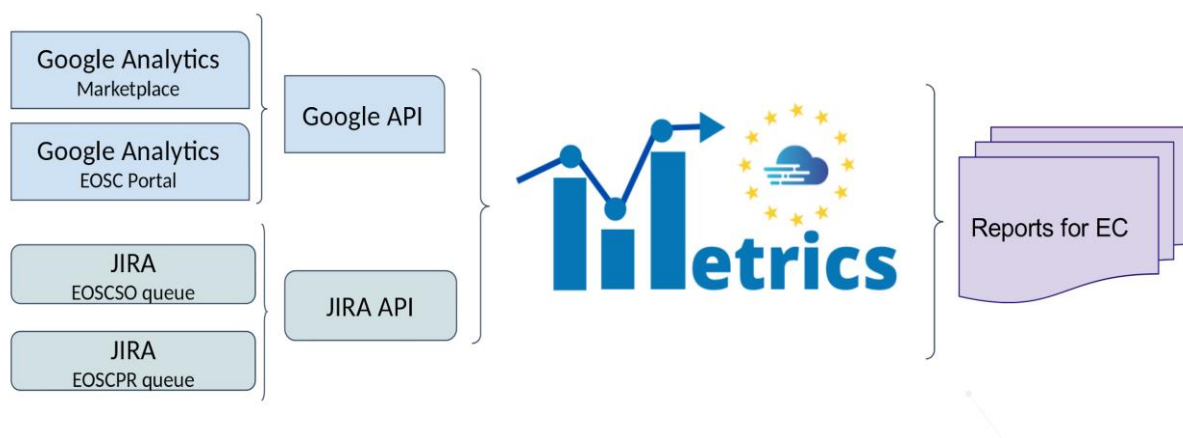
Fig.5-4 - Architecture of EOSC metrics module.

Figure 5-5 - Example of report for EC

3) **VO ID cards**

The VO ID Card is a tool allowing the registration of VO into the Operations Portal and allows to record the life cycle of a given VO, additionally you can give general information about the VO: homepage, description, administrative information: contacts, mailing lists and information about core services of the VO.

This feature is an historical feature of the Operations Portal and it was necessary to refactor this feature. So, this part has been reviewed and different improvements have been done.

Firstly, we have reviewed the ergonomics and the workflow of registration to simplify it for users. These new ergonomics are cleared, and we have compiled all mandatory sections at the beginning of the registration in order to skip eventually all the complex / more elaborated sections.

We have also removed obsolete fields and added new one following the evolution of the procedures and the services used by VO. Finally, we have integrated the possibility to add a new type of registries: CoManage and any external registry (different from historical registries: VOMS or Perun).

4) SOMB0

The Service Order Management Back Office (SOMBO) is the orchestration tool between Marketplace, service providers, service requesters and shifters/operators. The aim is to facilitate the daily operations made by shifters, ease the communication between all parties, facilitate the negotiation between service requesters and service providers, provide facilities to sign SLA/OLA.

The SOMBO application provides a complete dashboard (to shifters to list all service orders per status (on-going, accepted, rejected) and to operate them as shown in Figure 5-6.

The shifters can access the details of the service orders in order to:

● Update some fields and exchange comments with users requesting resources

● manage service orders, assign a service order to different resource providers, and negotiate with these providers (exchange of information, validation or rejection of the resource request as shown in Figure 5-7).

● generate OLAs and SLA documents for EGI services.
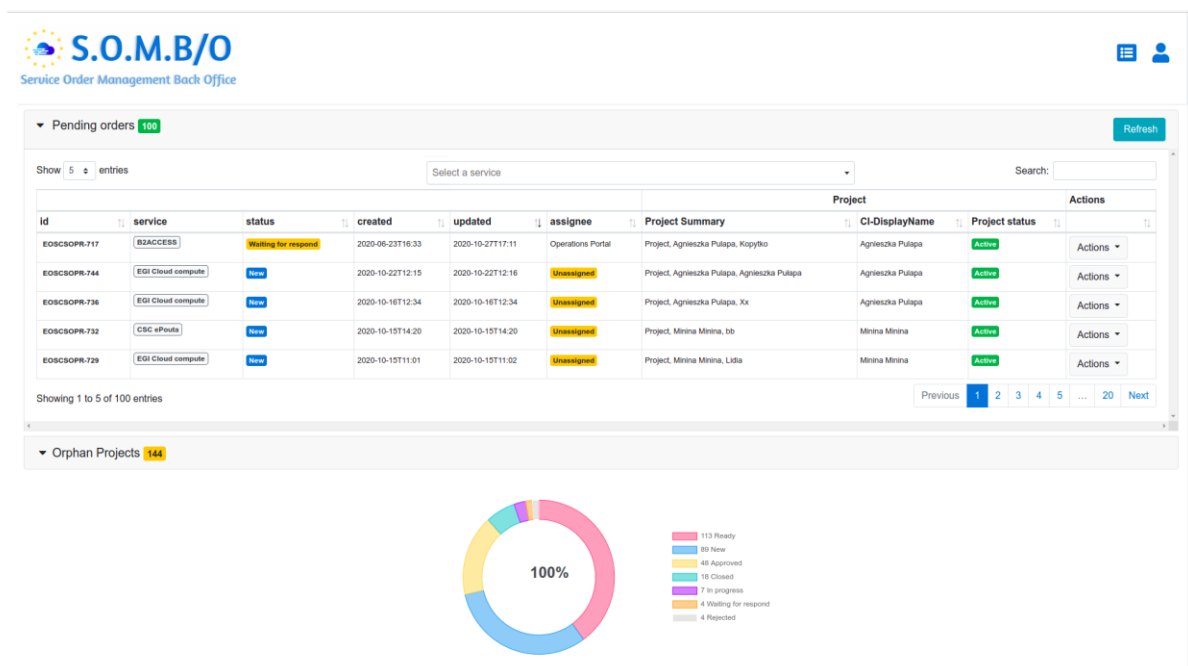
Figure 5-6 - SOMBO Dashboard for shifters.

SOMBO also provides tools to ease the communication and propose to exchange comments, feedback with the service requestor but also propose to add internal comment/note between operators/shifters.
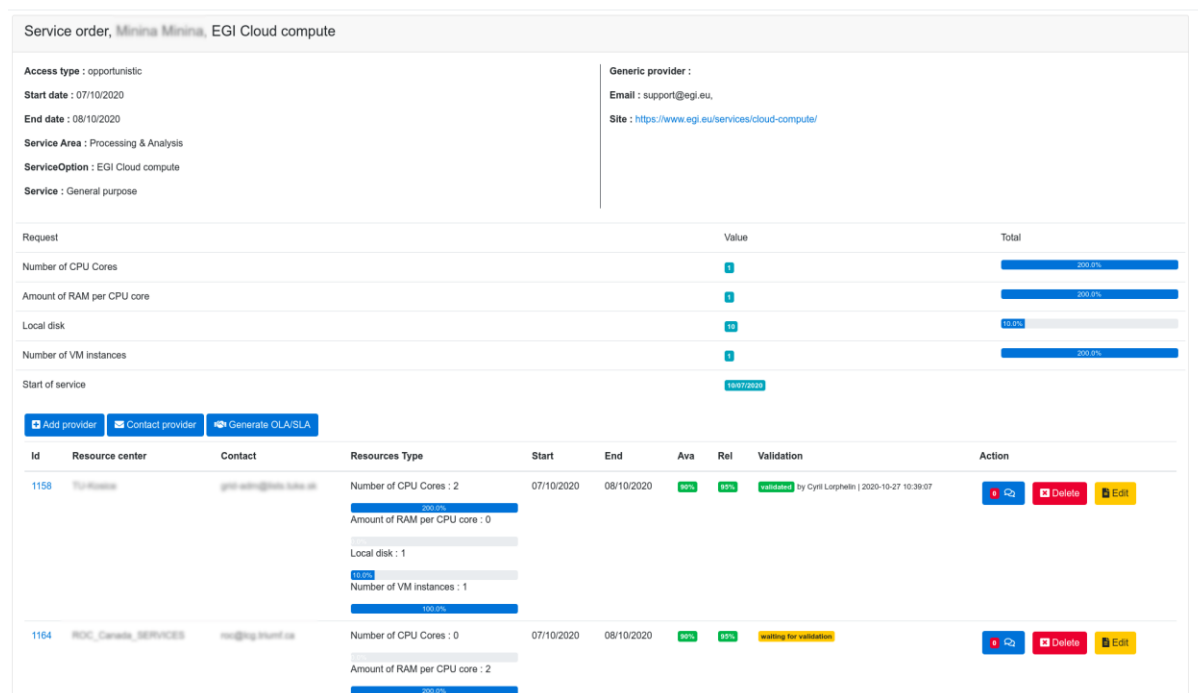
Fig.5-7 - Interface for management of service orders.

5) Dedicated instance for EOSC oriented tools

For a better visibility and clear separation of the different tools we have decided to build a new instance of the Operations Portal into the EOSC scope: https://opsportal.eosc-portal.eu

This instance is also the opportunity to make a technological upgrade with the use of the last version of the Symfony framework and also the last version of the css framework Bootstrap.

The authentication layer is ensured with the use of the EOSC AAI.

And for the content we have migrated the SOMBO Module and the EOSC Metrics described previously. With the upgrade of the different libraries, we have been obliged to make a quick review of the code to ensure a full compatibility.

Last but not least we have decommissioned the historical module in the Operations Portal and added different redirections to the new instance.

5.2.3   Future plans

The effort for the end of the project will be focused on these items:

- API: add new methods adapted to the need of users

- SOMBO: add a resource provider dashboard. This dashboard will provide a summary of the request per provider. This part could also provide different usage reports in a second phase. The remaining issue is the authentication of the provider, there is currently no mechanism to associate an authenticated user in EOSC AAI with a provider.

  We should study an alternative solution to AAI:

- Either gives open access to this dashboard

- Either creates local authorizations in SOMBO with requests of authorization validated by shifters.

5.3 GOCDB

A detailed description of the GOCDB service is given in D5.1 [R3]. The release notes for the reporting period are provided in Appendix A.

5.3.1 Maintenance activities

Various bugs were fixed, including allowing deletion of Site Extension Properties through the Write API as intended. To eliminate the need for portal users to download separate root certificates into their browsers, the certificates used by the service are both part of the IGTF trust bundle as well as trusted by most modern web browsers. We also improved our internal and external documentation [R31], and the depth at which knowledge is shared among the GOCDB team.

5.3.2 Summary of service enhancements

In the last year, we have completed the outstanding tasks on the roadmap:

- "Improve configuration management to ensure the long-term stability of the service." and "A second, configuration managed, production instance of GOCDB will be deployed behind our load balancer."

  - The GOCDB service has been migrated to a new, highly available, configuration managed architecture. This allows many updates to GOCDB to be made transparently to the user, improving the service availability and reliability. The extensive use of configuration management also allowed us to deploy a new pre-production instance which allows us to test new versions of GOCDB, new features and changes before they enter production.

- "Deploy an EOSC-hub specific view on the data in GOCDB"

  - Over the course of the project, functionality has been added to the software to allow URL based, community specific, views of GOCDB to be deployed.

  - These views grant the ability to customise the information available in the portal and API by default, the language used within the portal and the look of the portal. A production and pre-production view for EOSC-hub have been deployed at https://gocdb.eosc-hub.eu/portal and https://gocdb-preprod.eosc-hub.eu/portal respectively.

  - These views provide, as shown in Figure 5.8, an EOSC-hub centric GOCDB whilst: reducing the overhead to resource providers of maintaining information in multiple GOCDB; reducing the overhead needed to run the instance, as in future the instances can be consolidated onto the same underlying hosts; and maintaining a single source of truth across multiple, related, e-infrastructures.
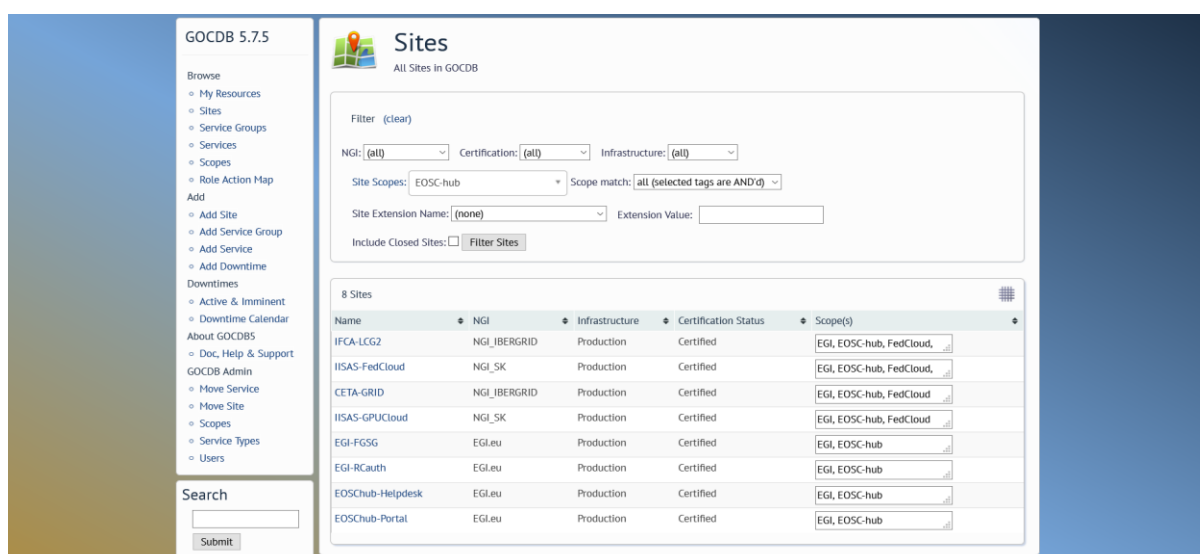
Fig.5-8 - GOCDB EOSC centric view

- "The functionality of the Write API will be expanded" and "Improvements and modification of 'Reserved Scopes' handling"
  - To meet expanding use cases from WLCG, the Write API was improved to support editing of services programmatically. The handling of "Reserved Scopes" has also been improved to allow scopes to be marked as reserved by a GOCDB administrator via the Admin interface. These new features will be released in GOCDB 5.8.

5.3.3 Future plans

GOCDB will continue to be operated and developed in future projects, such as EGI-ACE and EOSC-Future. We will expand our integration with EGI Check-In to include access to GOCDB API and integrate with the WLCG IAM. We will allow account linking with GOCDB for users who use both X.509 certificates and identities provided by EGI Check-In and WLCG IAM. We will also evolve the database abstraction layer to allow large scale use of the Write API.

5.4 Data Project Management Tool

The *Data Project Management Tool* (DPMT) is primarily used by the EUDAT Collaborative Data Infrastructure (EUDAT CDI) to support the operation of EUDAT services providing mainly storage - and to a lesser extent compute - resources. It is used to keep track of service and resource requests as well as offerings, deployed services and service components, and various other information needed to ensure the proper functioning of the CDI. All activities are grouped within projects connecting customers and their needs with providers and their offerings. A detailed description of the DPMT service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

5.4.1 Maintenance activities

General maintenance and operation have continuously been provided throughout the reporting period. No downtimes were required.

5.4.2 Summary of service enhancements

The maintenance activities during the last reporting period were focused on migration of the DPMT service software stack to the Python 3. The migration of the DPMT has been successfully accomplished.

Another important activity was focused on the introduction of a fully featured REST-API covering all aspects of the application. Not only does this allow for machine agents to access all information in machine readable format (assuming appropriate privileges have been granted to the agent) but also to update/change existing entries or to add new ones. Initially, it was planned to use this new feature to enable information to flow from other support services such as the Operations Portal or the Marketplace (e.g., to forward new orders that request services provided by the EUDAT CDI) but in the meantime a different approach was taken to implement this functionality (more details are given in Section 5.7.2).

To illustrate the new functionality by way of example consider a search request for all service options provided by the EUDAT CDI in the context of EOSC. Issuing the following call from a command line:

curl https://dp.eudat.eu/catalog/@search?Subject%3Alis=Service+option -H 'Accept: application/json'

A JSON-formatted output of this call is shown in screenshot below:

```
    "@id": "https://dp.eudat.eu/catalog/@search?Subject%3Alis=Service+option",
    "items": [
        {
        "@id": "https://dp.eudat.eu/catalog/B2FIND/options/researchers",
        "@type": "Document",
        "description": "Public service supporting cross disciplinary research, No
registration required.",
        "review_state": "external",
        "title": "B2FIND for Researchers"
        },
        {
        "@id": "https://dp.eudat.eu/catalog/B2FIND/options/data-managers",
        "@type": "Document",
        "description": "A local instance to use the B2FIND technology.",
        "review_state": "external",
        "title": "B2FIND for Data managers"
        },
        {
        "@id": "https://dp.eudat.eu/catalog/B2FIND/options/data-providers",
        "@type": "Document",
        "description": "Data repository owners can make their research data collections
stored in existing data repositories harvestable and discoverable via public B2FIND
service. ",
        "review_state": "external",
        "title": "B2FIND for Data providers"
        },
        {
        "@id": "https://dp.eudat.eu/catalog/B2HANDLE/options/researchers",
        "@type": "Document",
        "description": "Unique and persistent reference which are globally resolvable,
Network of service providers for high available PIDs and PID services, Sustainability
framework to sustain PIDs over long periods of time.",
        "review_state": "external",
        "title": "B2HANDLE for Researchers"
        },
        {
        "@id": "https://dp.eudat.eu/catalog/B2HANDLE/options/managers",
        "@type": "Document",
        "description": "B2HANDLE is the distributed service for minting, storing,
managing and accessing persistent identifiers (PIDs) and essential metadata (PID
records) as well as managing PID namespaces.",
        "review_state": "external",
        "title": "B2HANDLE for Data/Communiy Managers"
        },
```

A more detailed response can be obtained for individual services by calling, e.g.:

curl https://dp.eudat.eu/catalog/B2DROP/options -H 'Accept: application/json'

These example queries show how detailed, machine readable responses are now available via the API.

5.4.3 Future plans

The maintained and improved DPMT will continue to be used in follow-up projects such as the EU-funded project DICE (https://www.dice-eosc.eu), which provides European storage capacity and resources for the data management infrastructure for the EOSC.

5.5 Data Management Planning Tool

A detailed description of the Data Management Planning (DMP) service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A. The DMP tool is a collaboration between EOSC-Nordic and OpenAIRE. The OpenAIRE group worked on the front-end and back-end components of the DMP tool (shown in red in Figure 5-9 below) and the EOSC-Nordic team worked on the eestore (shown in orange in Figure 5-9). In addition, the EUDAT easyDMP Data Management Planning service is also used as a comparison.

Fig.5-9 High-level Architecture of Data Management Planning Service its components and their interactions. The components in the orange box are developed by OpenAIRE and those in the red box by EOSC-hub (diagram courtesy of G. Kakaletris).

5.5.1   Maintenance activities

During this reporting period the eestore and easyDMP underwent a migration from python2 to python3 and also more streamlined deployment in Docker containers managed by Kubernetes. Updates to the support for registries used by the eestore was also carried out.

5.5.2 Summary of service enhancements

The adoption of the RDA machine actionable DMP schema can be considered a service enhancement. It enables different DMP services to interoperate making it possible for consumers of DMPs to only need to support one schema. The first version of the implementation of the RDA schema in easyDMP enabled a plan to be exported from easyDMP and imported into another DMP tool (the Data Stewardship Wizard https://ds-wizard.org/ which was achieved during the RDA hackathon https://github.com/RDA-DMP-Common/hackathon-2020). This work has resulted in updates to the RDA schema and work continues to incorporate the latest version of the schema.

The eestore documentation has been updated and scripts have been improved to make the installation of the eestore locally, or in Docker containers much easier.

The DPMT provides access to EUDAT services and the integration of the eestore to be able to fetch information was investigated. The DPMT developers provided a means for the eestore to access information, and it was possible to access information on EUDAT services and present that through the eestore. This first version worked, and the data was visible in the easyDMP service that interfaced to the eestore (in principle any DMP service can interface to the eestore). There are updates to the DPMT that will require updates on the eestore side to handle the data.

With the DPMT team we had discussions about how we could update the DMP with new data when services have been provisioned, but the tools to allow this update are not yet in place. We plan to pursue this option in the future.

5.5.3 Future plans

The eestore has proved to be useful for openDMP and also for easyDMP. In another project, we plan to continue the work on integration with the DPMT to provide access to information on EOSC services. We are also pursuing better interoperation of easyDMP with services in the Nordics (such as storage provisioning) that will be able to take advantage of information in the DMP thus reducing duplication (researchers would only need to supply information in the DMP and services that require that information, such as project title, description, PI etc would be able to fetch it from the DMP).
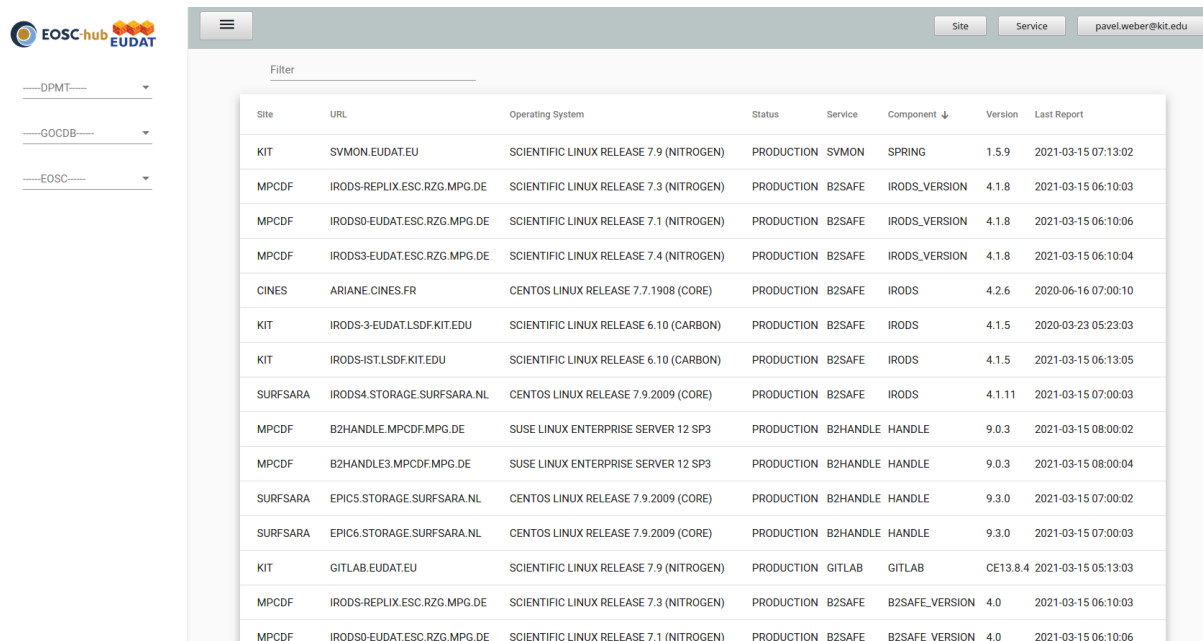
5.6 Service Versions Monitoring Tool

The Service Versions Monitoring Tool (SVMON) harvests the metadata of the service instances running at different sites and provides it via dashboard to the end-users. The tool is intended to perform the automatic updates of Configuration Management System and facilitate the Change Management Process providing the up-to-date information of the services running in the infrastructure. A detailed description of the SVMON service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

5.6.1 Maintenance activities

The maintenance activities were focused on multiple bug fixes in both front and back ends. The tool is based on Java Spring Framework with underlying MySQL Database running in High Availability mode.

5.6.2 Summary of service enhancements

Site	URL	Operating System	Status	Service	Component ↓	Version	Last Report
KIT	SVMON.EUDAT.EU	SCIENTIFIC LINUX RELEASE 7.9 (NITROGEN)	PRODUCTION	SVMON	SPRING	1.5.9	2021-03-15 07:13:02
MPCDF	IRODS-REPLIX.ESC.RZG.MPG.DE	SCIENTIFIC LINUX RELEASE 7.3 (NITROGEN)	PRODUCTION	B2SAFE	IRODS_VERSION	4.1.8	2021-03-15 06:10:03
MPCDF	IRODS0-EUDAT.ESC.RZG.MPG.DE	SCIENTIFIC LINUX RELEASE 7.1 (NITROGEN)	PRODUCTION	B2SAFE	IRODS_VERSION	4.1.8	2021-03-15 06:10:06
MPCDF	IRODS3-EUDAT.ESC.RZG.MPG.DE	SCIENTIFIC LINUX RELEASE 7.4 (NITROGEN)	PRODUCTION	B2SAFE	IRODS_VERSION	4.1.8	2021-03-15 06:10:04
CINES	ARIANE.CINES.FR	CENTOS LINUX RELEASE 7.7.1908 (CORE)	PRODUCTION	B2SAFE	IRODS	4.2.6	2020-06-16 07:00:10
KIT	IRODS-3-EUDAT.LSDF.KIT.EDU	SCIENTIFIC LINUX RELEASE 6.10 (CARBON)	PRODUCTION	B2SAFE	IRODS	4.1.5	2020-03-23 05:23:03
KIT	IRODS-IST.LSDF.KIT.EDU	SCIENTIFIC LINUX RELEASE 6.10 (CARBON)	PRODUCTION	B2SAFE	IRODS	4.1.5	2021-03-15 06:13:05
SURFSARA	IRODS4.STORAGE.SURFSARA.NL	CENTOS LINUX RELEASE 7.9.2009 (CORE)	PRODUCTION	B2SAFE	IRODS	4.1.11	2021-03-15 07:00:03
MPCDF	B2HANDLE.MPCDF.MPG.DE	SUSE LINUX ENTERPRISE SERVER 12 SP3	PRODUCTION	B2HANDLE	HANDLE	9.0.3	2021-03-15 08:00:02
MPCDF	B2HANDLE3.MPCDF.MPG.DE	SUSE LINUX ENTERPRISE SERVER 12 SP3	PRODUCTION	B2HANDLE	HANDLE	9.0.3	2021-03-15 08:00:04
SURFSARA	EPIC5.STORAGE.SURFSARA.NL	CENTOS LINUX RELEASE 7.9.2009 (CORE)	PRODUCTION	B2HANDLE	HANDLE	9.3.0	2021-03-15 07:00:02
SURFSARA	EPIC6.STORAGE.SURFSARA.NL	CENTOS LINUX RELEASE 7.9.2009 (CORE)	PRODUCTION	B2HANDLE	HANDLE	9.3.0	2021-03-15 07:00:03
KIT	GITLAB.EUDAT.EU	SCIENTIFIC LINUX RELEASE 7.9 (NITROGEN)	PRODUCTION	GITLAB	GITLAB	CE13.8.4	2021-03-15 05:13:03
MPCDF	IRODS-REPLIX.ESC.RZG.MPG.DE	SCIENTIFIC LINUX RELEASE 7.3 (NITROGEN)	PRODUCTION	B2SAFE	B2SAFE_VERSION	4.0	2021-03-15 06:10:03
MPCDF	IRODS0-EUDAT.ESC.RZG.MPG.DE	SCIENTIFIC LINUX RELEASE 7.1 (NITROGEN)	PRODUCTION	B2SAFE	B2SAFE_VERSION	4.0	2021-03-15 06:10:06

Fig.5-10 SVMON Dashboard.

The major goals for SVMON service, like integration with AAI, enhancement of HTTP API, integration of the service with GOCDB and DPMT, token-based authentication for SVMON clients for secure communication have been achieved during the first two years of the project. Figure 5-10 shows one of the SVMON dashboards which provides a snapshot of services and their components with current release versions. During the last year of the project, the work was focused on the improvement of the SVMON client, consolidation of the SVMON repositories and updates of the underlying Java Spring Framework.

5.6.3 Future plans

In the future projects we plan to extend the coverage of the SMVON and include more services. The SVMON client will be enhanced, and the list of services supported by the client will be extended. The integration of ARGO messaging system is planned to enable easy integration of SVMON with other services. Further enhancements of the SMVON UI are planned.

5.7 Integration activities

5.7.1 Integration of Operations Portal with Marketplace

5.7.1.1 Summary of integration activities

The integration of Marketplace is ensured through 2 mechanisms: the JIRA API and the marketplace API. JIRA API is ensuring that all data related service orders operated by both tools are synchronized. And the marketplace API is providing up-to-date information about service providers especially the way to contact them.

5.7.1.2   Identified integration gaps

SOMBO is providing different interfaces to negotiate the resources requested by users and the resources proposed by providers. This negotiation process is not reflected in the Marketplace. The plan is to provide this information through an API exposed to the Marketplace.

5.7.1.3   Future plans

As described previously SOMBO should expose - through an API - information about the negotiation process with resources providers and make it available to the end-user.

5.7.2   Integration of Operations Portal with DPMT

5.7.2.1   Summary of integration activities

The context of integration is the following: The Operations Portal is dealing with the management of the service orders registered in the marketplace. The SOMBO module is supposed to make the link between service orders and service providers. However, for resources offered by EUDAT this link, between service orders and service providers, is managed by the DPMT.Consequently if both components can interact together it will be a great progress.

The integration efforts between Operations Portal and DPMT have focused on 3 aspects.

The first one is to retrieve information from the DPMT with the use of the DPMT API. This functionality is now fully operational.

The second aspect is to define the minimal structure of information to pass from the Operations Portal to the DPMT and we have reached an agreement on the structure of the information to be provided during the last reporting period. Figure 5-10 shows the proposed snippet of the message format for exchange for completeness.

```
  ▼ additional: [
     ▼ {
           key: "request_type",
           value: "Development"
        }
     ],
     allowDiscussion: false,
     compute_resources: [ ],
     contributors: [ ],
     description: "For demonstration purposes only",
     endDate: "2022-03-31T00:00:00+00:00",
     expirationDate: null,
     id: "b2share-demo-request",
     identifiers: [ ],
     pid: null,
     portal_type: "ServiceRequest",
  ▼ preferred_providers: [
     ▼ {
           path: "providers/JUELICH",
           title: "JUELICH",
           uid: "7703449277cb4011972ef7223776f472"
        }
     ],
     registered_service: [ ],
     relatedItems: [ ],
     resource_comment: "<p>If applicable and already known how much ɪ
     later).</p>\n",
     rights: "",
  ▼ service: [
     ▼ {
           path: "catalog/B2SHARE",
           title: "B2SHARE",
           uid: "d2cb1386f880484d9569342bc61f7d48"
        }
     ],
  ▼ service_hours: [
     ▼ {
           path: "services/SLA/a-level-service-time-08-16",
           title: "A level: service time 08-16",
           uid: "89b776138bf54e5aa22d501ef74f1298"
        }
     ],
  ▼ service_option: [
     ▼ {
           path: "catalog/B2SHARE/options/default",
           title: "B2SHARE: Default",
           uid: "7f3dd440b80b4f0fa06854842d2cc158"
        }
     ],
     startDate: "2020-04-01T00:00:00+00:00",
  ▼ storage_resources: [
     ▼ {
           "storage class": "online+",
           unit: "TB",
           value: "20"
        }
     ],
     subject: [ ],
     text: "<p>More explanatory text could go here</p>",
     ticketid: "12345",
     title: "B2Share Demo Request",
     uid: "f367ca818f7a4a98b7d2611a4b55ebc8"
  }
```

Fig.5-10 A snippet of the message format for exchange of order information

The third point is to agree on the mechanism to exchange information and we concluded that a generic solution could be to use the ARGO Messaging system.

On the Operations Portal side with the help of a plug-in of our web service we are able to publish and consume information within the ARGO messaging system.

5.7.2.2 Integration gaps

The level of information coming from Marketplace is not sufficient to structure and properly fill the message sent to ARGO. So, an additional form is necessary in SOMBO to add the missing items to allow unambiguously identify certain elements within the message such as services (including their components and options), customers, providers, and users.

The proposal to overcome this gap by implementation of a "meta registry" which could provide uniform identifiers for all meta data published in the ARGO Messaging Service is still under discussion. The implementation of such a "meta registry" or any other solution would require a general integration strategy with respect to third order management systems and exchange of the order information across multiple systems. The implementation of the federated configuration management system could facilitate the exchange of order information between multiple order management systems.

5.7.2.3 Future plans

As described previously SOMBO needs to be completed with an additional form to be sure to expose all the minimal set of information required by the DPMT. This technical implementation would require a general strategy and standard approach for integration of third management systems.

5.8 Configuration of Federation Core Services and Configuration Management Plan

In this section we outline the work which has been accomplished with cooperation from WP4 in order to establish the general service data model which can be applied to the description of the EOSC-Core services and any other services in the EOSC.

During the assessment of the EOSC-hub Configuration Management System and related tools we came to conclusions, that the current services involved in the configuration management are missing the overall uniform service representation and general approach to federated operation of the Configuration Management process. This gap complicated any integration activities at technical level as many tools have their own database schemes often with lacking information on relations between services and service components.

To overcome this gap, we worked together with WP4 SMS experts to establish the service data model and configuration management plan which should provide guidelines for implementation of the EOSC federated configuration management system. In the following sections we provide a short description of the service data model and a few examples of its application to some federation services. A full description of the model has been recently published as a separate document and is available at [R32].

5.8.1 Service Data Model

Considering the complexity and diversity of the EOSC federation services with multi-tenant architecture and multiple services providers and service organisation, we focused on development of the flexible and modular service data model which can be applied to different services and their components without strict definitions for metadata schemas and relationships which should be proposed during implementation phase.

The goal was to provide simple guidelines and principles for the implementation of Configuration Management, using an unambiguous description of the resources and services, together with their components and relationships, resulting in a scalable Configuration Management System that can be easily adapted and used by all organisations, resource and service providers contributing to EOSC.

To establish the model, we followed a few major steps: collection of the requirements from the use cases, introduction of the general structure of the configuration items and their associated attributes, and finally definition of the relationship types between the Configuration Items (CI).

Fig.5-11 - General Service Data Model. The arrows and multiplicities between the CIs show possible relationships of the corresponding CIs which have to be chosen appropriately according to the governance rules of the implementation. An exception is made for the Owner CI and Provider CI, with their unique relationships towards the remaining CIs.

Figure 5-11 illustrates the general Service Data Model and shows the CIs which build the main constituents of this model, together with the possible associations (relationships). Self-loops indicate that the respective CI can be related to other CIs of the same type.

We consider the following CIs as main constituents of the service data model:

- Service
- Service Component (a group of CIs in the current model)
- Infrastructure
- Software
- User
- Provider

It has to be stressed that a limited number of CIs in this model has been chosen for the sake of simplicity. In the given model, the "Service Component" is defined as a group of CIs, which are parts of the service and enhance it, but do not create a value by themselves. For example, a service component could be a logical function or module of the main service. Thus, we can consider different types of service components depending on concrete implementation of the EOSC service or resource.

The Infrastructure CI group could include server, storage, virtual systems, docker containers, any running applications. Depending on the chosen complexity of the CMDB implementation, this analogy can be applied to any of the above CIs, which can become a group or container of different CIs types (sometimes referred as CI classes), instead of a single CI.

By Software CI we refer to any software product or software package with a given release version associated with a service. Again, this definition can be extended to multiple CIs categories e.g., operating systems, databases, software frameworks depending on the requirements of the particular project.

A User CI as a group includes multiple CI types of users, who could be identified by their roles, association to any group, organisation, service etc., for example, a User CI could include researchers, students, resource/service owners, resource managers etc.

The model also provides specifications for the types of the relationships between Configuration Items and describes the distributed composite services which contain multiple monolithic services and service components as building blocks.

5.8.2 Application of Service Data Model to the EOSC-Core candidate services

In this section we provide a few examples of application of the proposed service model to a few EOSC-Core candidate services. The colours of the boxes in the following examples are taken in accordance with Figure 11 and the colour-encoding of the CIs given there. Frames around CIs group them together in the following sense: an association attached to a frame signifies that the attached association, displayed by an association arrow or by a composition diamond which points into the direction of the text, holds for all elements inside this frame.

EOSC Portal
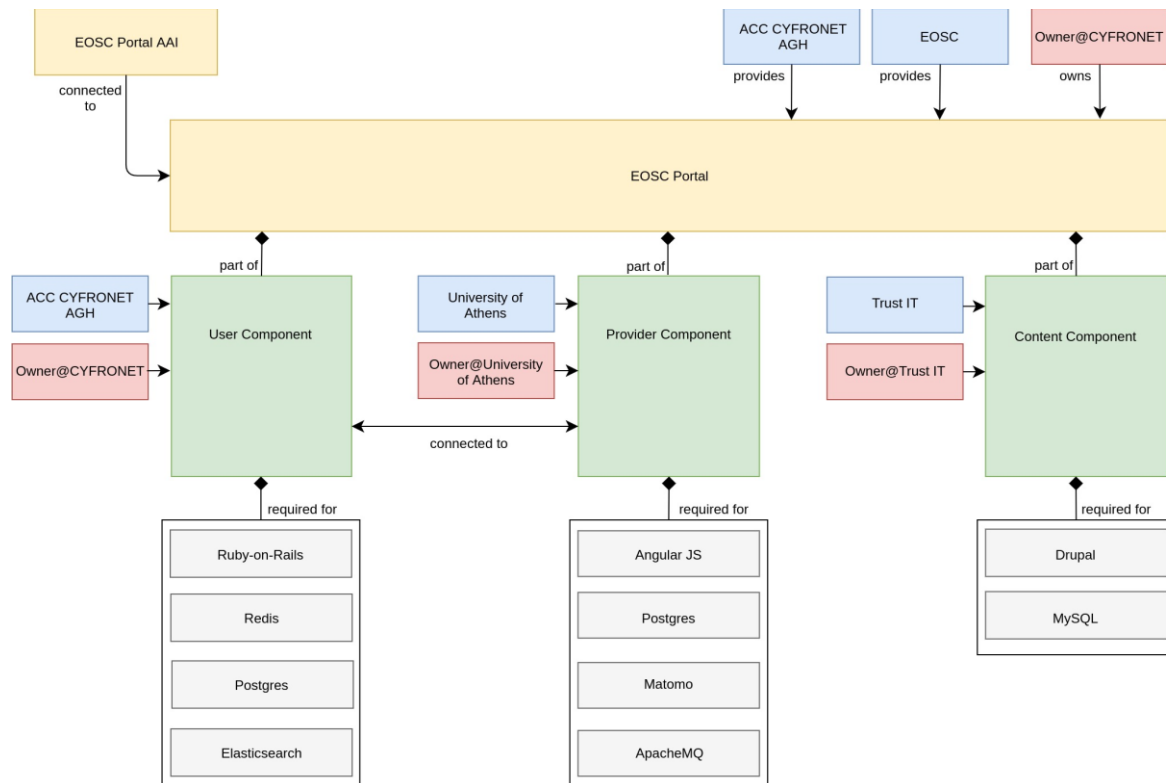
Fig. 5-12 - EOSC Portal diagram.

Figure 5-12 shows the configuration items of the EOSC Portal and their relationships according to the service model. The "connected to"-association between the User Component and Provider Component, which points in both directions, indicates that the two components are interconnected. Frames around CIs signify that the attached association holds for all elements of this frame. For example, each Software CI inside the grey frame is required for each Service Component inside the green frame

The EOSC Portal is composed of three different elements:

- User Component (EOSC-hub Marketplace), which publishes EOSC service catalogue and facilitates the service ordering.

- Provider Component, which manages the onboarding of new EOSC providers and their services.

- Content Component, which is responsible for the management of the web layout of the EOSC Portal.

EOSC-hub Helpdesk

EOSC-hub helpdesk, based on xGUS, is a single point of contact for all EOSC users for requesting help or for fixing issues. The EOSC-hub helpdesk is integrated with both EUDAT and EGI Helpdesk systems, allowing to handle the management of tickets received on xGUS and assign them either to EUDAT or EGI/WLCG infrastructures.

Figure 5-13 shows how the Data Model can be applied for the description of the Helpdesk. The EOSC-hub Helpdesk is a composite service, which consists of three different helpdesk systems: EGI/WLCG GGUS, EOSC Helpdesk (candidate) and EUDAT Helpdesk. As shown in Figure 5-13 the EGI/WLCG GGUS and EOSC Helpdesk are sharing the same software stack. For better readability, we grouped the Owner and Provider CIs into a dashed frame and combined the relationship texts at the corresponding arrows.
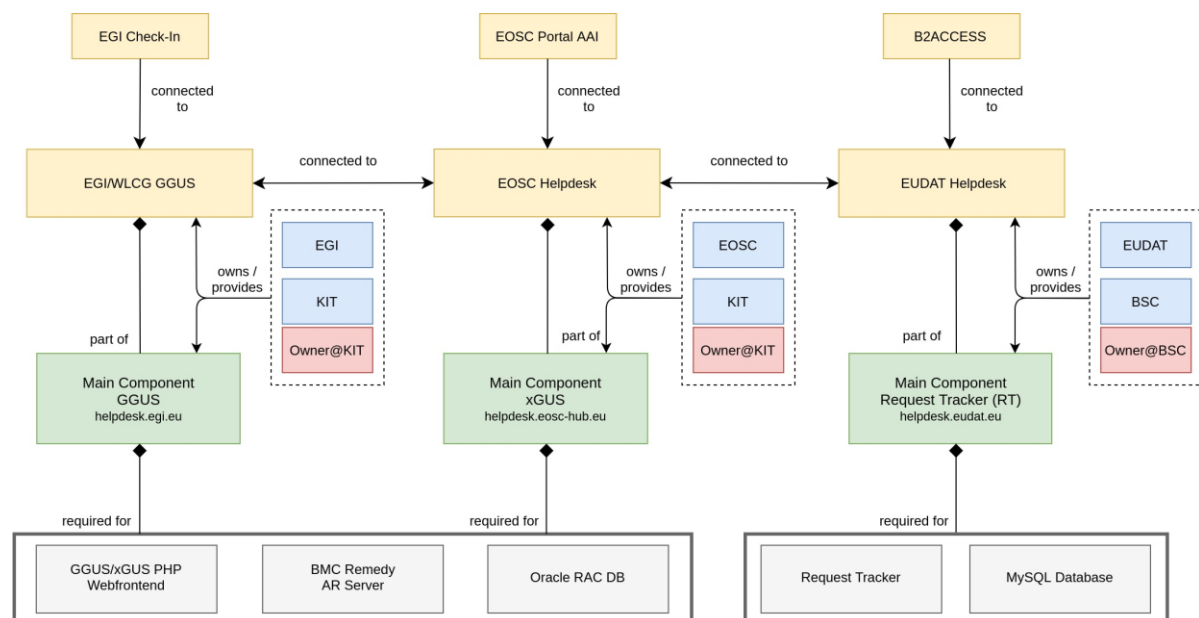
Fig. 5-13 - EOSC Helpdesk diagram.

EOSC Portal AAI

The EOSC Portal AAI is part of the infrastructure layer of the EOSC-hub AAI. As an infrastructure proxy, it is connected to multiple Community AAIs to allow researchers to access the underlying services and resources using their community identity, including their roles and other authorisation-related information managed by the community. In addition to the Community AAIs, the EOSC Portal is connected to the upstream home organisation IdPs (e.g., from eduGAIN/social) to enable researchers to access services and resources as members of their home organisation. As shown in Figure 5-14 the EOSC Portal AAI consists of four main service components: Federation registry, High Availability & Balancing service, IdP/SP proxy, and discovery service.

Fig. 5-14 - EOSC Portal AAI diagram.

Order Handling System

The EOSC-hub Order Handling System (OHS) allows customers to access the central EOSC-hub Marketplace, discover services and resources using the intelligent search and filtering mechanism, place an order and track it until its fulfilment and service or resource delivery.
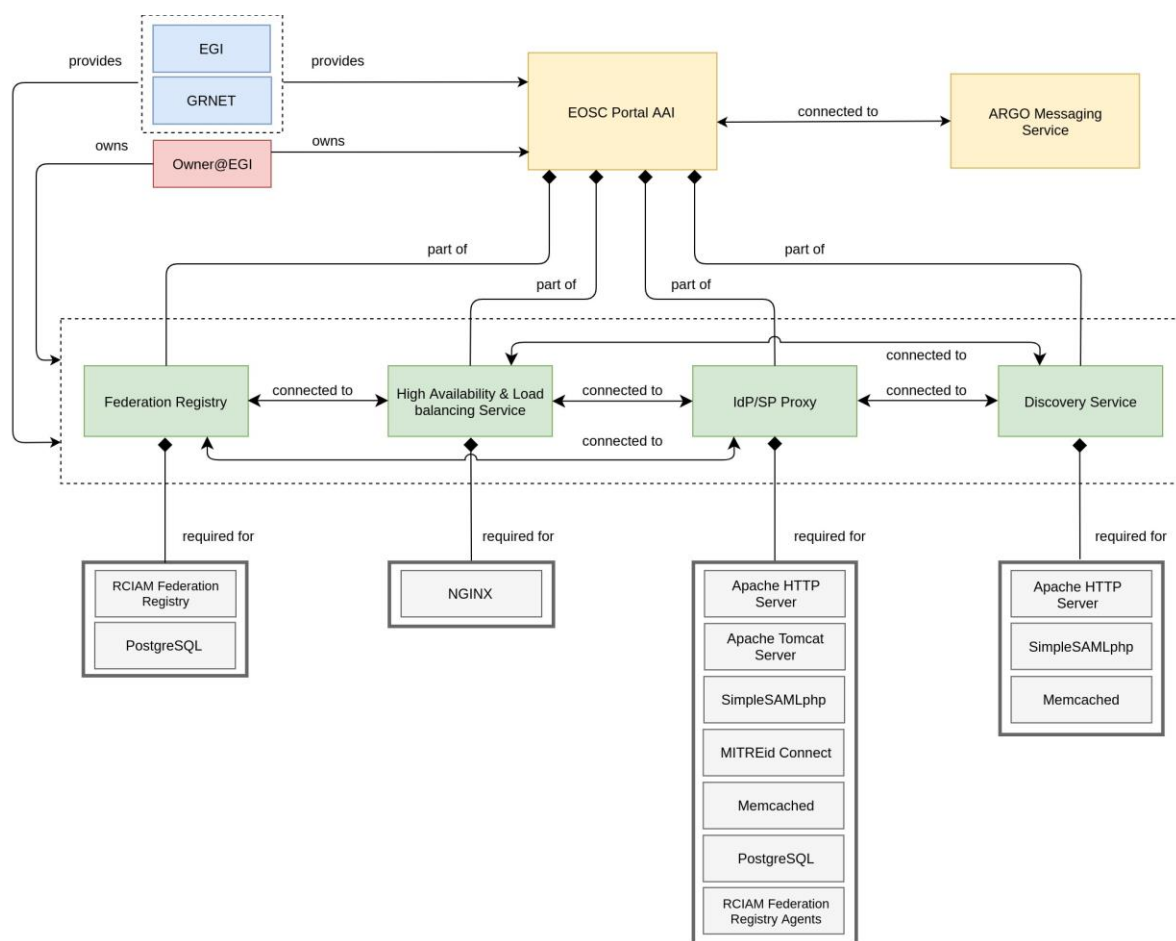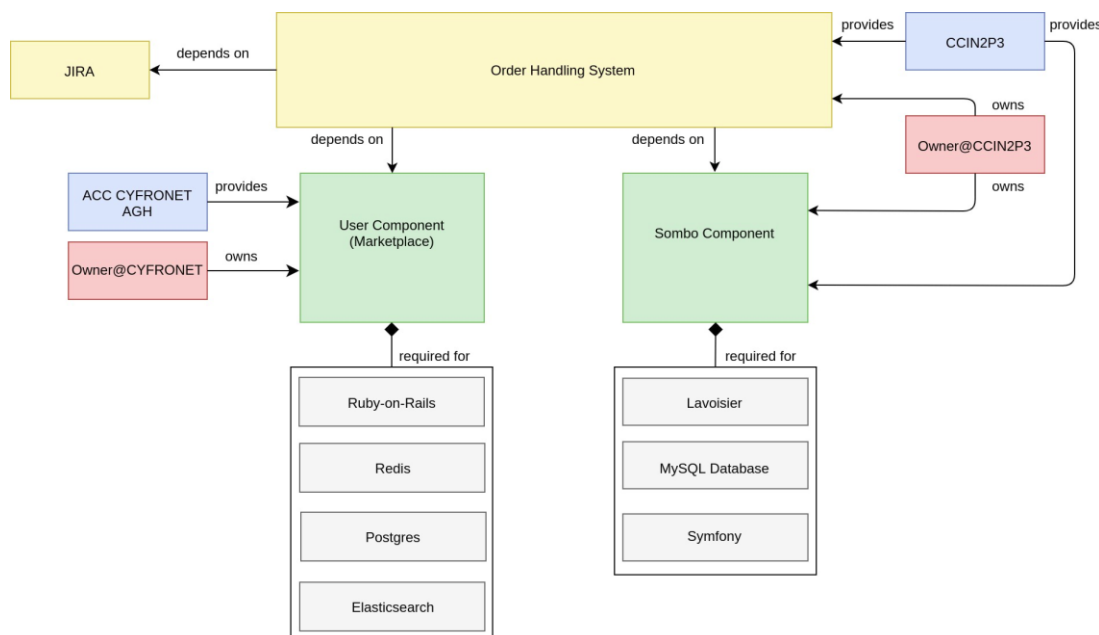

Fig. 5-15 - Order Handling System diagram.

As shown in Figure 5-15 the Order Handling System comprises two main components namely User Component (Marketplace) and Service Order Management Back Office (SOMBO) Component. The configuration diagram for OHS follows the configuration of composite service and demonstrates the configuration case, for which the composite service is built on service components, which are initially the parts of Operations Portal and EOSC Portal. In this way, the service data model describes the fact of sharing of service components between different services stressing the composability and combined usage of interoperable services and their components.

A few examples provided here demonstrate a successful application of the developed service data model to many federation services also ones with complex distributed architecture. It provides a basis for uniform description of multiple services and resources in the EOSC. The description of the other major candidate service for EOSC-Core together with detailed model specification is provided in the Configuration Management Plan[R32].

5.9 Summary and outlook

The activities carried out in this task were focused on several operational services - Operations Portal, GOCDB, DPMT, DMP and SVMON. Each service has undergone significant improvements with respect to interoperability, support of the operational processes, APIs. This section provides a short summary for the major results and enhancements for the services for the whole project time.

A remarkable amount of progress has been made in the development of the Operations Portal. A functional component of the Operations Portal - Service Order Management Back Office has been

developed from scratch and successfully deployed in production as a part of the Order Handling System according to established integration and development plan for Order Handling System. SOMBO has been integrated with Marketplace to receive, manage, and dispatch orders to multiple service providers. A new module has been put in place to collect metrics about orders in order to generate reports for the European Commission. A new instance of the Operations Portal in the EOSC scope has been deployed.

Many enhancements have been implemented for GOCDB including extended functionality of APIs, implementation of new scopes, integration with EOSC-hub AAI. In addition, a significant effort has been undertaken to provide High Availability setup for the service and migration to configuration managed architecture. The deployment of EOSC-hub specific view of GOCDB provides a consolidated list of EOSC federation services and will become one of the central repositories for the EOSC configuration management system in the future projects related to EOSC.

The Data Project Management Tool (DPMT) has significantly improved its interoperability by implementation of REST API covering all aspects of the application. A new functionality for resource and service offering have been implemented. The Storage Accounting Records (StAR views) exposed by DPMT for upstream consumption, e.g., by the EOSC Accounting Portal, have been enriched through the addition of further context information and become a basis for the initial integration with EOSC accounting system.

The Data Management Planning Tool consists of a front-end component ARGOS with a web UI, through which data management plans are managed and monitored, and a back-end service EEstore that collects and provides information from a variety of data service registries. The EEstore service has undergone a rearrangement of the API to make things easier to extend to include new registries. An integration plan with DPMT has been established. Another enhancement which has been carried out in collaboration with OpenAIRE is the adoption of the RDA machine actionable DMP schema. It enables different DMP services to interoperate making it possible for consumers of DMPs to only need to support one schema.

The Service Monitoring Tool has been integrated with EOSC AAI, GOCDB and DPMT and enriched with multiple dashboards to display the information about services and the operational metadata.

It has to be noted that the initially planned integration between GOCDB and DPMT has not been implemented. Although both services significantly improved their interoperability and technical integration was possible, the introduction of the EOSC Portal and change of focus to the central catalogue and integration of Marketplace with it made these plans obsolete. In addition, the lack of the general vision on the architecture of EOSC configuration management system and its components at the beginning of the project prevented the agreement on the integration of GOCDB and DPMT. With the introduction of the Configuration Management plan [R32] we have prepared a basis for the successful implementation of the federated EOSC configuration management system in the future projects.

6 Monitoring, Accounting, Messaging and Security Tools

6.1 Overview

This chapter provides the description of maintenance and integration activities performed for the ARGO Availability and Reliability Monitoring Service, ARGO Messaging Service, Accounting Repository, Accounting Portal, and Security Tools. We also describe what the next steps are towards the integration plan for each tool/service.

6.2 Accounting Repository

The Accounting Repository service is implemented using a software collection known as APEL. APEL is a computer resource usage accounting tool that collects, and stores compute (serial and parallel jobs), storage, and cloud resource usage data collected from Resource Centres of the EGI and EUDAT infrastructures. Accounting information is gathered from distributed sensors into a central Accounting Repository where it is processed to generate summaries that are then made available through the Accounting Portal.

A detailed description of the Accounting Repository service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

6.2.1 Maintenance activities

The Secure STOMP Messenger (SSM) component of APEL is used to transfer accounting records between sites. Since the release of the version that added support for the ARGO Messaging Service (AMS), there have been four other releases of APEL SSM that have steadily improved the integration with AMS to the point that it is as reliable as using SSM with a STOMP message broker service.

While compatibility of the APEL software with CentOS 8 has been demonstrated in development, further work on this has been suspended pending the outcome of high-level decisions on the future usage of CentOS within WLCG and scientific communities.

Work has taken place to improve the compatibility of the APEL software with Python 3; a milestone for this work was dropping support for the oldest version of Python that APEL was still supporting, making the codebase much more cross-compatible.

As in previous periods, there have been a number of small bug fixes developed for and patches applied to the software and systems used to run the service, and the latest IGTF Trust Anchor Distributions have been applied.

6.2.2 Summary of service enhancements

The latest version of APEL SSM that supports AMS has been rolled out to the production Repository server, enabling sites that were starting to have issues with the legacy message broker service to migrate to sending their records via AMS. There have also been some changes to the SSM software to improve its reliability and ease of use, particularly in relation to AMS, making it easier for users to switch to using AMS.

6.2.3 Future plans

While all of the APEL software supports RPM-based installation systems and some of it supports DEB-based systems, further work may be needed to ensure full compatibility with the latter depending on what direction is taken in relation to operating system usage in the near future.

Further work is required to make the codebase fully compatible with Python 3.

6.3 Accounting Portal

A detailed description of the Accounting Portal service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

6.3.1 Maintenance activities

In the maintenance of the portal activity, we improved the indexation of some tables to speed up response, added a new Cloud based graph to the Welcome Page in addition to the grid one, fixed the Date display on locales that are on the GMT - hemisphere (that caused small errors on the ending of a month only on non-european machines), and improved PDF creation, legend positioning and timestamping on some reports, like the WLCG ones.

Also, better unit support for Cloud UserDN and units in Storage accounting was introduced, periodical stale Record removing was done to fix and patch up publication problems with sites, maintenance of the SSM queuing system was done to ensure it was working all the time without interruptions, and also log and traffic analysis was done in order to support the capacity plan.

6.3.2 Summary of service enhancements

As for service enhancements, an AAI Implementation using mod_auth_mellon and SAML was implemented that allows users to be identified using different credentials and identity providers as VO or Site Admins in order to access the Restricted Views. Also, the cloud topology was separated from the HTC one in order to filter sites not in the FedCloud Scope in GOCDB, so only sites officially in FedCloud are displayed, since some sites published data without being registered

The topology backend was changed to support Computing Resource Information Catalogue (CRIC) from WLCG as a new topology source, and new WLCG long form federation names are also supported, as are Jupyter Notebook sites. Metrics in normal views now support multipliers, and the Tier1 REBUS report inherited from Rebus was removed as the new version is in CRIC.

EUDAT storage accounting was implemented as a static report that does not use inconsistent date information and Storage accounting was also implemented but waiting for summarisation to be moved to APEL to move to production.

6.3.3 Future plans

In the future, we plan to Improve performance and responsiveness of the portal specially the database backend, implement automated testing, for example using something like the Selenium framework [R33], and also implement memcached functionality to improve the performance of the server.

Also, we expect to create a separate model Layer Unit test suite and Automated regression testing facilities, improve the virtualization layer, complete the porting of the codebase to Python 3, evaluate other Web server possibilities and optimize queries in general.

Last, but not least, we expect to implement stress testing and security testing components.

6.4 Argo Service Availability and Reliability Monitoring

Service Availability and Reliability Monitoring is a key service needed to gain insights into an infrastructure, the applications, services, and even into processes/behaviours. ARGO monitors services by emulating typical user scenarios which allows them to infer the quality of service the actual user gets. It mimic's the actual end user behaviour without requiring special privileges or special configurations from the service provider side. As a result, ARGO offers near real-time status updates which allow both end-users and site admins to have an overview of the service offered at any given point in time. The major objective of the monitoring system is to quickly identify and correlate problems before they affect end-users and ultimately the productivity of the services, the infrastructure and finally the organization.

A detailed description of the ARGO monitoring service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

6.4.1 Maintenance activities

There is a standardized maintenance window every first Wednesday (devel instances) and Thursday (production instances) of each month. The maintenance is performed without downtime as all ARGO components operate in HA mode. These maintenance windows are used to apply regular operating system upgrades and stable releases. All necessary precautions (backing up the data etc) are taken care of beforehand by the monitoring team.

ARGO follows a development process that includes mandatory tests for checking the functionality and the quality, correctness of the software. This process consists of automated unit tests and code quality checks, running via a CI tool (jenkins). The argo team maintains a full replica of the production deployment (devel, staging instance) that is used for development, testing and integration validation of all the components that ARGO comprises (mon, poem, compute engine, API, and web UI).

6.4.2 Summary of service enhancements

ARGO Monitoring is a flexible and scalable framework for monitoring the status, availability and reliability of services provided by infrastructures with medium to high complexity. It uses the latest technology trends (etc apache flink, apache hdfs) as its main components to support Big Data analysis (transfer, store, stream, transform, analyse) and to offer near real time alerts. ARGO Monitoring is successfully used by both EGI and EUDAT infrastructures. Based on all these the work in the project has been focused on the development of the following features:

- A **unified web-portal** that combines services from a number of different providers/infrastructures.

- **Customer defined thresholds** which provide the ability to define custom thresholds for specific services/metrics/sites (for example to monitor the requirements of an SLA). This gives us the flexibility to provide different SLA targets to customers (e.g., different acceptable response time for a specific customer)

- **One Stop Shop** that simplifies and automates the operation and configuration of ARGO components so that it is able to update the data of existing tenants or to create and deploy new tenants, metrics, profiles by using the Argo Admin portal.

During the defined period new functionalities were introduced in the various components in order to support these features.

Argo Analytics Compute Engine: Argo analytics engine is the computational and analytics heart of ARGO. It is a distributed system, allowing automatic configuration and execution of real-time streaming jobs as well as batch ones to provide results and reports. During the defined period, apart from the migration of all python components to python3 a lot of new functionalities have been introduced in detail below:

- Changes in the computational model to better handle tenants with flat topologies that contain multiple instances of the same service that runs on a host.

- Changes in the computational pipeline to allow to enrich Availability Reliability (A/R) endpoint results with additional information (such as service urls, descriptions etc). This also led to various fixes on submission scripts to be able to use parameters and additional data or profiles optionally.

- Implementation of datastore index check mechanism which checks and ensures the proper index strategies have been deployed

- Implementation of new argo-web-api interface for the computational job to be able to retrieve all additional information (topologies, downtimes, weights, report configuration and various profiles) directly from web api instead of hdfs

ARGO POEM: Poem now represents a central point (UI) for managing most of the resources used in the ARGO monitoring engine. It follows the principal towards One-Stop-Shop functionality and using ARGO WEB-API as centralized. It is the central management interface for a tenant to prepare all the resources for the monitoring. POEM consists of SuperPOEM which is the management dashboard for the super-administrator and the POEM instance which is the tenant area for managing its own resources. "ARGO SuperPOEM" is introduced so that we can have a general view of all available tenants and resources shared between them. It is responsible for initiating and managing the tenants. It represents the central point for managing RPM repositories, probes, and metrics. It also has the functionality of a library of metrics. "ARGO POEM instance" is the dashboard of a tenant's resources management. The tenant can manage the users and create tokens for other services to use the poem resources. At the same time, it is now the duty of tenant admin to cherry pick between all available metric templates and create all resources needed to customize and bootstrap their monitoring. A number of profiles (metric, threshold, operations, and aggregation profiles) has to be configured via a friendly interface, for the computation of A/R and status of the tenant

infrastructure. In order to support different external sources POEM supports a) for authentication purposes a local account and both EGI-Checkin and B2ACCESS, b) for topology purposes data from CODB, DPMT and static json and csv files. When the profiles and the reports are ready the computations may start. Finally, as ARGO POEM resources are often referenced from report documents and wikis, "Resources Public pages'' were also introduced. Tenant resources such as configured metrics, metric profiles, aggregation profiles are now available as view-only resources served over an un-authenticated URL.

Here is complete list of features delivered with ARGO POEM:

- Support of reports, aggregation, and metrics profile

- Use of ARGO WEB-API as a centralized store for various resources

- Handling of RPM repositories with Nagios probes

- Versioning of packages and Nagios probes

- Mapping of probes and metrics with predefined metric configuration templates

- Mapping of metrics and service types and grouping of them into metric profiles

- Handling of aggregation profiles; definition of service type groups and logical operators within and between them so to define the status result deduction rules

- PostgreSQL with schemas for enabling of multi-tenancy

- Service type page with list of service types defined in tenants GOCDB-compatible services

- Flexible caching on the frontend

Argo-web-api is the central interface of the argo monitoring platform as a service which allows users, clients, and components to access a single source of truth, define new reports and computations, access results etc. It is implemented as a RESTful HTTP API service. During the defined period, many new functionalities have been introduced to the argo-web-api paving the way to transform it the central source of truth and orchestrator for all components of the argo monitoring system. In more detail:

- New interfaces and rest calls api to store and manage topology information. Connectors can directly access argo-web-api and store and manage different types of topology information per day per tenant. This information can be accessed and read using a plethora of advanced filters by other components. Also, reports have been updated to contain smart filters tied to the available topology. Each report automatically gets a subset of the available topology based on the filters it has defined.

- Manages operational parameters for various feeds such as topology feeds, weight feeds and downtime feeds which allow connectors to be autoconfigured by contacting argo-web-api.

- Argo-web-api displays a/r results from a top-down approach beginning with the higher-level groups (projects, groups), allowing users to drill down and explore the lower-level results (services, endpoints). Through user feedback it was clear that it was useful to provide a flat list with the available endpoint a/r results.

- The ability to quickly provide a list of issues containing results for all problematic endpoints of an infrastructure, through a new rest api call.

- A new set of api calls was implemented to allow more flexible management (search, add, remove, update and refresh credential for the users).

- Provide additional information per result item (for example provide url information on service a/r results etc).

- It was also updated (like all components) throughout the result calls to be able to support flat topologies. Also, a new mechanism was created for the web api to be able to provide additional information per result item (for example provide url information on service a/r results etc)

WEB UI: Two new types of indicators have been added into the Web UI. The first one is a page displaying all problematic results (ex. CRITICAL., WARNING) as shown in Figure 6-1. Results are gathered by group (site / servicegroups) and filters can be used to limit results to a specific name group or status. The second is a new page (Figure 6-2) showing all last results for a given metric. Status, and group name filters were added to help the users find the information they want. At the same time, detailed documentation was added for every action a user can perform (see Figure 6-3).
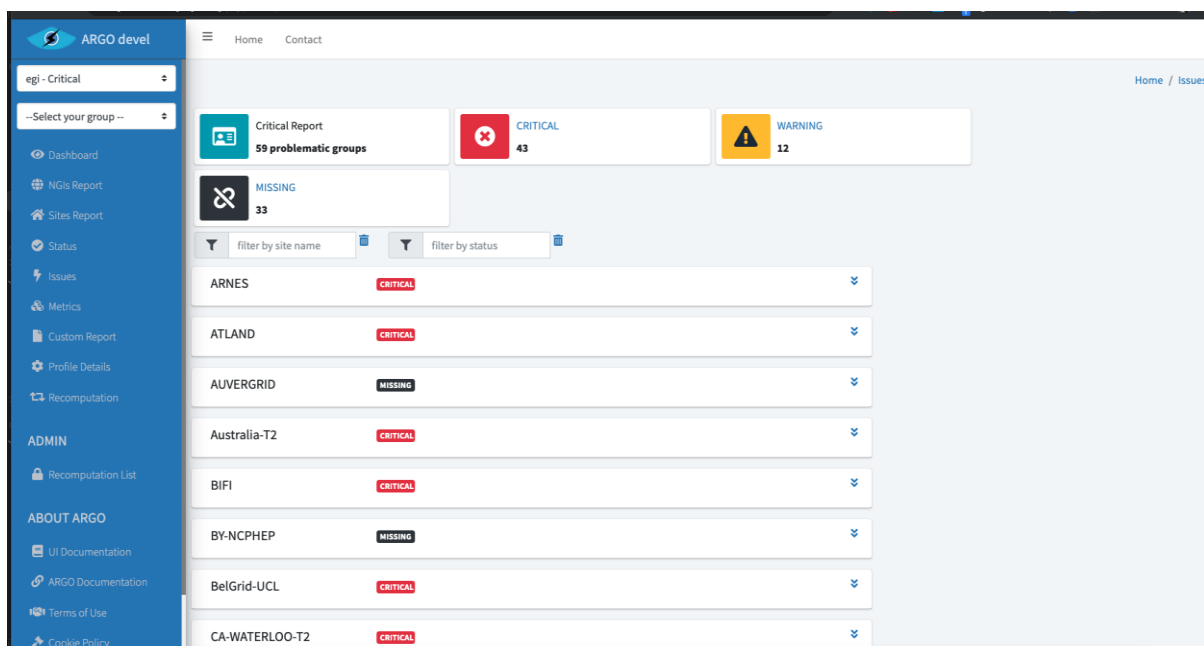
Fig.6-1 Problematic Sites View.
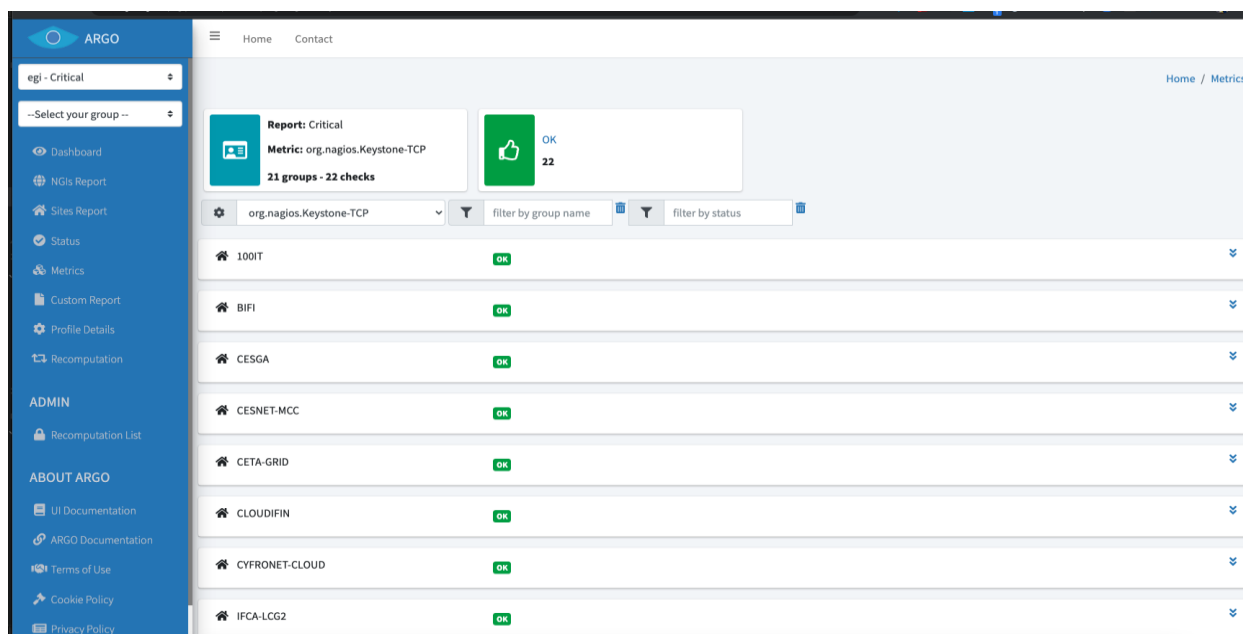
Fig.6-2 List by Service Metric View.
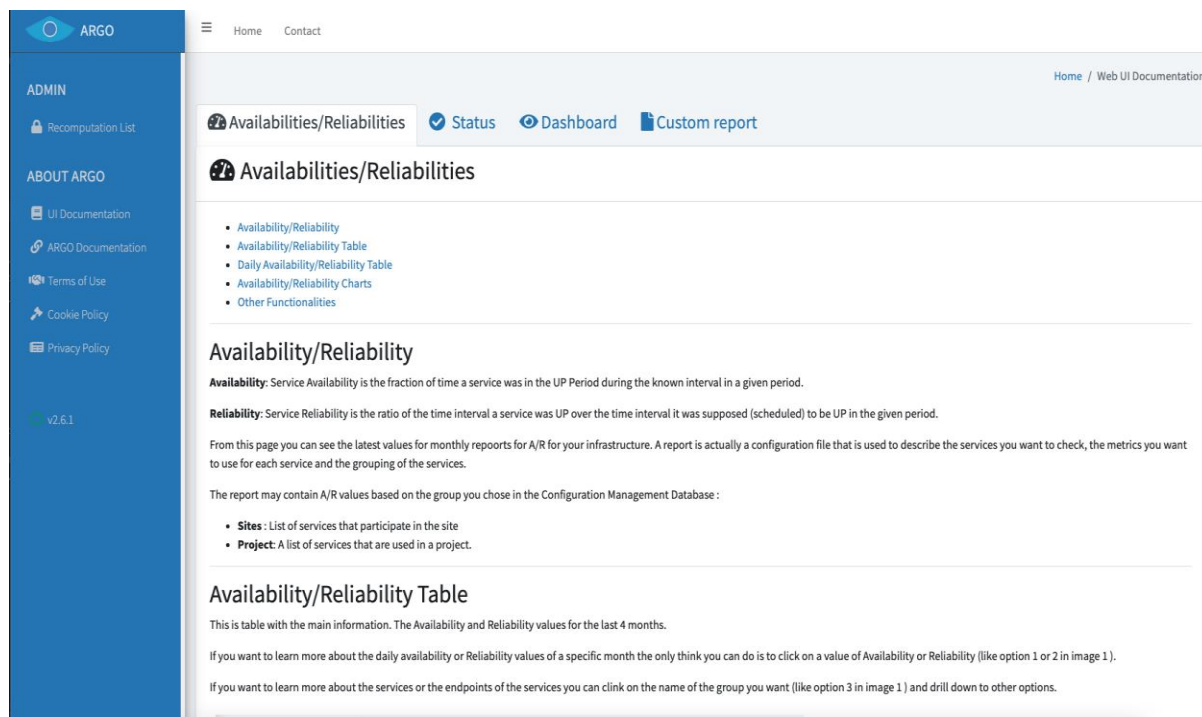
Fig.6-3 Argo Web UI Documentation.

6.4.3 Future plans

- Support the **unified web-portal** that will combine further services from a number of different providers/infrastructures.

- Add new features to **One Stop Shop** that simplifies and automates the operation and configuration of ARGO components so that it is able to update the data of existing tenants or to create and deploy new tenants, metrics, profiles by using the Argo Admin portal.

- Support the different types of Integration scenarios of the services with the EOSC

- Monitor Services / endpoints listed in EOSC portal

6.5 ARGO Messaging Service

The ARGO Messaging Service is a real-time messaging service that allows the user to send and receive messages between independent applications. It is implemented as a Publish/Subscribe Service. Instead of focusing on a single Messaging service specification for handling the logic of publishing/subscribing to the broker network the service focuses on creating nodes of Publishers and Subscribers as a Service. In the Publish/Subscribe paradigm, Publishers are users/systems that can send messages to named channels called Topics. Subscribers are users/systems that create Subscriptions to specific topics and receive messages. As shown in Figure 6-4, the current deployment of messaging service comprises a haproxy server, which acts as a load balancer for the 3 AMS servers running in the backend.

An initial description of the ARGO messaging service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

Fig.6-4 - Deployment of messaging service

Features

- **Ease of use**: It supports an HTTP API and a python library so as to easily integrate with the AMS.

- **Push Delivery**: AMS instantly pushes asynchronous event notifications when messages are published to the message topic. Subscribers are notified when a message is available.

- **Replay messages**: replay messages that have been acknowledged by seeking a timestamp.

- **Schema Support**: on demand mechanism that enables a) the definition of the expected payload. schema, b) the definition of the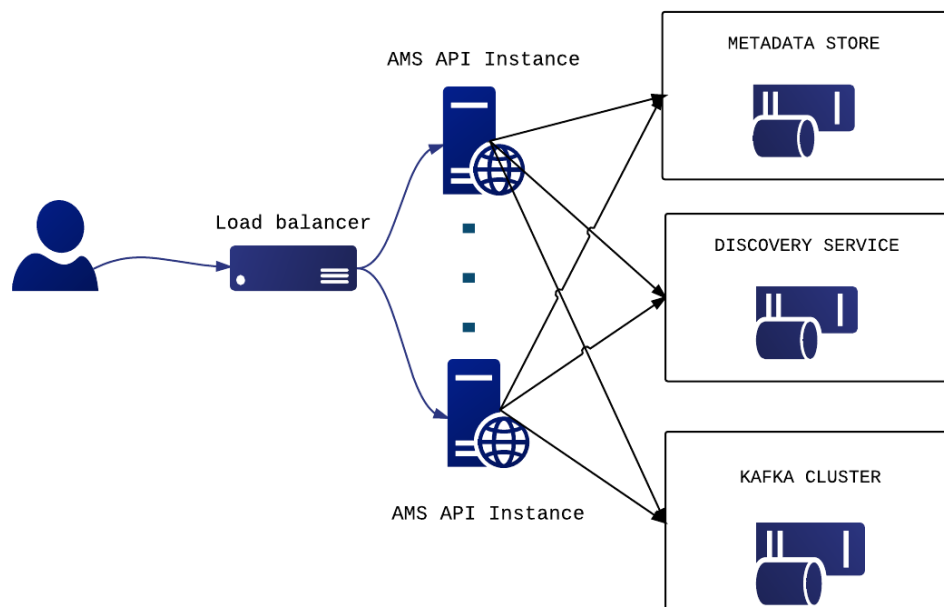 expected set of attributes and values and c) the validation for each message if the requirements are met and immediately notify the client.

- **Replicate messages on multiple topics**: Republisher script that consumes and publishes messages for specific topics (e.g., SITES).

It supports both push and pull message delivery. In push delivery, the Messaging Service initiates requests to your subscriber application to deliver messages. In pull delivery, your subscription application initiates requests to the server to retrieve messages.

Apart from all these the Messaging Service supports:

- Argo-ams-library: A simple library to interact with the ARGO Messaging Service.

- Argo-AuthN: Argo-authn is a new Authentication Service.

- AMS Metrics: Metrics about the service

6.5.1 Maintenance activities

There is a standardized maintenance window every first Wednesday (Devel instances) and Thursday (Production Instances) of each month. These maintenance windows are used for applying regular OS upgrades and stable releases. All necessary precautions (backing up the data etc) are taken care of beforehand by the monitoring team.

One major part of maintenance activities is the updates / upgrades of the software / library dependencies the AMS has. This follows a specific process where performance, features, and service stability are taken into consideration. When a reliable version of a software dependency is available, the development team deploys a new stand-alone instance to test the validity of all main features and decide on a list of changes required. When a stable version is implemented, it is deployed on the development instance for at least one month until it is deployed in the production service. Today all relevant components are using python3.

AMS follows a development process that includes mandatory tests for checking the functionality and the quality, correctness of the software. This process consists of automated unit tests and code quality checks, running via a CI tool (jenkins). Unit tests that test CRUD and domain logic functionality on all resource objects supported by the api, using mock interfaces on the datastore and broker layers (golang testify). At the same time AMS endpoints are tested as postman collections via newman. Newman is a command-line collection runner for Postman [R34]. This allows the user to effortlessly run and test a Postman Collections [R35] directly from the command-

line. It is built with extensibility in mind, and it can be easily integrated with ARGO's continuous integration server and build systems.

6.5.2   Summary of service enhancements

During the defined period, the AMS introduced **a number of new functionalities**. Our main goal was to support and facilitate the ease of use (client requests) and at the same time follow the Google Pub/Sub paradigm. The unique features mentioned below help deliver value to our service integrators (**EOSC-Core services**).

The AMS supports "**Schema Validation per topic**". It allows the user to define a schema for each topic and validate messages as they are published. It can protect topics from garbage, incomplete messages especially when a topic has multiple remote publishers to ensure data integrity on the client side.

The "**Replay messages**" feature is an offset manipulation mechanism that allows the client on demand to replay or skip messages. When creating a subscription (or editing an existing one), there is an internal option to retain acknowledged messages (by default up to 7 days, or more on request). To replay and reprocess these messages (ex. testing, error in manipulation etc), the client has the ability to go back and use the same messages just by seeking a previous timestamp. If the user needs to skip messages, he just has to seek an offset in the future.

The implementation for the "**push server**" has been deployed and used by the **AAI Federation Registry** to exchange information. The push server(s) are an optional set of worker-machines - deployed on demand - that are needed when the AMS wants to support push enabled subscriptions. The new implementation provides a gRPC interface in order to communicate with AMS api. A new security approach was also introduced to enable a secure handshake.

The AMS supports a flexible authorization based on roles. Access to topics and subscriptions based on roles defined by the admin user. One more functionality was to "**enable user management support at the project level"** (Project admins will be able to add or remove users from a project). At the same time, a **"user registration form"** approach was introduced to the AMS in general and/or to a project. User's may register with the service and admins may accept or decline the registration.

At the same time, the AMS introduced a number of **"new API calls"** with more functionality (internal api-calls) so as to simplify the processes.

Finally, during this period a number of **"new metrics"** are supported. These new improved metrics (Operational, Overall, Status and activity) are either metrics fetched from the different resources (Projects, Topics, Subscriptions, Users) or aggregated ones. These metrics are measured in a variety of ways. They may include "Daily Message Average" or "publishing-rate in topics: messages published per second between two last publish events". These different types of metrics give an idea of the usage of the service and at the same time help us understand the activity of our resources.

The **integration between different core services** using the ARGO Messaging Service (AMS) as transport layer was one of our main goals. The main services are: a) **EOSC Marketplace (beta)**: It uses the AMS Service to exchange information about the orders. b) **AAI Federation Registry (beta)**:

It uses the AMS Service to exchange information with the different deployers (ex, SimpleSamlPhp, Mitre Id, Keycloak). c) **Operations Portal:** Reads the alarms from predefined topics, stores them in a database and displays them in the operations portal. d) **Accounting:** Use of AMS as a transport layer for collecting accounting data from the Sites. The accounting information is gathered from different collectors into a central accounting repository where it is processed to generate statistical summaries that are available through the EGI Accounting Portal. e) **FedCloud**: Use of AMS as a transport layer of the cloud information system. It makes use of the ams-authN. The entry point for users, topics and subscriptions is GOCDB. f) **ARGO Availability and Reliability Monitoring Service:** It uses the AMS service to send the messages from the monitoring engine to other components.

Finally, apart from the main service a number of valuable components are also supported. These components are extensively used by the connected services. The Support, maintenance, and extension of the third components, so as to follow the evolution of the service. The components are a) **Argo-ams-library**: A simple library to interact with the ARGO Messaging Service using python programming language. b) **Argo-AuthN**: Argo-AuthN is a new Authentication Service. This service provides the ability to different services to use alternative authentication mechanisms without having to store additional user info or implement new functionalities. The AUTH service holds various information about a service's users, hosts, API urls, etc, and leverages them to provide its functionality. c) **AMS Metrics**: Metrics about the service.

6.5.3   Future plans

ARGO Messaging future plans

- Adding bookmarks /snapshots on specific subscription offsets by capturing the current state.

Support the users the services that want to start or continue using the service such as:

- Support, maintain, extend the AMS Service, AuthN Service, ams-library

- Support FedCloud Information System, EGI Information System, AppDB

6.6 Security Tools: Pakiti

Pakiti provides a monitoring mechanism to check the patching status of Linux systems. Pakiti uses the client/server model, with clients running on monitored machines and sending reports to the Pakiti server for evaluation. The report contains a list of packages installed on the client system, which is subject to analysis done by the server. The Pakiti server compares versions against other versions which are obtained from various distribution vendors. Detected vulnerabilities identified using CVE identifiers are reported as the outcome, together with affected packages that need to be updated.

A detailed description of the Pakiti service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

6.6.1   Maintenance activities

The production Pakiti server was upgraded to the latest major version of Pakiti, which finished the transition from the older implementation that was not designed to cope with current load and scope of monitoring services. At this moment, the production instance uses the current Pakiti code base, while the secondary instance uses the older version, which makes it possible to correlate results and spot possible irregularities in the reported results.

Other than that, the service was operated according to users' needs and without notable issues for the end user. The system was regularly maintained and kept up to date with security patches.

6.6.2   Summary of service enhancements

The most notable is the transition to the latest version of Pakiti service. Other enhancements include automated deployment using Ansible, revision and updates of the documentation, extended unit tests, proper support of Debian and Ubuntu distributions, and other minor changes.

6.6.3   Future plans

Final evaluation of the transition to the latest major version of Pakiti service. Ongoing maintenance and operations of the service will proceed.

6.7 Security Tools: Secant

Secant is a security cloud assessment framework that is used to check the security characteristics of virtual machines and their images. The framework instantiates the machine in a contained environment and runs a set of security probes against it. The probes combine external and internal checks and aim at typical configuration errors or vulnerabilities commonly misused by Internet attackers.

A detailed description of the Secant service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

6.7.1   Maintenance activities

During the reported period, the service was extended with connectors for OpenStack, which was needed to facilitate the management of the testing environment. The work was followed by further activities aiming at integrating the service with the new infrastructure. The main obstacle continued to be the full support of OpenStack in CloudKeeper. Activities performed during the service maintenance were focused on this integration and testing.

6.7.2   Summary of service enhancements

Activities were focused mainly on the integration of the service with services and components needed for Secant deployment and operations (namely OpenStack and the image management).

6.7.3   Future plans

Evaluation of the service, its performance with the OpenStack-based environment is planned.

6.8 Integration Activities

6.8.1 Integration of Accounting Repository and Portal with EUDAT Accounting Service

6.8.1.1 Summary of integration activities

The EUDAT Accounting information comprises bookkeeping information about disk space reserved by client projects, this information is sporadically updated manually, and date/time information refers to update time. This contrasts with the Accounting Repository, which gets the storage accounting daily, automatically, and the date corresponds with the use of the resources.

The EUDAT data was adapted to STAR records like the other existing data, but there were a lot of semantic mismatches from the situation described above, and misinterpretation of some fields, a new dummy topology had to be created, and code repurposed and/or newly created to adapt to this mapping.

6.8.1.2 Identified integration gaps

Apart from having to introduce sub-terabyte units due to the granularity of the EUDAT data, various attempts to adapt the multi-variate, dynamic, date centred approach of the portal were not successful due to the sparsity of EUDAT data. In the end, a static report which excluded using dates was created to better adapt the semantics of EUDAT data.

6.8.1.3 Future plans

Once the evaluation of the static-reports solution has been completed the integration activity will be reviewed and further improvements will be considered.

6.8.2 Integration of ARGO with EOSC Portal

6.8.2.1 Summary of integration activities

ARGO is integrated with the EOSC portal and uses the topology it provides to monitor the service listed and provide a view of their status at https://argo.eosc-portal.eu as shown in Figure 6-5. The main goal is to check the validity of the services onboarded in the portal and identify "dead" services. Through the Web UIl operators, customers and providers can now check the health status of a service and get information about its stability.
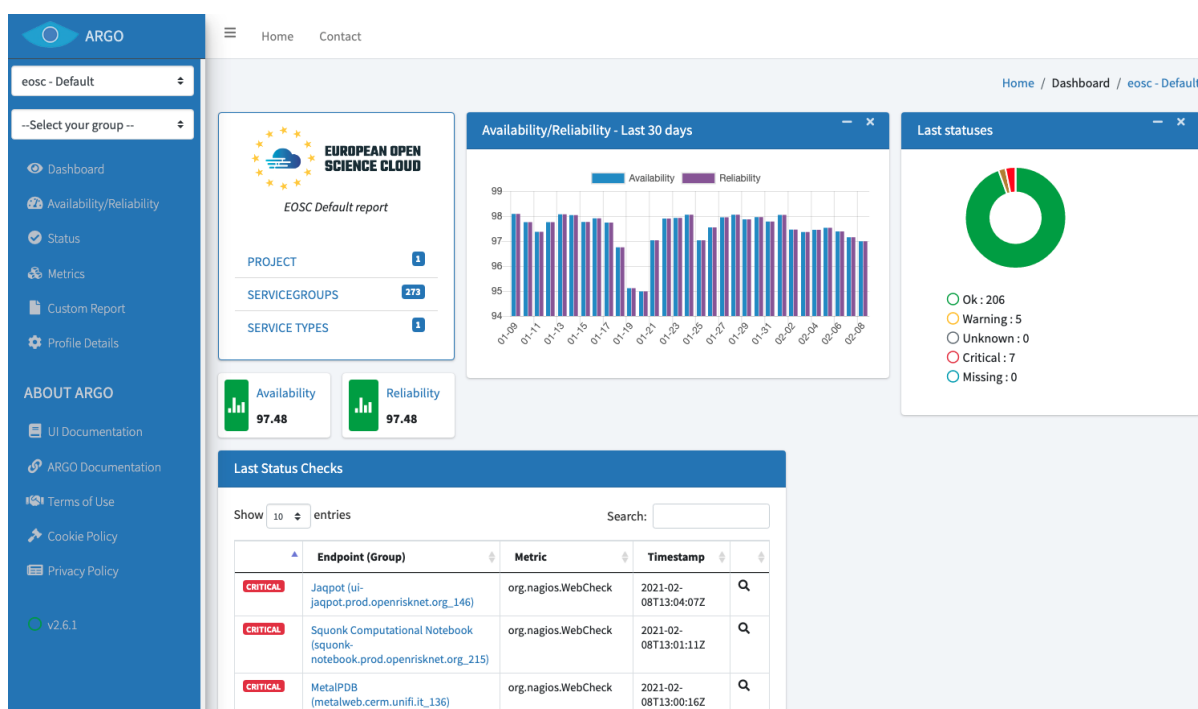
Fig.6-5 - ARGO monitoring dashboard for onboarded services.

6.9 Summary and Outlook

During the EOSC-Hub project ARGO Monitoring managed to achieve its goal of the One Stop Shop that simplifies and automates the operation and configuration of ARGO components.

ARGO web-api was extended with a number of new functionalities enabling it to be the central source of truth for all the components of ARGO. This led to the minimization of the manual intervention and therefore reduced the probability of configuration errors. New jobs and changes in the computational model and pipeline were added to the Analytics Engine. These changes in combination with the new argo-web-api interface and the automatic configuration led to the real-time updates of the source data.   Poem now represents a central dashboard (UI) for managing most of the resources used in the ARGO monitoring engine. It follows the principal towards One-Stop-Shop functionality and using ARGO web-api as centralized. It is the central management interface for a tenant to prepare all the resources for the monitoring. Web-UI also uses the web-api to configure itself, fetch the topology and the results which reduces dependency to external information providers. The ARGO monitoring service evolved drastically so it is now able to spawn a new tenant almost automatically. The preparatory work of setting the argo-web-api as the central point of truth and the integration with external and internal sources of information led ARGO Monitoring to be part of EOSC-Core where it will be required to handle a plethora of new metrics, profiles and topologies coming from different sources and evolve to act as an aggregator of Monitoring data coming from external sources and handle the use cases as they are described in EOSC-HUB Technical Specification [R36].

During the same period, Argo Messaging Service (AMS) introduced a number of new functionalities. These new functionalities, such as schema support, the replay messages help deliver value to our service integrators (EOSC-Core services). The push delivery of asynchronous events is used by the

AAI federation registry to exchange information with the Deployers. AMS via AuthN can now accept different methods of authentication such as x509 Certificates and with this functionality Sites, by using the ams-library, can now transparently authenticate to send their metrics to the Accounting service. All these features lead AMS to play the role of the transport layer for the secure exchange of information between the services (Accounting, Monitoring, AAI) additions will allow AMS to play an important role in EOSC-Core as the transport layer between EOSC-Core components and EOSC-Exchange providers that are willing to push Monitoring and Accounting data to EOSC-Core.

The Accounting Service (Repository and Portal) successfully addressed the following tasks, migrating from the deprecated Message Broker Network to AMS, support for EUDAT Storage Accounting records and to be based in Python 3. However In order to provide an accounting system capable to cater for all EOSC accounting needs (including different type of resources (Service, Portals, data(sets) etc) that is capable to provide via an API the necessary data for Dashboards for (providers, end-users, consumers, PMB, EC etc) the accounting service will need to be redesigned so that it becomes more flexible and transparent on the way it handles the accounting and usage records from a plethora of different services besides the traditional storage and computing types.

7   Helpdesk Services and Tools

7.1 Overview

The EOSC-hub Helpdesk consists of a ticketing system based on GGUS with interfaces to the support infrastructures like EUDAT and EGI. The activity consists in providing the 1st level support and dispatch incoming requests to the adequate service expert teams and other downstream support teams.

The design and implementation of the EOSC-hub helpdesk system as well as its integration of the EGI and EUDAT mature helpdesk systems according to the project roadmap has been described in detail in previous deliverables D5.3 [R37] and D5.5 [R38]. In the last period of the project the focus has been given to improvement of the EOSC-hub helpdesk, its usability and experience of users by restructuring support units to attend to users' requests. In addition, the interoperability guidelines [R39] have been defined and integration scenarios of EOSC-hub helpdesk with other services including newly onboarded services have been specified.

7.2 GGUS

A detailed description of the GGUS messaging service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

In addition, information on change, release and deployment are available in the EGI wiki at [R40] [R41].

7.2.1   Maintenance activities

GGUS releases take place in a bi-monthly release schedule. Releases are usually done on the last Wednesday of the release month. They are recorded and announced via GOCDB maintenance feature. All release dates are listed in EGI wiki. No major changes during the covered period have been performed. Maintenance activities of GGUS are documented in the GGUS release notes available at [R42].

7.2.2   Summary of service enhancements

The following enhancements have been implemented during the third period of the project:

- Improvements of the report generator.
- Reworked ticket search.
- Introduced new support units.
- Retired obsolete support units.
- Implemented VO-specific fields and issue types for CMS and ATLAS.
- Improved VOMS synchronization.

7.2.3   Future plans

- Improve GUI

7.3 EUDAT-RT

7.3.1 Maintenance activities

The EUDAT RT service has been regularly maintained, and tests for the correct communication between the EOSC-hub ticketing system and the EUDAT one has been carried out successfully.

EUDAT-RT upgrades and improvements have been implemented. The EUDAT ticketing system has demonstrated to be stable and no changes to the connection with the EOSC-hub ticketing system was necessary.

7.3.2 Future plans

The integration between the ticketing system is intended to be maintained and reinforced in the future. The evolution of the EOSC ticketing system will be followed by EUDAT in the context of the DICE project. The continuity of the service will be maintained as much as possible, to minimize the impact for the users.

7.4 xGUS

7.4.1 Maintenance activities

Maintenance activities are coupled with GGUS maintenance activities. About xGUS releases, they take place in a bi-monthly release schedule. Releases are usually done on the last Wednesday of the release month. They are recorded and announced via GOCDB.

7.4.2 Summary of service enhancements

Nothing to Report.

7.4.3 Future plans

Nothing to Report.

7.5 Current Support Units Structure
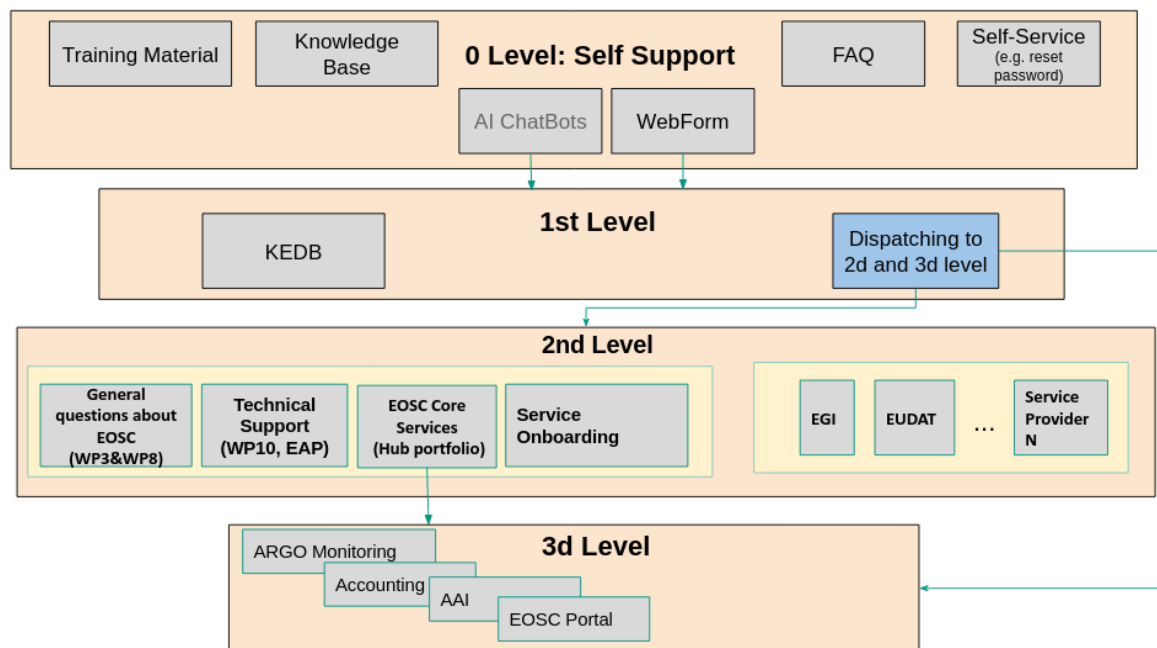
Fig.7-1 - EOSC-hub Helpdesk Support Levels and Units.

The Support Units (SU) structure has been significantly modified and enhanced during the last project period according to the requirements established in Incident and Service Request Management Process.   We distinguish between Level 0 (self-support through documentation, guidelines, etc.), Level 1 (for tickets classification and redistribution), Level 2 (generic technical questions) and Level 3 (experts) support, with the respective functions and scopes. The number and scope of the SUs have been organized according to the expertise required to cover all the EOSC-hub services. The internal coordination of each SU is a task assigned to the SU Responsible.

7.6 Helpdesk Offering and Integration Options
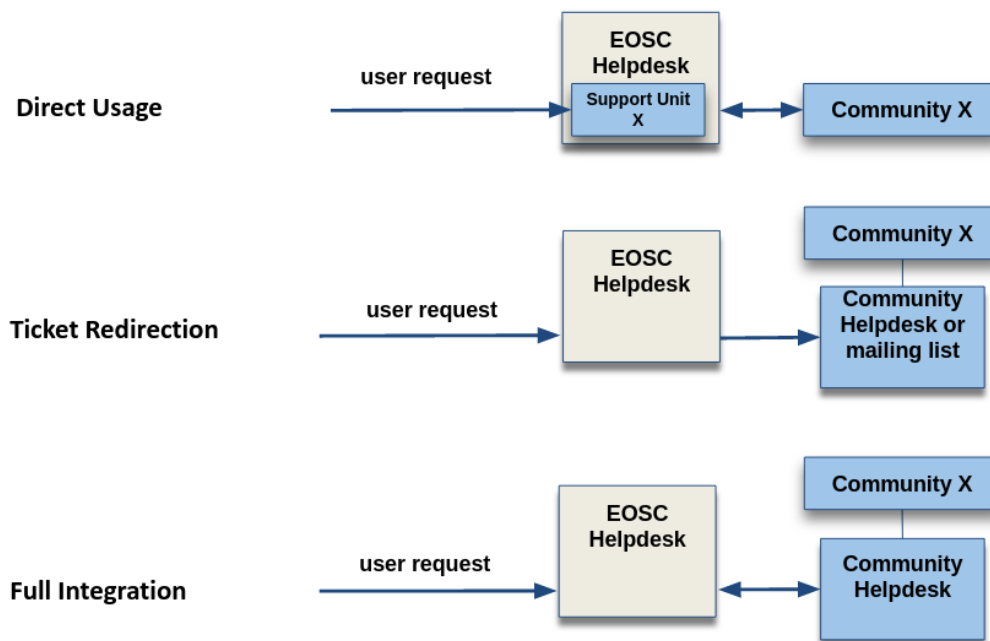
Fig.7-2 - Helpdesk Integration Options

The EOSC Helpdesk, being a part of EOSC-Core works as a unified ticketing system by managing the requests related to different services provided by different communities. After initial integration of the EOSC-hub Helpdesk with EGI and EUDAT Helpdesk systems, during the last reporting period we have identified the three major integration options or scenarios the EOSC Helpdesk should offer to any community or service provider as shown in Figure XX:

- **Direct Usage**: Use directly the EOSC helpdesk as the ticketing system. This scenario is to be implemented when a community does not have its own helpdesk and would like to use EOSC Helpdesk to manage the user requests, addressed to the community services. In this case the community obtains a support unit or a set of support units for its own disposal.

- **Ticket Redirection**: Use the EOSC helpdesk only as a contact point to redirect the entry request for the specific service to a mailing list or 2nd level ticketing system. In this case, the EOSC Helpdesk central service would simply redirect by e-mail or via API the incoming tickets to the external system.

- **Full Integration**: Bi-directional Integration of the community ticketing system with the EOSC helpdesk, which means the full synchronisation of the content between two systems. In this scenario the community tickets can be managed in any of the two helpdesks. For example, the ticket which initially has been opened in EOSC Helpdesk is propagated to community helpdesk and all changes done for this ticket in community helpdesk are visible also in EOSC Helpdesk.

7.7 Summary and Outlook

All major goals for the task have been achieved during the run-time of the project. The EOSC-hub Helpdesk has been established, the full integration with EGI and EUDAT helpdesk systems and with EOSC Portal have been accomplished. The support unit structure has been implemented and integration models for external communities and guidelines have been delivered. Documentation for service providers and for users have been completed and delivered. New procedures for handling incidents of services have been designed, discussed, and approved in collaboration with WP4. The EOSC-hub helpdesk service is in production and is regularly maintained.

The main effort in the next projects will be focused on integration of the helpdesk with supporting systems of onboarded communities and delivery of the Helpdesk-as-a-Service for interested groups in EOSC. A significant improvement of Self-Support Level is planned meaning the implementation of new user interfaces. In addition, the integration of the helpdesk with configuration management system is planned to facilitate the change management and problem management processes.

8 Application store, Software Repositories and other Collaboration Tools

8.1 Overview

This section details the maintenance and integration activities for the Application Database, Software Repository, and GitLab instance, along with enhancements made and plans for the future. The AppDB portal, VMOps dashboard, and Information System fully migrated to the GLUE2.1 schema for cloud site information, and OIDC authentication, obsoleting previous technologies used in place. Moreover, two new dashboards were released, the security dashboard and the endorsements dashboard, to complement the VMOps dashboard. The Software Repository components were refactored to support dockerized deployment and support for OS/UMD versions were improved.

8.2 Application Database

A detailed description of the AppDB service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

8.2.1 Maintenance activities

During the final year of the project, the AppDB portal received one major and one minor release, and the VMOps portal, Information System, and Continuous Delivery system received two minor/revision releases, each. In addition to the service enhancements addressed in the following sections, these releases addressed a variety of fixes pertaining to bug fixes and performance improvements.

8.2.2 Summary of service enhancements

The main focus of enhancements across all components of the service was the migration to the GLUE2.1 schema for cloud site information, along with the migration to OIDC authentication for VM deployment and management actions. Preliminary work on these activities, which had started since the previous period, was finalised and the changes were released into production with v7.0.0 of the AppDB portal, v1.4.0 of the VMOps dashboard, and v1.4.0 of the InfoSys, thus dropping support for the BDII service in favour of the EGI Cloud Information Provider, which provides data to the AppDB InfoSys over the Argo Messaging Service. Support for x509 authentication was also removed in favour of OIDC, wherever applicable.

In addition to the migration-related actions above, the Applications Database received several other improvements. Its InfoSys RESTful API, which was introduced during the previous period, was equipped with a new Swagger UI, which serves both to document the API itself, as well as a testing environment. The AppDB continuous delivery system received a security enhancement, where VM checksums may be checked against checksums provided by the VM author, and the service's integration with the EGI Operations Portal concerning VO information, was adapted to use its newly

introduced RESTful API, dropping the old XML-based VO ID card system. Finally, two new dashboards, complementing the existing VMOps dashboard, were released, namely the Security and the Endorsements dashboard, with the purpose of providing better management of VO-wide image lists by managers and administrators.

The Security dashboard (see Figure 8-1) is aimed towards security experts, who may use it in order to review information for specific VM image versions, such as the FedCloud sites/endpoints that provide the VM image in question, under which VOs it is provided, whether its containing VO-wide image list that the sites are subscribed to is up-to-date, etc., along with basic information such as the image's size, location, checksum, release date, owner, etc., Security experts/officers may then tag the image for specific security issues and submit and monitor a GGUS ticket from within the dashboard as shown in Figure 8-2.
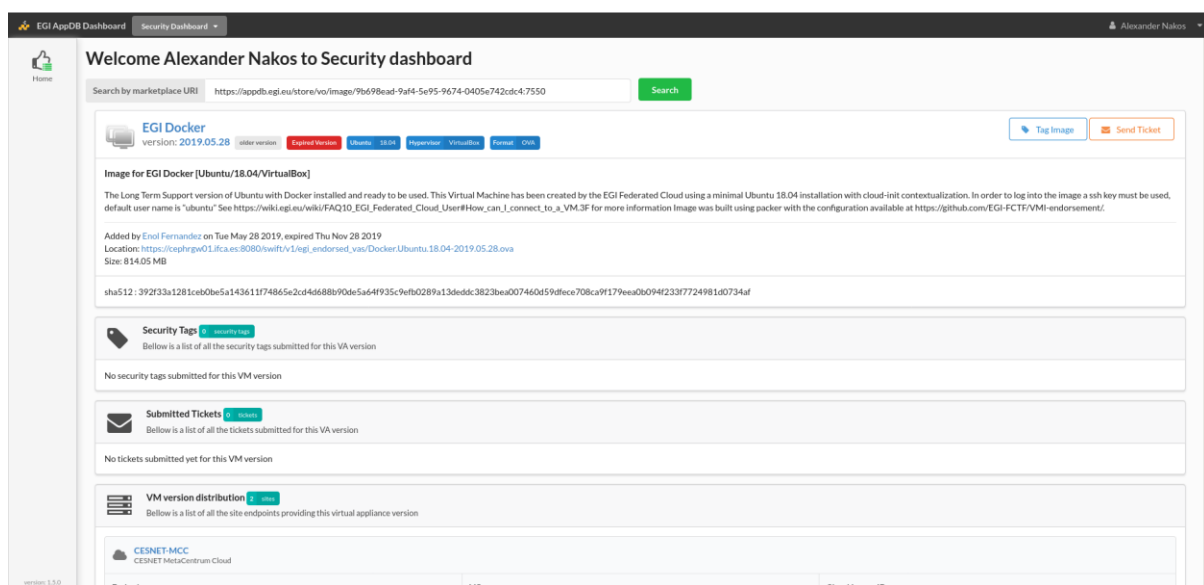
Fig. 8-1 Security dashboard main view example.

Fig. 8-2 Example of a GGUS ticket distribution dialog in the Security dashboard

The Endorsement dashboard (see Figure 8-3) may be accessed by all registered users, in order to provide a positive overall assessment for specific VM images, while security experts/offices may provide positive assessments from a security perspective. This information is communicated to VO managers when editing VO-wide image lists, helping them make better informed decisions about VM image inclusion.
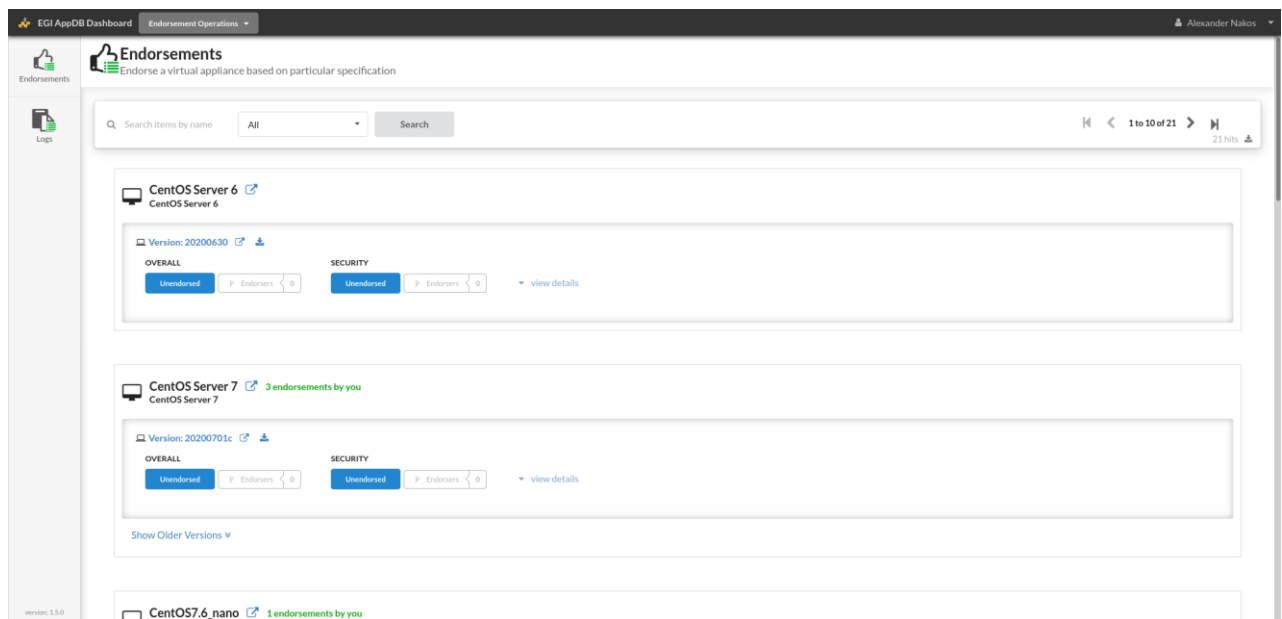
Fig. 8-3 Endorsement dashboard main view

8.2.3   Future plans

Other than providing support and maintenance for the AppDB and its components as a whole, future plans include making potential improvements on the latest components, based on feedback from the community. This would include the security and endorsements dashboards, which were released in Q4 of 2020, and the container image registry, developed during the project extension, which aims at providing support for running applications via container engines such as Docker within the FedCloud infrastructure.

8.3 GitLab

A detailed description of the GitLab service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

Projects: 129	Users: 328 ⊘	Groups: 26
New project	New user / Users statistics	New group

Statistics		**Features**		**Components**	up-to-date
Forks	16	Sign up	⏻	GitLab	13.8.4 (9fb9cbf50c3)
Issues	451	LDAP ❓	⏻	GitLab Shell	13.15.1
Merge Requests	149	Gravatar	✓	GitLab Workhorse	v8.59.0
Notes	1,526	OmniAuth ❓	✓	GitLab API	v4
Snippets	14	Reply by email ❓	⏻	Ruby	2.7.2p137
SSH Keys	72	Container Registry ❓	⏻	Rails	6.0.3.4
Milestones	5	Gitlab Pages ❓	⏻	PostgreSQL	12.5
Active Users	310	Shared Runners	✓	Redis	5.0.9
				Gitaly Servers	

Fig. 8-4. GitLab Dashboard with information on usage metrics, users, and software configuration.

8.3.1   Maintenance activities

During the current reporting period we performed regular maintenance activities including daily backups and updates according to the GitLab release schedule. No interruptions have been observed.

8.3.2   Summary of service enhancements

The usage of the GitLab during the project has significantly increased. We have started with less than 50 users at the beginning of the project. Currently as shown in Figure 8-4 the user base of GitLab contains more than 300 active users with 129 projects. The GitLab instance provides a wide range of functions to manage the software development and perform the DevOps operations.

The number of runners is increased according to the needs of the users. Currently five active runners are implemented. The storage space has been significantly extended.

8.3.3   Future plans

We are expecting a significant growth of user base and general increase of GitLab usage not only as a repository of code snippets, but also as a tool for automation of the DevOps operations and continuous integration. The future activities include the provisioning of the Docker image repository, extension of runners according to the user requests and extension of storage space.

8.4 EGI Software Repository

A detailed description of the EGI Software Repository service is given in D5.1 [R6]. The release notes for the reporting period are provided in Appendix A.

8.4.1 Maintenance activities

In addition to the usual support activities and bug fixes, maintenance activities during the period in question include reconfiguring the service in order to properly support the upgraded EGI RT service, which was migrated to a new server, and improving security by supporting TLS1.2, defaulting to https, and ensuring no mixed content existing when viewing web pages over https.

8.4.2 Summary of service enhancements

Service enhancements realised during the period include the restructuring of the repository frontend and backend services, which have been converted to use a combination of docker container appliances. This allows for better infrastructure management and scalability, as well as easier updates of the individual components. Moreover, the first release definition for UMD 5.0 on CentOS 7 was released, and the Community repository gained support for CentOS 8 and Ubuntu Bionic Beaver LTS.

8.4.3 Future plans

No further changes are planned for the immediate future.

8.5 Integration activities

8.5.1 Integration with the EGI Operations Portal

8.5.1.1 Summary of integration activities

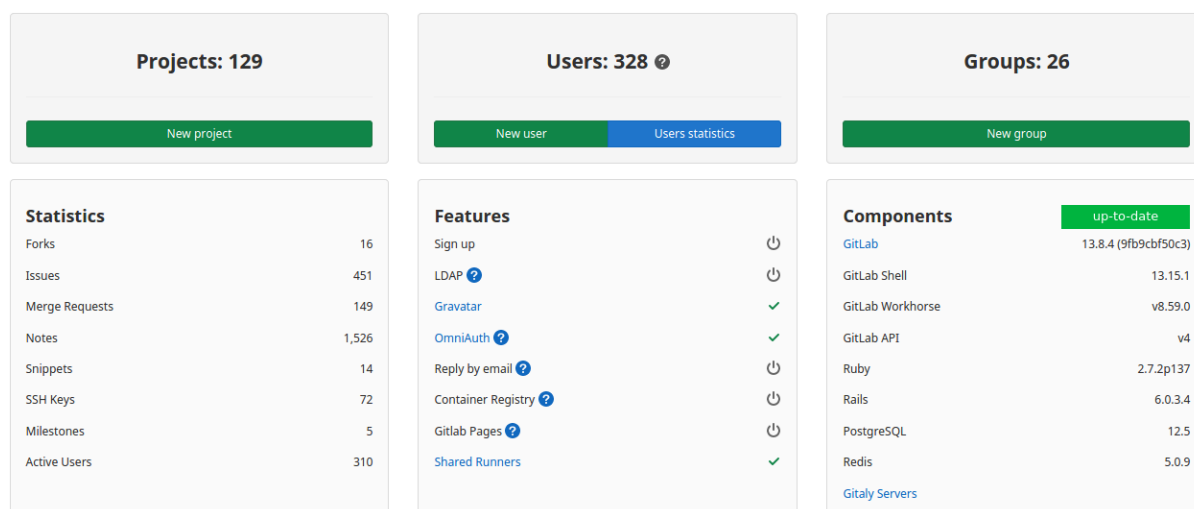The AppDB portal and related components mainly rely upon the EGI Operations portal for VO information. Integration with the operations portal was based, up to recently, on the latter's provisioning of the so-called VO ID cards, which are XML dumps of VO information. With the advent of the new RESTful API of the operations portal, which obsoletes the ID cards, AppDB components were adapted to using the new API in order to sync VO information.

8.5.1.2 Identified integration gaps

There is a 1-1 correspondence of the information previously provided, with the information provided by the new API, by applying the proper transformations. Information previously provided which is missing in the new API is either obsolete or was re-added after communication with the operation's portal provider.

8.5.1.3 Future plans

There are no further plans for the immediate future, as integration with the new API is complete.

8.5.2 Integration with EGI Check-In

8.5.2.1 Summary of integration activities

All of the AppDB components rely upon the EGI Check-In service for user authentication and authorization. After the migration from x509 certificates to OIDC as far as cloud sites are concerned, AppDB uses OpenID connect access tokens to perform user actions to the EGI FedCloud infrastructure, prompting users for their consent on-site. In order to monitor the outcome of such

actions, AppDB makes use of a dedicated service account, which can access read-only information of the cloud infrastructure via its own short-lived OIDC access tokens.

8.5.2.2 Identified integration gaps

Deprovisioning cannot be triggered without user interaction, i.e., having the user log into the service, in order to refresh entitlements. A bulk deprovisioning process, covering services reliant upon EGI Check-In has been under active discussion.

8.5.2.3 Future plans

Once a bulk deprovisioning process has been agreed upon, AppDB components will adapt in order to implement it.

8.5.3 Integration with AMS

8.5.3.1 Summary of integration activities

Up until the migration to GLUE2.1, AppDB components relied upon BDII servers in order to synchronize cloud information. After migrating to GLUE2.1, all of the related components have switched to using the EGI Cloud Info Provider over the ARGO Messaging Service (AMS), instead.

8.5.3.2 Identified integration gaps

The FedCloud infrastructure is actively working upon a centralized version of the Cloud Info Provider, which should eliminate potential problems related to discrepancies among cloud sites. This implies centralized monitoring amongst other things, which is currently under testing by using the AMS against the AppDB information System API, with a positive outlook, so far.

8.5.3.3 Future plans

Future plans include making any necessary changes in the AppDB Information System API, in order to ensure the proper functioning of the actions mentioned above.

8.5.4 Integration with GGUS

8.5.4.1 Summary of integration activities

The AppDB VMOps dashboard has long been integrated with the GGUS portal in order to open user tickets related to issues that may arise while instantiating / operating VMs on FedCloud sites. With the release of the Security dashboard, integration has been reworked in order to support creating tickets related to security issues that may be identified by security experts/officers, which apply to specific VM images.

8.5.4.2 Identified integration gaps

No integration gaps have been identified.

8.5.4.3 Future plans

No further changes are planned for the immediate future.

8.6 Summary and Outlook

The focus of the activities set out for this task have been on improving the quality of the services provided to the community. The Applications Database has delivered its initial goals of an enhanced service, without any deviations from the roadmap, except for the push-back of certain milestones due to minor delays created by the COVID-19 situation. In particular, some notable major achievements are the deployment of two new dashboards which enhance the quality of offered VM services, the migration to the GLUE2.1 cloud information schema which provides better support for VM operations on cloud sites, and the introduction of a new information system which provides a consolidated view of the FedCloud infrastructure, both to AppDB components and to external services alike, though its RESTful API.

As for the Software Repository, certain plans that were to be investigated with respect to their potential impact and benefit, during the duration of the project, such as the migration to a different system for the provisioning process or the potential move from EGI SSO to EGI Check-In for authentication, were not deemed necessary as, in the former case, the upgrade of the existing RT system superseded any such plans and, in the latter case, the fact that the user-base for the administrative part of the service is strictly limited to a few select people overseeing it, makes SSO management a better fit.

The usage of the GitLab during the project has significantly increased with rapid growth of the user base and number of software development projects. The capacity of the GitLab instance has been significantly extended to meet the requirements and user requests.

9   Summary

The deliverable provides a detailed overview of the progress and achievements during the last reporting period of the EOSC-hub project. It outlines the final status of the integration work for all federation and collaboration services included in the Work Package 5 which has been started based on the objectives of the project and following the initial roadmap developed in the initial project phase according to the assessment of the initial requirements of the project stakeholders.

Following the detailed assessment of the progress provided in the summary section of each task we conclude that all major objectives for Work Package 5 were accomplished. A significant effort was dedicated to the preparation of the documentation for different target groups of the users with different technical levels and roles.

The work carried out in WP5 was two-folded: to provide stable operation of the core federation services to support the EOSC-hub infrastructure, and to integrate and enhance multiple federation services, improve their interoperability in order to address the use cases and needs of the multiple EOSC research communities, other EOSC-hub technical work packages and finally shape the initial EOSC-Core with a set of capabilities like discovery, access, ordering, monitoring, accounting, helpdesk support etc.

All services in WP5 demonstrated high availability and no significant breaks or unscheduled downtimes of the service, during the whole project period. The single exception to this was a short outage of the Marketplace which was caused by severe hardware problems at the infrastructure level at site. This outage was properly handled and resolved within a few hours.

With respect to the integration and service enhancement activities we faced much more challenges considering the natural complexity of this task in the multi-tenant EOSC federated environment with rapidly changing requirements, impacts of other EOSC projects and governance decisions. We consider mentioning some of these challenges in this final summary.

A remarkable progress has been achieved within the AAI task. Even so, in the AAI integration activities we faced the significant challenge in an establishment of trust between integrated services. Despite the adoption of common technical and policy guidelines, the establishment of M:N relationships can pose scalability issues, particularly with the growing number of Community AAI and Infrastructure Proxy services that need to be interconnected for enabling access to resources across infrastructures within the wider EOSC environment. To mitigate these scalability concerns the AAI task force has proposed the establishment of EOSC AAI Federation. Other gaps in the AAI integration have been clearly identified and resolution plans and workarounds are proposed and will be further evolved in the follow-up projects.

The introduction of the EOSC Portal and a significant amount of initially unplanned work was another challenge we faced during the project. This change of the strategy at governance level has been successfully assessed at the package technical level by adoption of the roadmap and shifting the focus to the new required integration work within the scope of the EOSC Portal.

A successful and dynamic development of the Marketplace, integration of related services and establishment of Order Management System is another significant achievement of the work

package. However, as was pointed out in the previous deliverable D5.5, the main challenge here is the development of the federated Order Management System which considers the involvement of multiple scientific communities and service providers and integration of their own Order Management Systems. We faced the fact that other communities do not have any mature order management systems to integrate with. As a response to this challenge, we developed the White Label Marketplace which could be adopted by any community and be integrated with less effort in the EOSC Order Management System to expose the services at EOSC level and focused the work on enhancement of the ARGO Messaging Service as a uniform transport layer for exchange of order information.

Another important component which would be required to fully automate the order management in federated scenario is the advanced interoperable Configuration Management System which should include not only unified service and resource catalogues, but also provide a uniform view on the underlying infrastructure and facilitate the order propagation to the involved service providers.

Although Accounting Service (Repository and Portal) has been significantly enhanced as summarised in the Chapter 6, a significant effort is still required to provide an accounting system capable of addressing all requirements of EOSC including the aggregation of accounting and usage records of multiple types of resources like services, storage systems, computing resources, software, container, apps. The system should be capable of providing this information via API to the dedicated dashboards for different target users (providers, end-users, stakeholders etc.). This challenging task requires redesign of the accounting and related services, development of system architecture and should be prioritised in the follow-up projects.

Despite the challenges and gaps mentioned above, which have been identified during the course of the project, we consider that WP5 provides a solid foundation in terms of set of service candidates for the EOSC-Core, gained experience, developed interoperable interfaces and knowledge required for further implementation of EOSC.

10 References

No	Description/Link
R1	https://github.com/rciam/comanage-registry-plugin-RciamEnroller
R2	https://github.com/rciam/comanage-registry-plugin-RciamStatsViewer
R3	https://github.com/rciam/comanage-registry-plugin-VomsProvisioner
R4	https://aai.egi.eu/sshkeys
R5	https://aai.egi.eu/registry/
R6	https://documents.egi.eu/public/ShowDocument?docid=3344
R7	https://zenodo.org/record/1308682#.Xti-U_lS9hF
R8	https://documents.egi.eu/public/ShowDocument?docid=3503
R9	https://tools.ietf.org/html/rfc8693
R10	https://aarc-project.eu/wp-content/uploads/2017/11/AARC-JRA1.4A-201710.pdf
R11	https://refeds.org/category/research-and-scholarship
R12	https://aarc-community.org/guidelines/aarc-g027/
R13	https://aarc-community.org/guidelines/aarc-g025/
R14	https://ssllabs.com/ssltest
R15	https://git.scc.kit.edu/zj6298/watts-secure/blob/master/CredentialStore.md
R16	https://aarc-community.org/guidelines/aarc-g045/
R17	https://op.europa.eu/s/oMBC
R18	https://info.orcid.org/faq/how-does-3-legged-oauth-work/
R19	https://www.keycloak.org/
R20	https://github.com/eosc-kc
R21	https://github.com/keycloak/keycloak
R22	https://github.com/keycloak/keycloak-documentation
R23	https://openid.net/specs/openid-connect-federation-1_0.html
R24	https://refeds.org/wp-content/uploads/2016/01/Sirtfi-1.0.pdf
R25	https://wiki.geant.org/download/attachments/123766285/WISE-SCI-Baseline-AUP-V1.0.1-draft.pdf
R26	https://aarc-community.org/about/aegis/

R27	https://github.com/cyfronet-fid/egi-marketplace/blob/master/README.md
R28	https://marketplace-11.docker-fid.grid.cyf-kr.edu.pl/api_docs/swagger/index.html
R29	https://www.eosc-portal.eu/sites/default/files/EOSC-Profiles-v3.00.pdf
R30	https://www.google.com/url?q=https://wiki.eosc-hub.eu/download/attachments/34637786/EOSC%2520Portal%2520Collaboration%2520Agreement%2520v0.4.pdf?version%3D1%26modificationDate%3D1558632423697%26api%3Dv2&sa=D&source=editors&ust=1615483520922000&usg=AOvVaw1n0IJQc5Dltwmlf0_s-cvc
R31	https://wiki.egi.eu/wiki/GOCDB/Documentation_Index https://github.com/GOCDB/gocdb/blob/dev/README.md
R32	https://zenodo.org/record/4040865
R33	https://www.selenium.dev
R34	https://getpostman.com
R35	https://www.getpostman.com/docs/collections
R36	https://confluence.egi.eu/display/EOSCDOC/Monitoring
R37	https://documents.egi.eu/document/3503
R38	https://documents.egi.eu/document/3645
R39	https://wiki.eosc-hub.eu/display/EOSCDOC/Helpdesk
R40	https://wiki.egi.eu/wiki/GGUS
R41	https://wiki.egi.eu/wiki/GGUS:Main_Page
R42	https://ggus.eu/index.php?mode=release_notes
R43	https://repository.eosc-portal.eu/index.php/s/62ZHQNezK3oyKca#pdfviewer

Appendix I.   Service INFORMATION

A.1 B2ACCESS

Service/Tool name	B2ACCESS
Service/Tool url	https://b2access.eudat.eu
Service/Tool information page	https://www.eudat.eu/services/b2access
Description	The B2ACCESS service is an Identity and Access Management (IAM) system which arbitrates authenticated access to registered services. The role of the B2ACCESS service is to allow these services to perform authentication, to take authorisation decisions, and to perform any other processing of user information (e.g., harmonisation or translation), when end users access these services.
Value proposition	B2ACCESS acts as a proxy IdP, following the AARC Blueprint Architecture, which allows users to sign in with their preferred primary identities. These identities can be provided by external identity providers, e.g., Shibboleth IdPs of the users' home organisations or OpenID Connect providers such as the Google IdP, or they can be provided by the B2ACCESS service itself, if the users registered genuinely on this service.

B2ACCESS supports multiple protocols for authentication, such as SAML and OpenID Connect/OAuth2, for external identity and service providers. It translates the attributes from one protocol to another. This, for instance, allows users of a service, connected via the OAuth2 protocol, to sign in with their home organisation identity provider, connected via SAML.

Besides identity management, B2ACCESS also provides group and attribute management. Accounts can be extended by attributes, which are needed by connected services, but not provided by the external identity provider, e.g., assurance information. Hierarchical groups allow for flexible group management, e.g., separations by resources or thematic diversity. Both features offer the possibility for fine grained authorisation decisions.
The attribute, identity and group management can be done by the web interface or the REST API. |
Customer of the service/tool	Resource Provider; Research Communities
User of the service/tool	Community/VO managers, researchers, Operations Managers for research infrastructures/collaborations
User Documentation	https://eudat.eu/services/userdoc/b2access-management

Technical Documentation	Service integration: https://eudat.eu/services/userdoc/b2access-service-integration Unity manual: http://www.unity-idm.eu/documentation/unity-2.8.2/manual.html
Product team	JUELICH
License	http://www.unity-idm.eu/opensource/[1]
Source code	Unity: https://github.com/unity-idm/unity EUDAT extension: https://github.com/EUDAT-B2ACCESS/b2access-unitytheme

Release notes	2.5.20:
	  − Update to new major release
	  − Rework of the UI on user facing parts to an easier usage
	17.11.20:
	  − Several bug fixes
	  − Rework of the UI on the administration facing parts; harmonisation with the user facing parts

Testing	Each new release of the underlying software must pass a set of tests. These tests are conducted in two steps. First, the basic functionality of the software itself is tested. There is no integration with external authentication services and only demonstration services are connected as service providers. In the second step, the software is tested in an environment closely resembling the production system. In addition to the test of the specific operating system level, the integration of external authentication services like eduGAIN or Google is tested. All tests are done by operators, who know the service, and users who do not know the setup. If some tests fail, the problem is investigated. If there is no solution to pass the test, e.g., because of a bug inside the software, the version is skipped.
TRL	9

---

[1] The Unity-IDM license complies with the Open Source Definition since redistribution and use in source and binary forms, with or without modification, are permitted provided that the conditions listed in http://www.unity-idm.eu/opensource/ are met.

A.2 Check-in

Service/Tool name	EGI Check-in
Service/Tool url	https://aai.egi.eu/
Service/Tool information page	https://wiki.egi.eu/wiki/AAI
Description	The EGI Check-in service is an Identity and Access Management solution that makes it easy to secure access to services and resources.
Value proposition	Through Check-in, users are able to authenticate with the credentials provided by the IdP of their Home Organisation (e.g., via eduGAIN), as well as using social identity providers, or other selected external identity providers. Check-in provides an intuitive interface for communities to manage their users and their respective groups, roles, and access rights. For communities operating their own group management system, Check-in has a comprehensive list of connectors that allows communities to integrate their systems as externally managed Attribute Authorities.
Customer of the service/tool	Research Infrastructures, Research Communities, Resource Providers
User of the service/tool	Community/VO managers, researchers, Operations Managers for research infrastructures/collaborations
User Documentation	https://docs.egi.eu/users/check-in/
Technical Documentation	https://docs.egi.eu/providers/check-in/
Product team	GRNET
License	Apache License Version 2.0
Source code	https://github.com/rciam https://github.com/EGI-Foundation/simplesamlphp-module-themeegi
Release notes	See Section A2.1
Testing	Functional and user interface testing is performed before every change. Higher risk changes are reviewed by the EGI Change Advisory Board before being released in production.
TRL	9

A2.1 Release notes

v21.03.1 - 2021-03-16

Added

- Support for expressing identity assurance information based on the REFEDS Assurance Framework

v21.02.1 - 2021-02-21

Changed

- Create user's full name information based on the given name and family name registered for the user. Full name information is released through the `displayName` in SAML or the `name` claim in OpenID Connect.

v21.01.1 - 2021-01-19

Added

- Support for releasing the ssh public key(s) associated to the user's profile in the Membership Registry (COmanage Registry). When available, SSH keys are released through the `urn:oid:1.3.6.1.4.1.24552.500.1.1.1.13` (`sshPublicKey`) attribute in SAML or the `ssh_public_key` claim in OpenID Connect.

Changed

- Adoption of new Check-in Acceptable Use Policy (AUP) based on the WISE baseline AUP model [R25]. The new AUP is available at https://aai.egi.eu/aup/check-in.html. Users will be required to accept the new AUP upon login.

v20.12.1 - 2020-12-23

Changed

- Increase email verification link validity from 24 to 72 hours
- Change sender information (from field) in Check-in email notifications
- Update message template for notifying users that they need to verify their email address

Fixed

- Bug that prevented service identities using certificate credentials to register their Distinguished Name (DN) in the Check-in user profile

v20.11.4 - 2020-11-27

Added

- Support for renewing acceptance of updated AUPs from the user profile page
- "Actions" dropdown in the linked identities panel that replaces the existing "Link new identity" button. The "Actions" dropdown includes the following options:
    - Link new identity

　　　○　Link RCauth certificate

v20.11.3 - 2020-11-20

Added

- Support for releasing the `cert_entitlement` claim to provide information about the user's certificate subject(s) and the associated VO(s).

v20.11.2 - 2020-11-11

Added

- Support for expressing the email verification status in the SAML attribute assertions and the OIDC claims

v20.11.1 - 2020-11-06

Fixed

- Bug that prevented users from signing up with Check-in when their attribute assertion contained multiple certificate subject Distinguished Names (DNs)

v20.10.4 - 2020-10-23

Changed

- Simplify explicit identity linking flow:
  - User selects "Link New Identity" and is prompted to start the process
  - User is presented with the list of Identity Providers in order to choose the one to be linked
  - User authenticates at the selected Identity Provider and is redirected back to his/her profile

v20.10.3 - 2020-10-22

Changed

- Configure ORCID login plugin to use the ORCID member API which allows for reading limited-access information on an ORCID record (e.g., email) in addition to any information made publicly available by that record's holder.

v20.10.2 - 2020-10-14

Changed

- New release of VomsProvisioner plugin including support for the SOAP and REST VOMS Admin API. The implementation has been tested against VOMS Admin API v3.7.0. In addition, the new version of the VomsProvisioner plugin provides enhancements to the Plugin's Admin Configuration UI. The new features are:
  - Management of the Private Key and Certificate of the Robot user, which will handle the communication with VOMS

○ Checking of the Robot User's Certificate validity

○ Bulk import of VOMS servers using JSON format

v20.10.1 - 2020-10-08

Added

- Support for skipping email verification during user enrollment with Check-in. The email verification step is only omitted if the user's authenticating Identity Provider (IdP) is already releasing a previously verified email address to Check-in. This can be determined based on the availability of specific attributes:

  ○ `voPersonVerifiedEmail` attribute in the case of SAML IdPs

  ○ `voperson_verified_email` claim in the case of OIDC IdPs

  ○ `email` claim combined with `email_verified` claim set to true in the case of OIDC IdPs

v20.09.2 - 2020-09-11

Added

- Support for checking certain attributes of a registered user's identity before allowing that user to request membership in a VO

Changed

- Encode hierarchical VO group structure in entitlements expressing VO group membership information according to AARC-G002 [[R10](#)]

- Display hierarchy of groups to VO manager when managing group membership through the Check-in Membership Registry UI

v20.09.1 - 2020-09-04

Changed

- Simplify the UI presented to users during initial registration with Check-in:

  ○ Hide menu options that were confusing for some users

  ○ Improve look and feel of the different enrollment steps

v20.07.4 - 2020-07-31

Added

- Community statistics view providing information about the communities/VOs registered in Check-in, including time of creation and the number of active and suspended members

Changed

- New version of RCauth certificate linking plugin does not rely on modifications to the core code of the Check-in Membership Registry (COmanage Registry)

v20.07.3 - 2020-07-16

Added

- Support for showing the VO-specific AUP (if available) during the VO enrollment flow

v20.07.2 - 2020-07-09

Added

- Support for service identities which can be used by applications/resources (e.g., portals); service identities are associated with one or more personal identities, i.e. the service identity owners.

v20.07.1 - 2020-07-07

Added

- Usage statistics view for:
  - Number of user logins (publicly accessible)
  - Number of user logins per Identity Provider and Service Provider (accessible by group of privileged users only)
  - Number of user registrations (accessible by group of privileged users only)

v20.05.2 - 2020-05-15

Changed

- Simplify user registration flow for users coming from Community AAIs (e.g. eduTEAMS, DEEP Hybrid DataCloud):
  - Users are not required to validate their email address given that this is already verified by the Community AAI
  - Check-in stores the user's Community identifier from the eduPersonUniqueId attribute released by the Community AAI
- Configure GOCDB module to fail over to the backup instance of the GOCDB if the primary one fails to return role information
- Show warning page if role information cannot be retrieved from the GOCDB and allow user to continue accessing the service without role information

v20.05.1 - 2020-05-04

Changed

- Improve user experience during enrollment in Check-in through the display of a circular progress indicator for visualising the status of the enrollment request (e.g., signup or request to join VO)

v20.02.1 - 2020-02-17

Added

- Support for Proof Key for Code Exchange (PKCE): PKCE is a technique for securing public clients that do not use a client secret (https://tools.ietf.org/html/rfc7636)

- Support for redirecting the user to the end service they were trying to access after completing the user registration process

Changed

- Improve error message when a non-registered user follows an enrollment URL for requesting to join a VO. The updated error message includes a link to signup which is required for joining VOs.

A.3 eduTEAMS

Service/Tool name	eduTEAMS
Service/Tool url	http://www.eduteams.org
Service/Tool information page	https://wiki.geant.org/display/eduTEAMS
Description	eduTEAMS enables researchers, students and other members of the research and education community to create and manage virtual teams and securely access and share common resources and services using federated identities from eduGAIN and trusted Identity Providers.
Value proposition	The eduTEAMS service enables research communities to securely access and share common resources and services. Leveraging the ubiquitous presence of eduGAIN federated identities, eduTEAMS enables communities to securely authenticate and identify their users, organize them in groups, assign them roles and centrally manage access rights for using community resources. As research is not confined only to research institutes and universities, eduTEAMS caters also for users coming from the industry or citizen scientists who may not have access to eduGAIN. It does so by supporting external (non-eduGAIN) identity providers, such as social networks providing federated identities, community identity providers and other platforms that can provide federated user identities. Communities can use the eduTEAMS service as the community AAI for their virtual collaborations.
Customer of the service/tool	Research Infrastructures, Research Communities

User of the service/tool	Community/VO managers, researchers, students, faculty of academic institutions, IT support staff for RIs/RCs
User Documentation	https://wiki.geant.org/display/eduTEAMS
Technical Documentation	https://wiki.geant.org/display/eduTEAMS
Product team	GÉANT
License	Not applicable
Source code	eduTEAMS is based on open-source software: https://github.com/IdentityPython/ https://github.com/CESNET/perun https://github.com/CESNET/perun-services https://github.com/CESNET/perun-wui
Release notes	
Testing	The GÉANT Service Quality Assurance team provides QA testing to the GÉANT service. The QA involves: • Quality code audit - automatic code review completed by the code inspection (expert analysis) to examine the source code and identify potential bugs, bad code architecture, duplicated code and similar coding irregularities. • Security code audit - automatic code review completed by the code inspection (expert analysis) to examine the source code and identify the largest possible number of source code security flaws and vulnerabilities. • Vulnerability assessment (aka security testing) - a thorough process of system security testing from a user's as well as inside and outside (black-box) point of view together with testing of the underlying operating system, other software package dependencies and its configurations. • Documentation evaluation - usually the first sanity check aiming to help to identify early potential risks (i.e., the required documentation is missing), spaces for improvements and possibilities for optimizing the system (i.e., desirable documentation is missing). • Operational testing - in-depth review of the operational documentation against the completeness, correctness, and comprehensiveness. Someone not familiar with the service will try to reproduce all steps listed in the documentation and verifies the outcome.

	• Functional and user interface testing - it is composed of the usability and accessibility testing mixed with some elements of functional tests of the user interface and the web user interface. • Performance testing - to measure how the system behaves in various predefined conditions, to check if the service meets the expected KPI and to identify potential bottlenecks.
TRL	9

A.5 Perun

Service/Tool name	Perun
Service/Tool url	https://perun.egi.eu/
Service/Tool information page	https://perun-aai.org/
Description	Perun is an Identity and Access management software that covers management of the whole ecosystem around the users' identities, groups, resources, and services. Perun is well suited for managing users within organizations and projects, managing access rights to the services. Perun is designed to be flexible and customizable, therefore it can be easily integrated with other tools or incorporated into existing workflows. Moreover, Perun stresses decentralization of authorization decisions by empowering end users to manage groups within it and delegate this privilege to other users.
Value proposition	Identity and Access management system that can be offered as a standalone tool or it can be integrated with other EOSC-hub components like authentication proxies and delivered as an integrated service offer. Perun supports advanced features and use-cases like self-service, privilege delegation, account linking, provisioning and deprovisioning or integration with CSIRT.
Customer of the service/tool	Research Communities, Research Infrastructures
User of the service/tool	Virtual Organization Managers, Services Managers, Virtual Organization members, Members of CSIRT
User Documentation	https://perun-aai.org/documentation/user-documentation
Technical Documentation	https://perun-aai.org/documentation/technical-documentation
Product team	CESNET

License	BSD 2-Clause
Source code	https://github.com/CESNET/perun https://github.com/CESNET/perun-services https://github.com/CESNET/perun-wui
Release notes	https://github.com/CESNET/perun/releases
Testing	Automatic unit and integration tests are part of the development and deployment process. The code review is a part of the development process. Regular penetration testing every second year.
TRL	9

A.6 WaTTS

Service/Tool name	WaTTS
Service/Tool url	Prod: https://watts-prod.data.kit.edu Devel: https://watts.data.kit.edu
Service/Tool information page	https://watts-prod.data.kit.edu/docs/user/index.html
Description	WaTTS is a flexible and scalable Token Translation Service, supporting, among others, IGTF compatible (IOTA) X.509 certificates, including certificates obtained from an online CA (such as RCauthCA). Additionally, it supports plugin-based functionalities, where access to these functionalities is discriminated based on the received attributes. Additional functionalities include deployment of SSH keys, obtaining SSH certificates (in conjunction with having an SSH CA), access to storage service (e.g., object storage), and others.
Value proposition	Allow to (transparently) create X.509 certificates for a user. This makes usage of grid infrastructures easier (the user does not have to see the certificate). It also supports providing VOMS certificates to users and services. Furthermore, the service supports REST API, and therefore can provide these certificates via REST (which includes CLI access).
Customer of the service/tool	Research Communities
User of the service/tool	End-users accessing R/e-Infrastructure services using either PKIX or a combination of PKIX and SSH credentials.
User Documentation	https://watts-prod.data.kit.edu/docs/user/index.html
Technical Documentation	https://watts-prod.data.kit.edu/docs/code/index.html

Product team	KIT
License	Apache License Version 2.0
Source code	https://github.com/watts-kit/
Release notes	https://github.com/watts-kit/watts/releases

Testing	Visit page, use plugins:
	  − Info Plugin for minimal testing
	  − X.509 Plugin for X.509 certificate

TRL	9

A.7 MasterPortal

Service/Tool name	MasterPortal (reference service)
Service/Tool url	multiple instances accessible via REST API (no typical UI): https://aai.egi.eu/; https://masterportal-pilot.aai.egi.eu/ (EGI development instance); https://elevator.nikhef.nl/; https://*elixir*-cilogon-mp.grid.cesnet.cz/; *others* SSH proxy access interface: https://aai.egi.eu/sshkeys/
Service/Tool information page	https://wiki.nikhef.nl/grid/RCauth.eu_and_MasterPortal_documentation
Description	Provides a Token Translation capability from (primarily) SAML to X.509 leveraging the RCauth online CA and enabling pure web-based portals to access X.509 resources on behalf of their users. It forms a transparent caching service between Science Gateways and the RCauth online CA, handling the complexity of obtaining certificates for the Science Gateways and end-users. Additionally, it provides the capability to upload SSH public keys and to retrieve proxy certificates using those.
Value proposition	Allowing the use of X.509-based credentials, while hiding all the complexity for the end-users. An ancillary capability allows authentication to community portals and science gateways via OpenID Connect for users usually authenticating via SAML (implicit SAML-to-OIDC translation) when used in conjunction with the RCauth.eu operational service.
Customer of the service/tool	Either Science Gateways needing X.509 credentials, or 'power-users' that can leverage SSH key authentication to obtain proxy certificates.
User of the service/tool	End-users accessing R/e-Infrastructure services using either PKIX or a combination of PKIX and SSH credentials.

User Documentation	https://wiki.nikhef.nl/grid/RCauth.eu_and_MasterPortal_SSH_Key_Portal - end-users https://wiki.nikhef.nl/grid/RCauth.eu_and_MasterPortal_VOPortal_integration_guide - VOportal developers/operators
Technical Documentation	https://wiki.nikhef.nl/grid/RCauth.eu_and_MasterPortal_documentation
Product team	Nikhef, GRNET
License	Apache License Version 2.0
Source code	https://github.com/rcauth-eu
Release notes	https://github.com/rcauth-eu/aarc-master-portal/blob/0.2.1-release/RELEASE-NOTES.md
Testing	Each subcomponent comes with unit tests that are run after each release candidate build. The integration test is performed using an ansibleised virtual container environment (accessibility testing of the operational is performed with nagios from within the operating site)
TRL	9

A.8 RCauth - Online CA

Service/Tool name	RCauth.eu
Service/Tool url	http://pilot-ca1.rcauth.eu/
Service/Tool information page	https://rcauth.eu/
Description	The RCauth.eu service is a token translation service (TTS) that can on-the-fly identify entities based on federated credentials and issue them PKIX credentials in real-time, focussing on converting SAML-to-PKIX. Primarily intended as an operational resource for user and community-facing credential management portals, such as WaTTS and other 'master portals', it provides an OpenID Connect authenticated capability to provide globally trusted PKIX credentials at the DOGWOOD [RFC6711] assurance profile.
Value proposition	Allows token translation services and BPA proxy components to completely hide the use of PKIX credential issuance from the end-user.

Customer of the service/tool	AARC BPA Proxy and token translation service operators on behalf of both Research and generic e-Infrastructures.
User of the service/tool	End-users accessing R/e-Infrastructure services by means of PKIX credentials
User Documentation	MasterPortal operators: https://wiki.nikhef.nl/grid/Master_Portal_Administrator_Guide Science GateWay operators/developers: https://wiki.nikhef.nl/grid/RCauth.eu_and_MasterPortal_VOPortal_integration_guide End-users: Not applicable
Technical Documentation	https://www.rcauth.eu/tech-resources
Product team	Nikhef, GRNET, STFC
License	Apache License Version 2.0
Source code	https://github.com/rcauth-eu
Release notes	https://github.com/rcauth-eu/aarc-delegation-server/blob/0.2.1-release/RELEASE-NOTES.md
Testing	Each subcomponent comes with junit tests that are run after each release candidate build. The integration test is performed using an Ansible virtual container environment (accessibility testing of the operational is performed with nagios from within the operating site)
TRL	8

A.9 EOSC Portal AAI

Service/Tool name	EOSC Portal AAI
Service/Tool url	https://aai.eosc-portal.eu/proxy
Service/Tool information page	https://wiki.egi.eu/wiki/EOSC_Portal_AAI
Description	The EOSC Portal AAI enables researchers in Europe to use their Research Community AAI or academic/social account of choice for obtaining information on EOSC, as well as to provide feedback,

	discover and request services and resources that are offered through the EOSC portal.
Value proposition	The EOSC Portal AAI is connected to multiple Community AAIs to allow researchers to access the underlying services and resources using their community identity, including their roles and other authorisation-related information managed by the community. In addition to the Community AAIs, the EOSC Portal AAI is connected to the upstream home organisation IdPs (e.g., from eduGAIN/social) to enable researchers to access services and resources as members of their home organisation. Therefore, both community-based and home organisation-based access scenarios can be supported.
Customer of the service/tool	Research Infrastructures, Research Communities, Resource Providers
User of the service/tool	Community/VO managers, researchers, Operations Managers for research infrastructures/collaborations
User Documentation	https://wiki.egi.eu/wiki/EOSC_Portal_AAI#Documentation
Technical Documentation	https://wiki.egi.eu/wiki/EOSC_Portal_AAI_guide_for_SPs https://wiki.egi.eu/wiki/EOSC_Portal_AAI_guide_for_IdPs
Product team	GRNET
License	Apache License Version 2.0
Source code	https://github.com/rciam https://github.com/EGI-Foundation/simplesamlphp-module-themeeosc
Release notes	See Section A9.1
Testing	Functional and user interface testing is performed before every change. Higher risk changes are reviewed by the EOSC-hub Change Advisory Board before being released in production.
TRL	9

A9.1 Release notes

v21.03.1 - 2021-03-03

Changed

- Upgrade SAML IdP/SP proxy to support new encryption algorithms (AES-GCM/CBC) used by Shibboleth IdP v4 IdPs
- Upgrade PostgreSQL-based DB cluster to v11

v20.12.1 - 2020-12-10

Added

- Support for OAuth2 Device Authorization Grant (RFC8628)
- Publishing of "refresh_token" in "grant_types_supported"
- Support for Proof Key for Code Exchange (PKCE) (RFC7636)
- orcid scope and claim
- email_verified and voperson_verified_email claims

Fixed

- Error message response type when creating a new client with duplicate client id
- Missing ContentType in generate-oidc-keystore response
- Remove empty claims from Introspection endpoint response
- Disable HTML escaping in json response

v20.10.1 - 2020-10-22

Changed

- Configure ORCID login plugin to use the ORCID member API which allows for reading limited-access information on an ORCID record (e.g., email) in addition to any information made publicly available by that record's holder.

A.10 Marketplace

Service/Tool name	Marketplace
Service/Tool url	https://marketplace.eosc-portal.eu
Service/Tool information page	https://wiki.eosc-hub.eu/display/EOSC/Marketplace
Description	Marketplace (MP) is a user-facing platform where productional EOSC-hub services can be promoted, discovered, ordered and accessed. A set of functionalities implemented in Marketplace supports efficient order

	management and facilitates the interactions of users with e-infrastructures.
Value proposition	Common platform to facilitate activities of service users, customers, and providers in scope of EOSC services. It provides functionality to support the full user path between service discovery and service access. It brings an environment for service providers to appropriately manage offers of their resources and services. It follows best practices of UX to ensure best user experience.
Customer of the service/tool	Researchers, Research Groups, Business Representative
User of the service/tool	Researchers, Research Groups, Business Representatives, Service Owners, Service Providers
User Documentation	https://wiki.eosc-hub.eu/display/EOSC/Marketplace (Work in Progress)
Technical Documentation	https://github.com/cyfronet-fid/marketplace
Product team	ACC Cyfronet AGH
License	Apache License Version 2.0
Source code	https://github.com/cyfronet-fid/marketplace
Release notes	https://github.com/cyfronet-fid/marketplace/blob/master/CHANGELOG.md
Testing	Unit and Integration testing integrated within the MP RoR application is a part of the development and deployment process (Travis CI based). The code review is a part of the development process. Functional and user interface testing is performed before every release. New features are approved by WP2 & WP4 before being released in Production.
TRL	9

A.11 Service Portfolio Management Tool (AGORA)

Service/Tool name	AGORA/SPMT
Service/Tool url	https://eosc.agora.grnet.gr & https://eosc-hub-devel.agora.grnet.gr
Service/Tool information page	https://grnet.github.io/agora-sp/
Description	The Service Portfolio Management Tool (SPMT/AGORA) provides a full list of services and allows managing service descriptions according to the service management guidelines of FitSM.
Value proposition	It manages service descriptions to the granularity of service components and allows the service management according to the guidelines of FitSM. The SPMT also allows to export service descriptions to other tools and service catalogues, such as the one to be established by the eInfraCentral project and https://www.eosc-hub.eu/catalogue
Customer of the service/tool	Service Providers, Resource Provider; Research Communities
User of the service/tool	Service Providers, Service Portfolio Managers
User Documentation	https://grnet.github.io/agora-sp/
Technical Documentation	https://grnet.github.io/agora-sp/
Product team	GRNET
License	AGPL-3.0
Source code	https://github.com/grnet/agora-sp https://github.com/grnet/agora-sp-admin https://github.com/grnet/agora-probes https://github.com/grnet/agora-catalogue-react-view
Release notes	https://github.com/grnet/agora-sp/blob/master/CHANGELOG.md
Testing	Unit and Integration testing is performed on the Staging instance (https://eosc-hub-devel.agora.grnet.gr). New features are approved by WP2 & WP4 before being released in Production.
TRL	8

A.12 Operations Portal

Service/Tool name	Operations Portal
Service/Tool url	http://operations-portal.egi.eu http://operations-portal.egi.eu/vapor
Service/Tool information page	https://wiki.egi.eu/wiki/Operations_Portal
Description	The Operations Portal provides VO management functions and other capabilities which support the daily operations of EGI. It is a central portal for the operations community that offers a bundle of different capabilities, such as the broadcast tool, VO management facilities, different dashboards that are used to display information about failing monitoring probes and to open tickets to the Resource Centres affected. The dashboards also support the central grid oversight activities. It is fully interfaced with the EGI Helpdesk and the monitoring system through messaging. The Operations Portal provides tools supporting the daily running of operations of the entire infrastructure: Infrastructure oversight, security operations, VO management, broadcast, availability reporting.
Value proposition	- Improve and enrich existing tools - Adapt or develop tools with needs expressed by new communities - Adapt or develop tools within the evolution of the EOSC environment
Customer of the service/tool	RI; Resource Provider; Research Communities; Virtual Organisations
User of the service/tool	Site admins; Operations Managers; Virtual Organisations; large research group
User Documentation	Home · Wiki · OpsPortal / sf3
Technical Documentation	Home · Wiki · OpsPortal / sf3
Product team	CNRS
License	Apache License Version 2.0

Source code	https://gitlab.in2p3.fr/opsportal/sf3
Release notes	https://operations-portal.egi.eu/home/tasksList
Testing	Automated Tests: https://forge.in2p3.fr/projects/opsportaluser/wiki/Continuous_Integration Release procedure: https://wiki.egi.eu/wiki/PROC23
TRL	9

A.13 GOCDB

Service/Tool name	GOCDB
Service/Tool url	https://goc.egi.eu
Service/Tool information page	https://wiki.egi.eu/wiki/GOCDB
Description	GOCDB is a central registry to record information about the topology of an e-Infrastructure. This includes entities such as resource centers (sites), services, service-endpoints, and their downtimes, contact information and roles of users responsible for operations at different levels. The service enforces a number of business rules and defines different grouping mechanisms including object-tagging for the purposes of fine-grained resource filtering.
Value proposition	GOCDB is a key tool for the configuration management of the EGI Federation and WLCG. It is a definitive information source, with the emphasis on user communities to maintain their own data. It is intentionally designed to have no dependencies on other operational tools for information.
Customer of the service/tool	EGI Operations and WLCG
User of the service/tool	Site/service admins, NGI managers and Security teams
User Documentation	https://wiki.egi.eu/wiki/GOCDB
Technical Documentation	https://wiki.egi.eu/wiki/GOCDB

Product team	UKRI-STFC
License	Apache License Version 2.0
Source code	https://github.com/GOCDB/gocdb
Release notes	https://github.com/GOCDB/gocdb/releases
Testing	Before every production release, GOCDB development is frozen, and a period of testing is announced that lasts for approximately two weeks to one month using the GOCDB test instance. This testing phase is widely disseminated using the relevant mail lists, and all operational tools and users are invited to perform tests against this instance.
TRL	9

A.14 Data Project Management Tool

Service/Tool name	Data Project Management Tool (DPMT)
Service/Tool url	https://dp.eudat.eu
Service/Tool information page	https://github.com/EUDAT-DPMT
Description	DPMT is EUDAT's internal coordination tool. Information about providers and customers as well as the projects that they are engaged in are documented in DPMT. EUDAT is currently running services, service components and resources provided through them are registered with the DPMT.
Value proposition	DPMT is a web-based portal application designed to allow new and existing data projects to be enabled, managed, and monitored with the help of the partners of the EUDAT CDI. Machine agents can gather information about all EUDAT services, service components and resources through an API that is compatible with the GOCDB API (see above). A central deployment of the DPMT serves the entire EUDAT CDI reducing the maintenance costs. Through multiple, tailor made interfaces it supports easy and effective interoperability with EOSC's operational tools.
Customer of the service/tool	EUDAT's Service and Resource Providers; Research Communities

User of the service/tool	Site admins; Operations Managers; Project PIs; Community Managers
User Documentation	not applicable
Technical Documentation	https://github.com/EUDAT-DPMT https://dp.eudat.eu/help https://gitlab.mpcdf.mpg.de/rjr/dpmt-config/wikis/operation (not public)
Product team	MPCDF
License	GPL Version 2.0
Source code	https://github.com/EUDAT-DPMT
Release notes	https://github.com/EUDAT-DPMT/pcp.contenttypes/commits/py3 https://github.com/EUDAT-DPMT/dpmt_buildout_p5/commits/py3 (changelogs of recent and current development)
Testing	MPCDF operates a development instance of the DPMT where all new features and components can be demonstrated and tested before they are rolled out in production.
TRL	8

A.15 Data Management Planning Tool

Service/Tool name	EasyDMP
Service/Tool url	https://easydmp.eudat.eu
Service/Tool information page	https://www.sigma2.no/data-planning
Description	EasyDMP is an EUDAT tool for creating data management plans. The tool also makes use of the EESTORE a service which a uniform interface to information from third-party registries that are required when completing a data management plan.

Value proposition	Provides a configurable web interface that makes it easier for researchers to create data management plans. The intention is to further integrate with EUDAT services to allow the provisioning of services as part of the creation of the data management plan rather than having the two activities (creating a plan and provisioning services) separated. This will also enable the plan to be verified at a later date (i.e., is the project following the approved plan?).
Customer of the service/tool	Researchers, Resource providers
User of the service/tool	Researchers
User Documentation	https://www.sigma2.no/easydmp/how-to
Technical Documentation	https://github.com/hmpf/easydmp https://gitlab.eudat.eu/dmp/eestore
Product team	Sigma2 (EOSC-Hub), Athena Research and Innovation Centre (OpenAIRE)
License	MIT
Source code	https://github.com/hmpf/easydmp https://gitlab.eudat.eu/dmp/eestore
Release notes	https://gitlab.eudat.eu/dmp/eestore/-/blob/stable/2.0.x/CHANGELOG.rst for the eestore https://github.com/hmpf/easydmp/releases for the easydmp
Testing	The code makes use of the Django unit test framework. The tests are run before each release. New tests are created based on feedback from users.
TRL	7

A.16 Service Versions Monitoring Tool

Service/Tool name	SVMON
Service/Tool url	https://svmon.eudat.eu
Service/Tool information page	(in progress)
Description	SVMON collects software versions of EUDAT services and their corresponding components in EUDAT CDI.
Value proposition	SVMON collects information on software versions, stores and displays collections in a compact view. SVMON also uniquely provides information about service attributes.
Customer of the service/tool	EOSC-hub customers
User of the service/tool	EUDAT service providers, site administrators
User Documentation	https://wiki.eosc-hub.eu/pages/viewpage.action?pageId=61374702 (in progress)
Technical Documentation	https://gitlab.eudat.eu/jie.yuan/svmon-app/ (in progress)
Product team	KIT
License	Apache License 2.0 The MIT License Copyright (c) 2014-2018 Google, Inc.
Source code	https://gitlab.eudat.eu/jie.yuan/svmon-app
Release notes	https://gitlab.eudat.eu/EUDAT-TOOLS/SVMON/svmon-app
Testing	Unit test, functions test A testing instance https://svmon-dev-test.scc.kit.edu
TRL	8

A.17 Accounting Repository

Service/Tool name	APEL
Service/Tool url	http://apel.github.io/
Service/Tool information page	https://wiki.egi.eu/wiki/Accounting_Repository
Description	The Accounting Repository stores compute (serial and parallel jobs), storage, and cloud resource usage data collected from Resource Centres of the EGI and EUDAT infrastructures. Accounting information is gathered from distributed sensors into a central Accounting Repository where it is processed to generate summaries that are available through the Accounting Portal.
Value proposition	Combined reporting of EGI and EUDAT storage resource usage, giving unified EOSC-hub usage accounting. Improvements to the client-side software making it easier to operate and enabling problems to be diagnosed more rapidly.
Customer of the service/tool	RI; Resource Provider; Research Communities
User of the service/tool	Site admins; Operations Managers; large research groups
User Documentation	https://wiki.egi.eu/wiki/APEL
Technical Documentation	https://wiki.egi.eu/wiki/APEL
Product team	STFC
License	Apache License, Version 2.0
Source code	https://github.com/apel/apel (client and server software) https://github.com/apel/ssm (messaging tool)
Release notes	https://github.com/apel/apel/releases (client and server software) https://github.com/apel/ssm/releases (messaging tool)
Testing	The APEL project uses a development workflow based around GitHub, which includes a semi-automatic testing procedure used to assess the quality of software releases. This procedure comprises automated unit tests and code quality checks, peer review, test builds, testing on a pre-production system, and deployment to test sites.

TRL	9

A.18 Accounting Portal

Service/Tool name	EGI Accounting Portal
Service/Tool url	https://accounting.egi.eu/
Service/Tool information page	https://wiki.egi.eu/wiki/Accounting_Portal
Description	The Accounting Portal provides data accounting views for users, VO Managers, NGI operations and the general public.
Value proposition	The Accounting Portal acts as an interface to different accounting records, integrating them with other data and metadata from several providers and presents a homogeneous view of the data gathered and a user-friendly access.
Customer of the service/tool	VO Managers, NGI operations and the general public
User of the service/tool	VO Managers, NGI operations and the general public
User Documentation	https://accounting.egi.eu/static/EGI%20Accounting%20Portal%20User's%20Guide.pdf
Technical Documentation	https://wiki.egi.eu/wiki/Accounting_Portal_API
Product team	CESGA
License	Apache License Version 2.0
Source code	https://github.com/cesga-egi/accounting
Release notes	- AAI Implementation using mod_auth_mellon and SAML - Cloud topology was separated from the HTC one - Topology backend was changed to support CRIC from WLCG as a new topology source. - New WLCG long form federation names are also supported, as are Jupyter Notebook sites. - Metrics in normal views now support multipliers.

	● Tier1 REBUS report inherited from Rebus was removed as the new version is in CRIC. ● EUDAT storage accounting was implemented as a static report. ● Storage accounting was also implemented but waiting for summarisation to be moved to APEL to move to production.
Testing	Testing using development version and pre-production version by a dedicated EGI Operations Tools Advisory Group.
TRL	9

A.19 ARGO Monitoring

Service/Tool name	ARGO Monitoring
Service/Tool url	http://argo.egi.eu
Service/Tool information page	https://wiki.egi.eu/wiki/ARGO
Description	ARGO is a flexible and scalable framework for monitoring status, availability, and reliability
Value proposition	ARGO provides monitoring of services, visualization of their status, dashboard interfacing, notification system and generation of availability and reliability reports. The dashboard design enables easy access and visualisation of data for end-users. Third parties can gather monitoring data from the system through a complete API. A central deployment of the ARGO monitoring engine can serve a large infrastructure reducing the maintenance costs.
Customer of the service/tool	RI; Resource Provider; Research Communities
User of the service/tool	Site admins; Operations Managers; large research group
User Documentation	http://argoeu.github.io; http://argo.egi.eu
Technical Documentation	http://argoeu.github.io
Product team	GRNET, SRCE, CNRS

License	Apache License Version 2.0
Source code	https://github.com/ARGOeu/
Release notes	https://github.com/ARGOeu/argo-web-api/releases https://github.com/ARGOeu/argo-alert/releases https://github.com/ARGOeu/argo-ncg/releases https://github.com/ARGOeu/poem-2/releases https://github.com/ARGOeu/argo-nagios-ams-publisher/releases https://github.com/ARGOeu/argo-egi-connectors/releases https://github.com/ARGOeu/argo-streaming/releases
Testing	ARGO Monitoring follows a development process where tests that check the functionality and the quality, correctness of the software are mandatory. This process consists of automated unit tests and code quality checks, running via a CI tool (jenkins). All main components (where applicable) of ARGO monitoring follow the same approach. The types of tests are: ● [Connectors] - Unit tests for all different functionalities for connectors. ● [POEM] - There are currently two apps in Poem project: poem and api. For both of these apps unit tests (python) that test the functionality are supported. ● [Compute Engine] - End-to-end *testing* of all *Flink jobs.* Unit tests for batch and streaming jobs of the compute engine. ● [WEB-API] - Unit tests that test crud and domain logic functionality on all resource objects supported by the api, using mock interfaces on the datastore and broker layers. (golang testify) ● [WEB-API] - External test: Web API endpoints are tested as postman collections via newman. Newman [R3] is a command-line collection runner for Postman [R4]. It allows you to effortlessly run and test a Postman Collections [R5] directly from the command-line. It is built with extensibility in mind, and it can be easily integrated with ARGO's continuous integration server and build systems. ● [argo-alerts] - Unit tests that gather data from GOCDB and create contact lists to send the alerts.

TRL	9

A.20 Argo Messaging Service

Service/Tool name	ARGO Messaging Service (AMS)
Service/Tool url	http://argoeu.github.io
Service/Tool information page	https://wiki.egi.eu/wiki/Message_brokers
Description	AMS enables reliable asynchronous messaging for the EOSC-hub infrastructure
Value proposition	AMS provides a scalable HTTP Messaging Service with: ● An HTTP API for client access ● Transparent scalability & high availability ● Access controls implemented at the API layer ● Multi-tenant support ● Instrumentation at the API layer
Customer of the service/tool	NGI; RI; Resource Provider; Research Communities
User of the service/tool	Site admins; Operations Managers; large research group
User Documentation	http://argoeu.github.io;
Technical Documentation	http://argoeu.github.io
Product team	GRNET, SRCE
License	Apache License Version 2.0
Source code	https://github.com/ARGOeu/
Release notes	https://github.com/ARGOeu/argo-messaging/releases https://github.com/ARGOeu/argo-ams-library/releases https://github.com/ARGOeu/ams-push-server/releases https://github.com/ARGOeu/argo-api-authn/releases

Testing	AMS follows a development process that includes mandatory tests for checking the functionality and the quality, correctness of the software. This process consists of automated unit tests and code quality checks, running via a CI tool (jenkins).
	The types of tests are:
	- Unit tests that test crud and domain logic functionality on all resource objects supported by the api, using mock interfaces on the datastore and broker layers. (golang testify) - External test: AMS endpoints are tested as postman collections via newman. Newman [R3] is a command-line collection runner for Postman [R4]. It allows you to effortlessly run and test a Postman Collections [R5] directly from the command-line. It is built with extensibility in mind and it can be easily integrated with ARGO's continuous integration server and build systems.
TRL	8

A.21 Pakiti

Service/Tool name	Pakiti
Service/Tool url	https://github.com/CESNET/pakiti-server https://github.com/CESNET/pakiti-client
Service/Tool information page	https://pakiti.egi.eu/ https://pakiti.cesnet.cz/egi/
Description	Pakiti provides a monitoring mechanism to check the patching status of Linux systems. Pakiti uses the client/server model, with clients running on monitored machines and sending reports to the Pakiti server for evaluation. The report contains a list of packages installed on the client system, which is subject to analysis done by the server. The Pakiti server compares versions against other versions which are obtained from various distribution vendors. Detected vulnerabilities identified using CVE identifiers are reported as the outcome, together with affected packages that need to be updated.

Value proposition	Proper security patch management is a crucial service to achieve a secure environment, yet it often is not straightforward to implement reliably. Pakiti detects missing security updates and notifies security teams and/or administrators so the vulnerabilities can be fixed before they cause security incidents.
Customer of the service/tool	RI; Resource Provider; NGIs
User of the service/tool	Site admins; Operations Managers; security teams of sites and infrastructures
User Documentation	https://github.com/CESNET/pakiti-server/tree/master/docs
Technical Documentation	https://github.com/CESNET/pakiti-server/tree/master/docs
Product team	CESNET
License	BSD 2-Clause
Source code	https://github.com/CESNET/pakiti-server https://github.com/CESNET/pakiti-client
Release notes	https://github.com/CESNET/pakiti-server/releases https://github.com/CESNET/pakiti-client/releases
Testing	manually controlled checks focused on handling typical tasks.
TRL	9

A.22 Secant

Service/Tool name	Secant
Service/Tool url	https://github.com/CESNET/secant
Service/Tool information page	https://github.com/CESNET/secant
Description	Secant is a security cloud assessment framework that is used to check security characteristics of virtual machines and their images. The framework instantiates the machine in a contained environment and runs a set of security probes against it. The probes combine external

	and internal checks and aim at typical configuration error or vulnerabilities commonly misused by Internet attackers.
Value proposition	Security of IaaS is largely determined by the running virtual clouds, so it is crucial the images used for their instantiation are securely configured. Secant makes it possible to reveal common errors and ease the maintenance of cloud images.
Customer of the service/tool	RI; Cloud Resource Provider; Communities
User of the service/tool	Site admins; Operations Managers; security teams of sites and infrastructures
User Documentation	https://github.com/CESNET/secant
Technical Documentation	https://github.com/CESNET/secant
Product team	CESNET
License	Apache License 2.0
Source code	https://github.com/CESNET/secant
Release notes	https://github.com/CESNET/secant/releases
Testing	Manually controlled checks focused on handling typical tasks.
TRL	6

A.23 GGUS

Service/Tool name	GGUS
Service/Tool url	https://ggus.eu
Service/Tool information page	https://wiki.egi.eu/wiki/GGUS
Description	GGUS helpdesk is a single point of contact for all EGI customers requesting help for fixing issues.

Value proposition	Besides WLCG GGUS covers a wide range of VOs and tool developers providing user support for their customers. It is connected to various ticketing systems of NGIs and infrastructures e.g., in the US.
Customer of the service/tool	EGI customers
User of the service/tool	Service providers, site admins, operations
User Documentation	https://ggus.eu/?mode=docu
Technical Documentation	https://wiki.egi.eu/wiki/GGUS
Product team	KIT
License	BMC Remedy (Closed source)
Source code	n.a.
Release notes	https://ggus.eu/index.php?mode=release_notes
Testing	https://test.ggus.eu/ggus/?mode=index
TRL	9

A.24 EUDAT-RT

Service/Tool name	EUDAT-RT
Service/Tool url	https://helpdesk.eudat.eu
Service/Tool information page	https://confluence.csc.fi/pages/viewpage.action?pageId=50874303 (page is protected)
Description	EUDAT-RT is the ticketing system used for EUDAT-CDI to manage the first level and 2nd level support request for all its services. The EUDAT-RT service is based on the Request Tracker software and it includes several support units to manage all the services of the EUDAT infrastructure.

Value proposition	The EUDAT-RT service is the main entry point for requests, problems, and incidents for the EUDAT infrastructure. The service supports federated access through B2ACCESS, and it is used by all the EUDAT staff and EUDAT users to submit and keep track of the problems concerning EUDAT services. The EUDAT-RT will be linked with the current EOSC-hub helpdesk system, based on xGUS, this integration will permit the management of tickets received on xGUS and assigned to EUDAT infrastructure. Any update on tickets generated on xGUS and migrated to EUDAT-RT will be automatically propagated to xGUS in order to have a full history of all the tickets on the xGUS TTS.
Customer of the service/tool	Research Communities, any user of EUDAT services.
User of the service/tool	Support units and 1st level support team of EUDAT.
User Documentation	https://confluence.csc.fi/download/attachments/50865867/eudat-TTS-Manual_2017v1.pdf?version=1&modificationDate=1502872233908&api=v2 (page is protected)
Technical Documentation	https://confluence.csc.fi/pages/viewpage.action?pageId=50874303 (page is protected)
Product team	BSC-CNS
License	RT- Request tracker from Best Practical - Version 2 of the GNU General Public *License*
Source code	https://bestpractical.com/download-page
Release notes	https://docs.bestpractical.com/release-notes/rt/5.0.0 No other changes have been done during the reporting period
Testing	Deploying a new version of the service requires tests for the following functions: ● creation of tickets ● movement and assignation of tickets to the different support units (queues) ● generation of e-mails from the system (send/recv) ● access to the system through B2ACCESS ● recovering of all the previous tickets and status (full RT DataBase comprovation)

TRL	9

A.25 xGUS

Service/Tool name	EOSC-hub helpdesk
Service/Tool url	https://helpdesk.eosc-hub.eu
Service/Tool information page	https://confluence.egi.eu/display/EOSC/xGUS (page is protected)
Description	EOSC-hub helpdesk is a single point of contact for all EOSC customers for requesting help for fixing issues.
Value proposition	EOSC customers do not need to know which infrastructure an issue is related to. They can submit their ticket in EOSC-hub helpdesk. It will be routed to the appropriate instances for fixing it.
Customer of the service/tool	EOSC-hub customers
User of the service/tool	Service providers, site admins, operations
User Documentation	https://ggus.eu/?mode=docu
Technical Documentation	https://wiki.egi.eu/wiki/GGUS
Product team	KIT
License	BMC Remedy (Closed source)
Source code	n.a.
Release notes	https://ggus.eu/index.php?mode=release_notes
Testing	https://test.ggus.eu/EOSC-hub
TRL	9

A.26 Applications Database

Service/Tool name	EGI Applications Database (AppDB)
Service/Tool url	https://appdb.egi.eu/
Service/Tool information page	https://wiki.egi.eu/wiki/AppDB
Description	The EGI Applications Database (AppDB) is a central service that stores and provides to the public information about: ● **software solutions** in the form of **native software products** and/or **virtual appliances**, ● the **programmers** and the **scientists** who are involved, and ● **publications** derived from the registered solutions ● enabling users to **deploy** and **manage Virtual Machines** to the EGI Cloud infrastructure through the VMOps Dashboard [R8] Reusing software products, registered in the AppDB, means that scientists and developers may find a solution that can be directly utilized on the European Grid & Cloud Infrastructures without reinventing the wheel. This way, scientists can spend less or even no time developing, porting or even using a software solution to the Distributed Computing Infrastructures (DCIs). AppDB, thus, aims to avoid duplication of effort across the DCI communities, and to inspire scientists less familiar with DCI programming and usage.
Value proposition	● Users can promote their software solutions and resources, reaching a large audience of peers, by registering them and describing them in a dedicated central database ● Users can reach a larger audience outside their peers, by having information related to their software solution propagated to other third-party services e.g., Resource Providers, ARGO, OpenAIRE, through interservice integration via its web-API ● Users gain a medium of directly interacting with the computing infrastructure in a graphical way.
Customer of the service/tool	RI; Resource Providers; Research Communities;
User of the service/tool	Site admins; Operations Managers; large research groups; Individual researchers

User Documentation	https://wiki.appdb.egi.eu/
Technical Documentation	https://wiki.appdb.egi.eu/
Product team	IASA
License	Apache License Version 2.0
Source code	https://github.com/iasa-gr
Release notes	https://wiki.appdb.egi.eu/main:about:appdb_release_notes_2020
Testing	Unit & functional tests performed on the AppDB development instance [R9].
TRL	9

A.27 GitLab

Service/Tool name	GitLab
Service/Tool url	https://gitlab.eudat.eu
Service/Tool information page	https://about.gitlab.com/
Description	GitLab is the first single application for the entire DevOps lifecycle.
Value proposition	GitLab provides an integrated environment for software development and continuous integration.
Customer of the service/tool	EOSC-hub customers
User of the service/tool	Research communities, individual researchers, service providers.
User Documentation	https://docs.gitlab.com/
Technical Documentation	https://docs.gitlab.com/
Product team	KIT
License	MIT License

Source code	https://gitlab.com/gitlab-org/gitlab-ce
Release notes	https://gitlab.com/gitlab-org/gitlab-foss/-/blob/master/CHANGELOG.md **2020-06-23:** Changed Update Gitlab to version 13.1.0 CPU Cores have been extended 4 CPUs core added (8 cores in total) **2019-09-02:** Changed Update Gitlab to version 13.2.8 Memory extended: 4 GB added **2019-12-24:** Changed Update Gitlab to version 13.7.1 Storage for repositories was extended, 50 GB added
Testing	Function tests (webview, api) are running in test environment on local test instance
TRL	9

A.28 EGI software repository

Service/Tool name	EGI Software Repository
Service/Tool url	http://repository.egi.eu/
Service/Tool information page	http://repository.egi.eu/about

Description	The EGI Software Repository ecosystem is a collection of services for supporting the management and the provisioning of the software artifacts that compose the **UMD** (Unified Middleware Distribution) and the **CMD** (Cloud Middleware Distribution), the **Community Repositories**, and the operational tools developed by the consortium. The following sub-services are included:
	- Repository back-end - Repository front-end - Composer - UMD, CMD & Community repositories
	The Repository back-end and the Composer services are the units within the EGI Software Repository ecosystem that are responsible for the construction of UMD and CMD releases and their related repositories.
	The Repository front-end is for making the produced repositories and all the required information, available to the public.
	Finally, the EGI Software repository is strongly integrated with the Application Database (AppDB). In this case, the AppDB acts as the backend "engine" for creating and managing the Community repositories populated through the EGI Software Repository system.
Value proposition	The EGI Software provisioning infrastructure (including RT) supports technology providers on their effort in delivering releases with respect to their products.
	From the other end, the provisioning infrastructure is responsible for supporting the verification of submitted releases, from a quality perspective, and for delivering ready-to-use repositories to the end-users, i.e., site admins, operation managers, and research communities.
Customer of the service/tool	RI; Resource Provider; Research Communities
User of the service/tool	Site admins; Operations Managers; large research group
User Documentation	http://repository.egi.eu/category/umd_releases/distribution/umd-4/ http://repository.egi.eu/category/os-distribution/cmd-os-1/ http://repository.egi.eu/category/one-distribution/cmd-one-1/
Technical Documentation	https://wiki.egi.eu/wiki/EGI_Software_Provisioning https://wiki.egi.eu/wiki/Middleware

	https://wiki.egi.eu/wiki/EGI_Cloud_Middleware_Distribution
Product team	IASA
License	Apache License Version 2.0
Source code	https://trac.iasa.gr/trac/egi-repo/
Release notes	- Added support for CentOS 8 and UMD 5.0 - Added support for TLS1.2+ - Added support for new EGI RT server - Changed behaviour to default to HTTPS - Added support for deployment via docker containers
Testing	Unit and integration tests are part of the development and deployment process. The code review is a part of the development process. In addition, there is a dedicated flow, under which, changes in the code that will potentially affect the smooth operation of the EGI repository are tested in a fully operational environment prior they are committed to the master branch and therefore pushed into production.
TRL	9

A.29 INDIGO IAM

Service/Tool name	INDIGO IAM
Service/Tool url	https://github.com/indigo-iam/iam
Service/Tool information page	https://www.indigo-datacloud.eu/identity-and-access-management
Description	The INDIGO IAM (Identity and Access Management) service provides user identity and policy information to services so that consistent authorization decisions can be enforced across distributed services.

Value proposition	● OpenID connect provider based on the MitreID OpenID connect library ● SCIM user provisioning and management APIs ● SAML authentication support ● OpenID Connect authentication support ● Flexible OAuth token exchange support
Customer of the service/tool	Research Communities
User of the service/tool	Users who need to translate credentials from different infrastructures and different authentication protocols.
User Documentation	https://indigo-iam.github.io/docs/v/current/user-guide
Technical Documentation	https://indigo-iam.github.io/docs/v/current/admin-guide
Product team	INFN
License	Apache License Version 2.0
Source code	https://github.com/indigo-iam/iam
Release notes	https://indigo-iam.github.io/docs/v/current/
Testing	Internal continuous integration test suite: https://sonarcloud.io/dashboard?id=indigo-iam_iam
TRL	9