D5.1 Initial maintenance and integration plan for federation and collaboration services

Lead Partner:	KIT
Version:	1
Status:	Under EC review
Dissemination Level:	Public
Document Link:	https://documents.egi.eu/document/3344

Deliverable Abstract
This document outlines the initial maintenance and integration plan for EOSC-hub federation and collaboration services. The plan has been established based on analysis of requirements coming from associated user communities, contributions of e-Infrastructure service providers. The deliverable provides high-level description of the architecture of each service in WP5 package, its maintenance plan and defines the initial integration activities and preliminary development roadmap. It elaborates the work plan for collaboration between WP5 and OpenAIRE initiative.

OSC-HUB RECEIVES FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME UNDER RANT AGREEMENT NO. 777536.

COPYRIGHT NOTICE

DELIVERY SLIP

	Name	***Partner/Activity***	***Date***
From:	Pavel Weber	KIT/WP5	27.07.2018
Moderated by:	Malgorzata Krakowian	EGI Foundation	
Reviewed by	Giacinto Donvito Diego Scardaci Johannes Reetz	INFN EGI Foundation MPG	
Approved by:	AMB		31.07.2018

DOCUMENT LOG

Issue	***Date***	***Comment***	***Author***
v 0.1	15.03.2018	Table of Content ready	KIT
v 0.2	22.03.2018	Description of coll. services in tasks 5.1,5.2,5.3,5.4, 5.5, 5.6 added	GRNET, BSC,CYFRONET, MPG,
v 0.3	02.04.2018	Description of coll. services added for all tasks	Pavel Weber
v 0.4	10.04.2018	Maintenance plans added for most of the services	Pavel Weber
v 0.5	15.04.2018	Integration plans added	Pavel Weber
v 0.6	30.04.2018	Correction of integration plans based on results of EOSC-hub AHM	Pavel Weber
v 0.7	05.05.2018	Security services added, finalization of all sections	Pavel Weber
v. 0.8	15.05.2018	All sections are provided	Pavel Weber
v 0.9	21.05.2018	Ready for external Review	Pavel Weber
v 1.0	27.07.2018	Implementation of corrections from external Review	Pavel Weber

TERMINOLOGY

https://wiki.eosc-hub.eu/display/EOSC/EOSC-hub+Glossary

AAI - Authorization and Authentication Infrastructure

AARC - Authentication and Authorisation for Research and Collaboration

AOD - Applications on Demand service

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

GGUS - Global Grid User Support

GOCDB - Grid Configuration Database

HA - High Availability

IAM - identity and access management system

IdP - Identity Provider

LoA - Level of Assurance

OIDC - OpenID Connect

OLA - Operational Level Agreement

SLA - Service Level Agreement

PID - Persistent Identifier

REFEDS - Research and Education FEDerations group

RTIR - Request Tracker for Incident Response

SP - Service Provider

Sirtfi - Security Incident Response Trust Framework for Federated Identity

SAML - Security Assertion Markup Language

VO - Virtual Organisation

VOMS - Virtual Organization Membership Service

WAYF - Where Are You From

Table of Contents

Executive summary

European Open Science Cloud (EOSC) is an ecosystem for open research and data and services, which aims to create a trusted environment for scientists for storing, processing and sharing the data. EOSC-hub contributes to EOSC by providing the central Hub with a set of services, policies and management system and acting as a central entry point for researchers to discover, access and use the EOSC resources. The services from the EGI e-Infrastructure, EUDAT Collaborative Data Infrastructure (CDI), INtegrating Distributed data Infrastructures for Global ExplOitation (INDIGO)-DataCloud and major research e-Infrastructures included in EOSC-hub are classified as Common, Thematic, Collaborative and Federation.

The focus of Work Package 5 (WP5) is on the collaborative and federation services. WP5 aims to seamlessly integrate these services and support their interoperability to create a framework that will enable the service federation in the EOSC. Further integration of the federation services with common (WP6) and thematic services (WP7) will be fostered during the project lifetime. Another major task of WP5 is to maintain the software and implement the changes and enhancements of the collaborative and federation services in order to improve their performance and functionality according to the evolving requirements of user communities. In addition, the WP5 services supports and leverages the implementation of EOSC-hub Federated Service Management System following and adopting the requirements developed in Work Package 4 (WP4).

The initial integration and maintenance plan reported in this deliverable defines the integration and development roadmap including software maintenance procedures for collaborative and federation services in the first year of the EOSC-hub project. The preparation work on initial plan and activities in WP5 during the first months was done in close collaboration with WP10, which is responsible for technical coordination of EOSC-hub project, with WP4 to follow the operational policies and WP2, which is responsible for overall EOSC-hub project strategy. Below, the major achieved objectives of this recent work are outlined, the detailed description is given in the corresponding sections of this document.

The EOSC-hub Authentication and Authorization Infrastructure (AAI) will contribute to the EOSC infrastructure implementation roadmap by enabling seamless access to a system of research data and services. The EOSC-hub AAI will be built on existing AAI solutions from EGI Federation, EUDAT CDI, and INDIGO-DataCloud in cooperation with other relevant actors in the sector like GEANT. Currently, the AAI development and integration strategy for EOSC-hub is developed and being finalized. It is based on multi Blueprint Architecture (BPA) under definition by the Authentication and Authorisation for Research and Collaboration (AARC) project. The detailed integration plans for AAI services and tools are in place. The initial integration of Check-in and B2ACCESS AAI services together with harmonisation of user identifiers is accomplished.

EOSC-hub Marketplace is a user-facing platform that will facilitate promotion, discovery, ordering and access of the productional EOSC-hub services. The initial integration plan of Marketplace with other operational services is agreed and planned among all partners involved. Particularly relevant will be the integration with the Service Portfolio Management Tool (SPMT) that will allow the

marketplace to automatically retrieve and publish information about the services. SPMT and Marketplace instances are already deployed for EOSC-hub.

The initial Configuration Management Database (CMDB) for EOSC-hub will be made of two systems: the Grid Configuration Database (GOCDB) and the Data Project Management Tool (DPMT). The interconnection of both systems, GOCDB and DPMT, to Marketplace will allow the last to publish information about capacity and available service instances. Agreement to develop an uniform API integration interface to facilitate easy connection of CDMB to Marketplace, with focus on GOCDB and DPMT, is achieved and integrated in work plan. This interface will provide a scalable solution for connection of any community or Research Infrastructure (RI) CMDB to the Marketplace in future.

The Operations Portal is a central platform for the operations community that offers a bundle of different capabilities, such as the broadcast tool, Virtual Organization (VO) management facilities, a security dashboard and an operations dashboard that is used to display information about failing monitoring probes and to open tickets against underperforming Resource Centres. The integration plan for Operations Portal defines the interface and integration with Marketplace to facilitate and make semi-automatic the service order management and the resource request process.

The SVMON and Pakiti systems provide the possibility to monitor the software versions installed at data centers participating in EOSC-hub project. The focus of the Pakiti system is a security monitoring, while the SVMON facilitates release and deployment management. As both tools have much in common and provide complementary functionality on the client side, it was decided to integrate their clients.

The initial integration of SVMON client with Pakiti client is already accomplished, which allows running SVMON client as a module connected to Pakiti client. This approach unifies the installation of SVMON client together with Pakiti one and provides a service owner a simple way to choose which client to use or keep both active.

The initial plans for EGI and EUDAT accounting systems define the integration steps to be done to ingest the accounting data from EUDAT sites into APEL central repository for later visualization via the accounting portal.

ARGO is a flexible and scalable framework for monitoring status, availability and reliability of services provided by infrastructures with medium to high complexity. The monitoring systems based on ARGO framework are already successfully used by both EGI and EUDAT infrastructures. Thus, the integration plan determines the development of a web-portal with unified EOSC-hub view on monitoring information. The prototype of this web-portal is already deployed.

Two mature helpdesk systems GGUS for EGI and Request Tracker (RT) for EUDAT provide a complete helpdesk service for both infrastructures, used in the incident and service request management and also in the problem management processes. The integration plan for the helpdesk systems defines a unified system for EOSC-hub, based on developed lightweight clone of GGUS, which will permit a basic level of interoperability with GGUS and RT. The unified ticketing system will provide a centralized place to manage the first level support tickets for all users without losing the information of the tickets forwarded to the EUDAT or EGI helpdesk systems.

The Applications Database (AppDB) is a central service that stores and provides to the public information about software solutions, virtual appliances. In addition, AppDB is responsible for distributing the registered VM images to the resource providers and enabling users to deploy and manage Virtual Machines to the EGI Cloud infrastructure. The integration plan for AppDB includes the connection of AppDB dashboard with GGUS system in order to provide the possibility for the users to communicate their issues to site admins, integration of AppDB with EOSC-hub GitLab service to provide users the source code repository for any registered software item, AppDB integration with B2HANDLE, for enriching registered digital objects with persistent identifies (PIDS).

The work plans in the scope of collaboration with OpenAIRE are established for WP5 and mainly focused on integration of EOSC-hub AAI with OpenAIRE AAI, integration of AppDB with OpenAIRE Research Impact Dashboard and development of Data Management Planning (DMP) tool.

1 Introduction

This report of work package 5 provides an overview of EOSC-hub collaborative services and describes their initial integration and maintenance plans, which have been prepared and agreed among the WP5 partners. The implementation of many integration plans presented in this document has been already started in the first months of the project and some initial results have been already achieved, while the complex and strategic integration plans are being finalized.

The description of the collaborative services and related maintenance and integration plans grouped into 6 major sections according to the structure of WP5 divided into 6 tasks. Each section contains the overview of the services included in the corresponding task together with integration and maintenance plans. The last section provides a short summary of integration activities with OpenAIRE collaboration.

2 Identification, Authentication, Authorisation and Attribute Management

2.1 Overview of AAI services

2.1.1 Check-in

2.1.1.1 Check-in High-Level Service Description

The Check-in [R1] is the authentication and user management service for the EGI e-infrastructure. It enables users to access services (web and non-web based) using existing credentials from their home organisations. Specifically, Check-in comprises two main components, namely the Identity Provider (IdP)/Service Provider (SP) Proxy, later as **IdP/SP Proxy** and the **User Enrolment and Group/Virtual Organisation (VO) Management**.

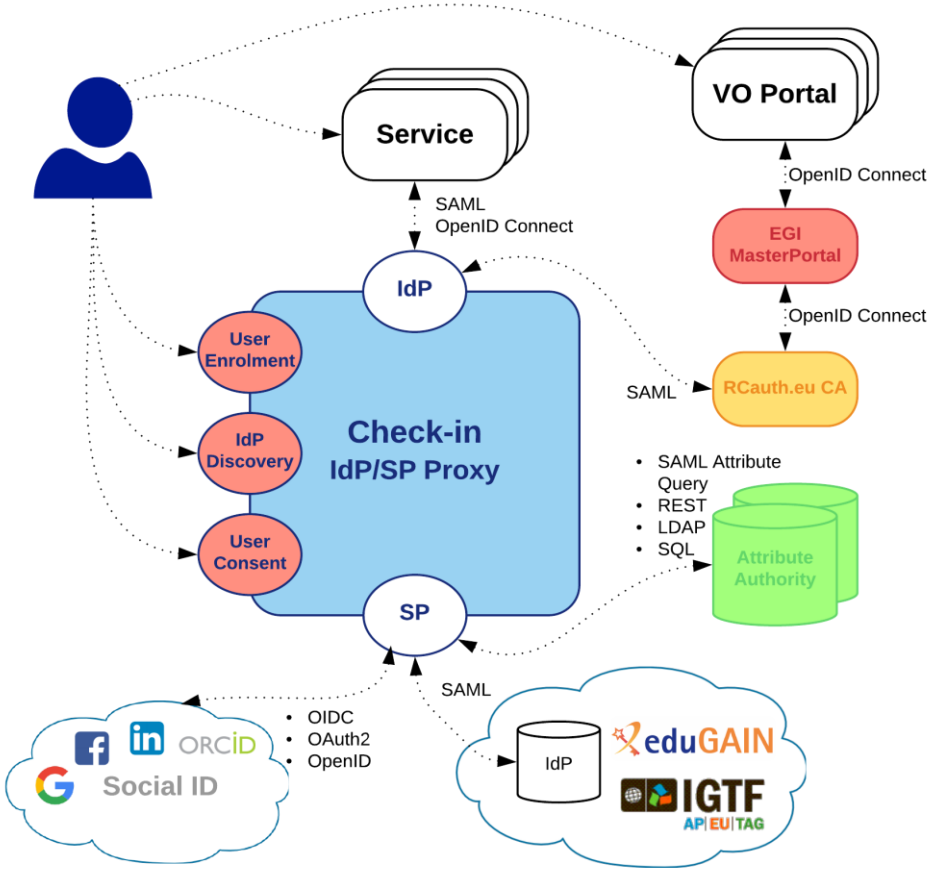

Figure 1 Check-In service

Figure 1 shows the Check-in top-level architecture diagram and interconnections to other AAI services, IdPs and tools. The IdP/SP Proxy component acts as a Service Provider towards the external Identity Providers and, at the same time, as an Identity Provider towards the Service Providers (e.g. GGUS, AppDB, etc). Through the IdP/SP proxy, users are able to sign in with the credentials provided by the IdP of their university or research institute that participates in eduGAIN [R2], as well as using social identity providers, or other selected external identity providers, such as Google, Facebook, LinkedIn, and ORCID. To achieve this, the proxy supports different authentication and authorisation standards, such as SAML 2.0, OpenID Connect (OIDC) 1.0 and OAuth 2.0. The proxy also provides a central Discovery Service (Where Are You From – WAYF) for users to select their preferred IdP. The core underlying software components of the proxy, namely SimpleSAMLphp [R3] and MITREid Connect [R4], have large user communities and a number of contributors from different organisations.

The User Enrolment and Group/VO Management component, which is based on COmanage [R5], supports the management of the full life cycle of user accounts in Check-in. This includes the initial user registration, the acceptance of the terms of use of the infrastructure, account linking, group and VO management, delegation of administration of VOs/Groups to authorised users and the configuration of custom enrolment flows for VOs/Groups via an intuitive web interface. For VOs, operating their own Group/VO Management system, the Check-in service has a comprehensive list of connectors that allows integrating their systems as externally managed Attribute Authorities.

2.1.1.2 Maintenance plan

The production operation of the EGI Check-in service involves 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 uninterruptible and performant operation. This activity will also be responsible for gathering new requirements for the improvement of the service.

2.1.2 B2ACCESS

2.1.2.1 High-Level Service Description

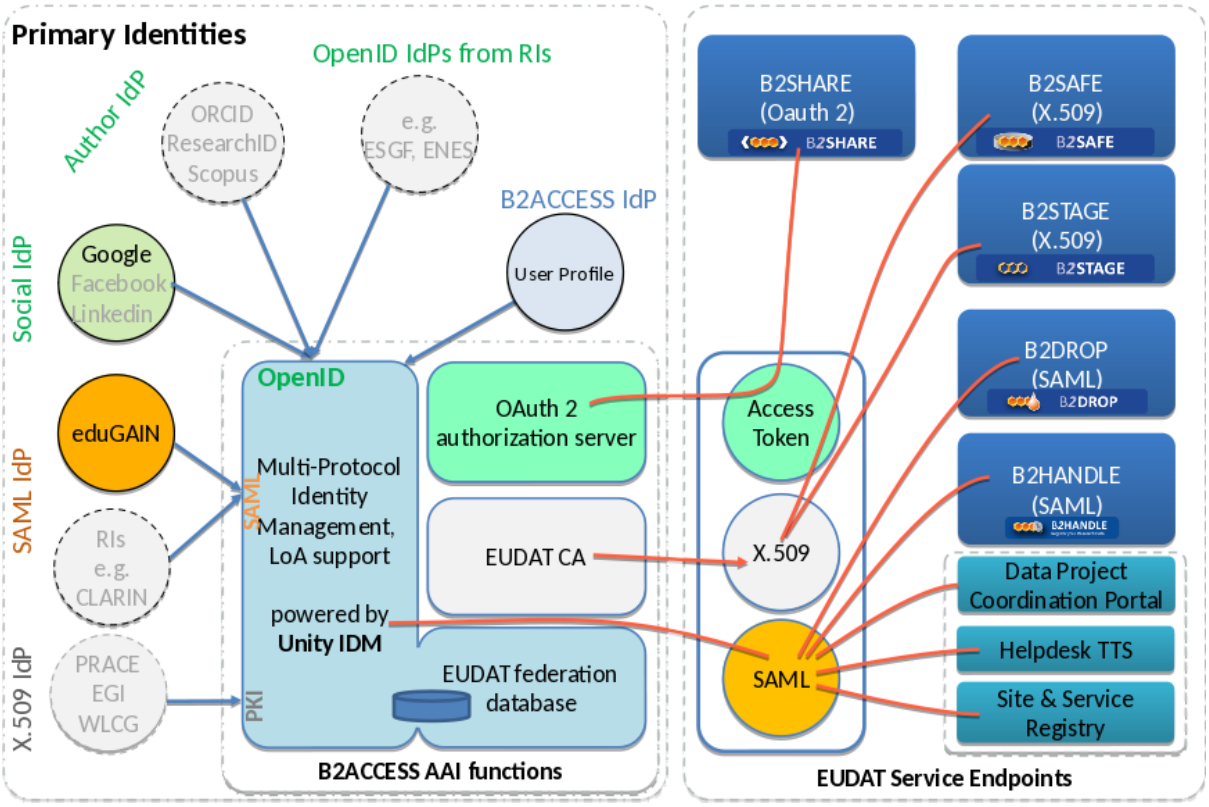

Figure 2 EUDAT AAI Description

B2ACCESS [R6] is an authentication and user management platform for EUDAT CDI [R7]. The B2ACCESS top-level architecture diagram with interconnections to other AAI services, IdPs and EUDAT services is shown in Figure 2. It is based on open source Unity Identity Management system [R8]. It follows Identity Management as a Service (IDaaS) approach by externalising the user authentication from protected services. It enables users to access (Web or non-Web Browser based) infrastructure services while authenticating with their home organisation provided credentials. Since B2ACCESS supports multiple authentication protocols (for example, X.509, OAuth 2.0/OIDC, SAML, LDAP and PAM), several types of credentials can be used. B2ACCESS specifically offers following main functions to the infrastructures: IdP/SP proxy, User and group management, Form management, Self-service and external Attribute provider management.

The IdP/SP proxy component of B2ACCESS is derived from "Distributed Hub and Spoke Model", which is based on principal of decoupling the identity (IdP) and service provider (SP) federations by introducing a proxy in-between. The B2ACCESS proxy IdP/SP acts as an intermediary and significantly reduces the complexity of trust and authentication management between the two entities (IdPs and SPs). Alongside simplified trust management, B2ACCESS plays an important role

in connecting the IdPs and SPs that may not rely on similar authentication protocols (for example, SAML based IdP and OIDC based SP). In that case, B2ACCESS performs user authentication from her IdP in a protocol agnostic fashion and transparently generates the types of credentials based on the SP requirements (for example, access token for the OIDC SP or SAML assertion for the SAML SP).

The user and group management is one of the core functions of B2ACCESS and managed through its versatile administrator Web GUI. It enables the B2ACCESS administrators to manage the groups and the users therein. The user provisioning can be automatic (through user registration form) or manual (under the administrator Web user interface) by dragging the users from one group to the other. The groups are hierarchically organised, consequently the privileges assigned to the users are inherited from parent to the children groups. In addition to that, each group (and its users) are managed independently by its administrators. Furthermore, B2ACCESS uses MVFLEX Expression Language (MVEL) to let administrators define specific rules for the groups. This is an advanced feature and has been very useful in creating dynamic user attributes in (sub-)groups.

The form management sub-system enables B2ACCESS administrators to create user registration forms for the new users. The form is invoked upon successful user authentication. It is very common that an infrastructure relying on B2ACCESS has changed its policy and requires its 'existing' users to provide new attribute(s); in that case B2ACCESS provides enquiry forms to extract that additional information from the users. There is also Invitation form to register a specific set of users, the form is therefore not publicly available.

B2ACCESS provides a self-service user homepage (separate from administrator Web GUI) where infrastructure users can manage their profile containing all their information/attributes, additionally they can also update their credentials under the home page.

Finally, B2ACCESS implements User Import Management (UIM) sub-system, which enables importing the user attributes from external attribute providers/authorities in a configurable manner. Currently SAML, LDAP and OIDC based external attribute providers are supported.

2.1.2.2 Maintenance plan

The operation of the EUDAT B2ACCESS service involves technological and security upgrades of the underlying framework and software stack in order to take advantage of robustness and the continuous optimisation to ensure the uninterruptible operation of the service. This activity will also be responsible for gathering new features requirements to improve the robustness, usability and interoperability of the service.

2.1.3 Perun

2.1.3.1 High-Level Service Description

Perun [R9] is an identity and access management system (IAM) designed to build strong identity and authorisation layer on top of an existing infrastructure. Perun is developed by CESNET and Masaryk University and the source code is publicly available on GitHub [R10]. It offers complete support for VO management and whole user life-cycle form enrolment to suspension/leaving the

VO. Users and groups can be managed directly in the tool by the VO manager or by users themselves as a part of registration flow, alternatively the users can be synchronised from existing external source like LDAP, VOMS or SQL database. Combination of all mentioned approaches is also possible.

Perun doesn't create or store any users' credentials, it's designed to work with existing identities like identities from eduGAIN, social accounts, digital certificates or even local accounts stored in for example LDAP server. Account linking is fully supported in an intuitive way which enables end users to manipulate with their identities. Identities together with personal attributes are considered private and sensitive data, therefore they cannot be manipulated with by anyone but the user itself.

Perun have a capability to register facilities, which represent anything from single machines, clusters, storage, elements to even software licenses in various infrastructure sizes (from managing single access to software license to creating accounts in cloud-like environment with thousands CPUs). Consequently, the facility manager can provide a resource to any VO, by which he/she defines roles, access rights and rules how the VO can utilize the facility. VO manager then decides which users within the VO will be allowed to use the resource. That enables facility managers who provides some services of infrastructure to negotiate terms on per VO basis and the managing of individuals uses are delegated to VO level.

All the authorizations' settings managed in Perun can be provisioned to other systems. Preferred way is to use the push models, where Perun provisions configuration to the service using standardized mechanisms customizable connectors. Alternatively, the services can obtain data from Perun using API or querying some auxiliary services like LDAP or SAML2 attribute authority. Push model eliminates online dependency which increases fault tolerance. Auxiliary services do not have these benefits; however they are not integrated into Perun, but deployed as standalone components, which enables to easily operate them in high availability configuration.

Within EOSC-hub, Perun is currently being used to manage VOs and users' access to Federated Cloud [R11] resources.

2.1.3.2 Maintenance plan

Perun is operated in several instances where one of them is utilized by EOSC-hub. CESNET is responsible for operating this instance and also for support. All critical components (e.g. ones that are part of an authentication of users) are deployed in high-availability configuration. The non-critical ones are designed and deployed in a way that they can automatically recover from possible outage. All components are monitored by Nagios server and the end user have option to easily report any problem they might encounter directly to the operations team.

Main development team is a combination of employees from CESNET and Masaryk University in Brno. Both these institutions use Perun for the internal purposes, therefore they are contributing manpower not only to the operations of instances of Perun but also to the development.

Main points for further development:

- Extend number of attributes available through LDAP interface, which serves as common integration point with IdP/SP proxies.
- Add support for more complex life-cycles of users within the VOs.
- Expand user documentation.

2.1.4 WATTS

2.1.4.1 High-Level Service Description

WaTTS (the INDIGO-DataCloud Token Translation Service) [R12] is a plugin-based Token Translation Service. Having modular architecture, WaTTs provides translation to various credentials (e.g. SSH keys, X.509, S3 access tokens, etc.), enabled by plugins. WaTTS accepts many OIDC providers (EGI-CheckIn, B2Access, Human Brain, Google and INDIGO-DataCloud IAM).

The primary usage of WaTTS within EOSC-hub will be via the rcauth-plugin. In this combination, WaTTS acts as one implementation of a so-called MasterPortal for the RCAUTH Online CA. After authenticating in WaTTS with any OIDC provider, a certificate may be requested. For this, the user is redirected to the RCauth CA WAYF. There, she may choose which home IdP to use for authenticating to RCauth. In this authentication step, the Level of Assurance (LoA) must be high enough to fulfil policy requirements. To date B2ACCESS, Checkin and a set of individual IdPs support these requirements.

Upon successful authentication at RCauth.eu, the users' browser is redirected back to WaTTS, where he is given a proxy of his certificate. The full certificate is stored inside a myProxy secure credential store. This store is not part of WaTTS, but a separate service (also shown in Figure 3). Further development of WaTTs, which aims to provide integration with VOMS, i.e. a VOMS proxy, is in progress.

The authentication diagram with WaTTS service is shown in Figure 3.

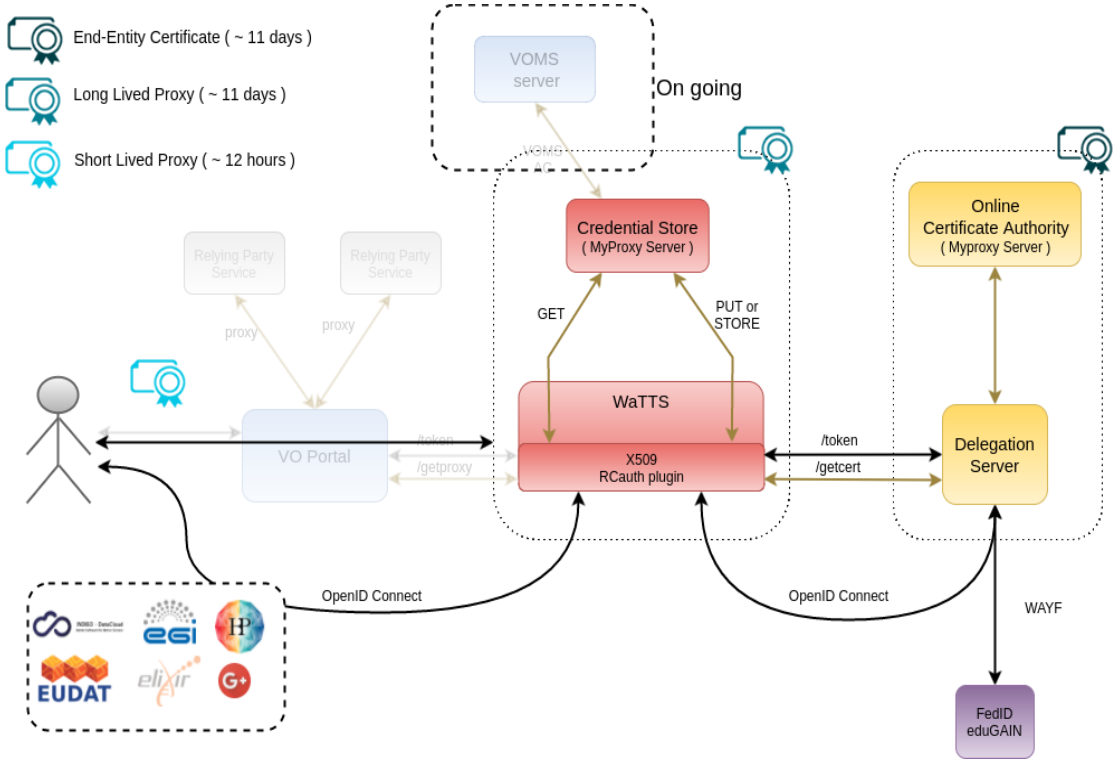

Figure 3 Authentication diagram with WaTTS service

The authentication flow is following:

1. User accesses (or is redirected to) the **WaTTS** page, and selects his OIDC provider to log in. This may be EGI, EUDAT, Google, Human Brain Project, INDIGO-DataCloud (further INDIGO) IAM or others.
2. User is redirected to the desired OIDC provider at which he authenticates. It's important to mention, that the LoA is not relevant at this point (since the user will have to authenticate to RCauth separately).
3. User is prompted to accept the release of information, and at the end, the information about the user is returned to WaTTS (i.e. LoA, issuer, name, mail, etc.)
4. Back at WaTTS, user selects the **X509_RCauth**.
5. If already authenticated with RCauth see point 7. Otherwise, user is notified that he will be redirected to authenticate at RCauth. This authentication step is only required once every 11 days, since that is the lifetime of the End Entity certificate.
6. User is redirected to **RCauthWAYF** and follows the respective login procedure. RCauth only accepts IdPs with an appropriate LoA.
7. User is presented with a page that allows him to download its X.509 proxy certificate.

Command-line / API Access:

REST interface

Once the user has his OIDC-Access token on the command line, he may also use the REST interface of WaTTS (we encourage the appropriate client **wattson** [R13]) to retrieve a grid proxy.

For command-line access with **wattson** the above general flow is different in the following steps:

4. User is also presented with his OIDC Access Token.

5a. If already authenticated, user may use **wattson** to follow the command-line flow. User's proxy will be placed to /tmp/x509up_uxxxx.

5b. If not authenticated, user is informed that he has to authenticate by pointing his browser to WaTTS.

SSH interface

In addition, a user may use an SSH public key to obtain the proxy certificate. For this, the ssh public key has to be uploaded to WaTTS, a WaTTS plugin for this is available.

For command-line proxy retrieval with **ssh** the above general flow is different in the following steps:

7. Upload ssh public key to WaTTS

8. SSH to the pickup node (e.g. watts-pickup.) The grid-proxy is returned from the ssh session.

2.1.4.2 2.1.4.2 Maintaince plan

WaTTS service is hosted by KIT. In EOSC-hub, the plan is to extend WaTTS installations with implementation of high-availability feature.

For implementation we will evaluate the optimal strategy together with the related components, i.e. the MasterPortal and RCAuth.

For operation WaTTS installations in a production quality, the plan is:

- Extend the current Lifescience-AAI pilot installation by an Installation for EOSC at KIT.
- Support installations outside KIT (Candidates: GRNET, NIKHEF, INFN. To be discussed and decided).
- Include all installations into one high-available instance.

2.1.5 MasterPortal

2.1.5.1 High-Level Service Description

The MasterPortal, developed by Nikhef in the AARC project [R14] as part of the pilot that resulted in the RCauth online CA, acts as a caching intermediary service between end-services and that RCauth Online CA. It is designed to provide end-services such as science gateways with proxy certificates for their users. The need for an intermediary is two-fold: it is necessary for a scalable

trust model based on a single online CA for Europe, and secondly, it hides all the complexity of certificate and key handling for the end-services. In this pure-webflow, the MasterPortal is seen by the end-users only as a 'redirect' between the end-service and the RCauth.eu online CA: authentication at the MasterPortal is effectively outsourced to that CA [R15]. Since the online CA accepts a wide range of IdPs, including any IdP in eduGAIN that publishes Research & Scholarship (R&S) Attribute Bundle and asserts Security Incident Response Trust Framework for Federated Identity (Sirtfi) [R16], the MasterPortal can be instructed by its client end-services to request a specific IdP during the login.

In addition to providing portals with proxy certificates, it can also provide end-users with a means to use ssh key authentication to retrieve proxy certificates on the commandline. For this functionality the MasterPortal allows them to upload a ssh public key directly, but it can also reuse those keys uploaded to e.g. a COmanage instance, this latter option -- using COmanage -- is currently enabled on the testing instance running in EGI.

2.1.5.2 Maintenance Plan

Several MasterPortals are currently running in production, one of which by EGI connected to Check-in service as was shown in Figure 1, while others are provided to research infrastructures. They are now independent instances, which is not sufficient to allow service delivery with high-availability guarantees. In addition, the MasterPortal provides both complementary as well as overlapping functionality compared to WaTTS. However, the specific role of the MasterPortal as caching and proxying intermediary is unique and must be evolved, to separately support high-availability. The maintenance plan comprises:

- Addition of high-availability options for data centre deployments.
- Alignment of capabilities with WaTTS, to ensure a consistent offering to communities (it must either be clear which solution is appropriate, and the solutions must be gapless).
- Capability to use a distributed back-end on-line RCauth.eu CA service.
- Provision of deployment tooling for (consortia of) service operators.
- Work towards the minimisation of the code base that has to be independently maintained.

2.1.6 RCauth - Online CA

2.1.6.1 RCauth - Online CA High-Level Service Description

The RCauth.eu Online CA [R17] is developed by Nikhef under the Authentication and (AARC) project [R14] as one of its pilots, based on a modified code-base of the CILogon online CA run in the US. It has been approved by the IGTF and is accredited under the DOGWOOD assurance profile [R18]. The current installation is running in a secure part of the Nikhef datacenter. The setup is fully production from a security and policy point of view, but the hardware is currently running as a Proof of Concept.

The online CA uses a filtering WAYF to connect to eduGAIN, accepting IdPs that assert the Research and Scholarship Entity Category (R&S) and Sirtfi [R16]. In addition, several other IdPs are accepted, with which an individual exchange of metadata is performed. This is in particular useful

for IdP/SP proxies such as EGI Check-in and the ELIXIR AAI and for IdPs that for example cannot assert R&S into eduGAIN. The WAYF currently only has support for SAML-based IdPs. Support for OIDC providers in planned.

Since the WAYF connects to a wide-range of IdPs, it is possible to pass an IdP hint to it during the login flow, which (if available) will not present the user with a selection menu, but directly forward to this IdP. This is useful for end-services wishing to always send their users to a fixed IdP(proxy).

2.1.6.2 Maintenance plan

The deployment model for RCauth.eu foresees multiple (3) operating partners that will be distributed across Europe, over potentially large distances (where network latency can be up to 50 ms, which poses additional challenges for the requisite state synchronisation between the issuing instances). The software will be adapted to support high-availability using several strategies: a coordinated back-end state mechanism to ensure uniqueness and traceability; mechanisms to distribute and align identifier assignment; and a fail-over/redundancy mechanism. In addition, collection of IT security and service management indicators will be added in support of such distributed operations.

The service deployment team, which will include both EOSC-hub and other European and national partners, will further guide the development and maintenance plan.

2.2 Integration plan

This section presents the integration plan for delivering an EOSC-hub AAI that enables seamless access to research data and services provided across nations and disciplines. The integrated EOSC-hub AAI will build on existing AAI solutions that follow the architectural and policy recommendations defined in the AARC project. Solutions from EGI, EUDAT and INDIGO that have successfully delivered a portfolio of operational services (Technology Readiness Levels above TRL 7) in this field over the last years are the initial basis of the integrated EOSC-hub AAI. These AAI solutions connect to eduGAIN as service providers but act as identity providers from the services point of view, thereby allowing users to use their credentials from their home organisations. Compliance with policy frameworks such as the REFEDS Research and Scholarship entity category and Sirtfi facilitates sufficient attribute release, as well as operational security, incident response, and traceability. Complementary to this, users without an account on a federated institutional Identity Provider are still able to use social media or other external authentication providers for accessing services. Thus, access can be expanded outside the traditional user base, opening services to all user groups including researchers, people in higher-education, and members of business organisations.

Research communities can leverage the EOSC-hub AAI services for managing their users and their respective roles and other authorisation-related information. At the same time, the adoption of standards and open technologies, including SAML 2.0, OpenID Connect, OAuth 2.0 and X.509v3, facilitates interoperability and integration with the existing AAIs of other e-Infrastructures and research communities.

A high-level view of the integrated EOSC-hub AAI is provided in Figure 4. The EOSC-hub AAI comprises different SP-IdP-Proxy services each of which acts as a bridge between the community-managed proxies managing the researchers' identity and the generic EOSC-hub e-infra services. Each community proxy in turn serves as a bridge between external identity providers and the proxies to the e-infrastructure services. Thus, researchers can sign in with their community identity via their Research Community AAI, e.g. B2ACCESS, Check-in, IAM, as well as GÉANT eduTEAMS as a result of the collaboration agreement with the GEANT4-2 project. Community-specific services are connected to a single Research Community AAI while e-Infra services are connected to a single e-infra AAI service gateway. Lastly, generic services (e.g. RCauth.eu Online CA) may be connected to more than one Research Community AAI proxies.

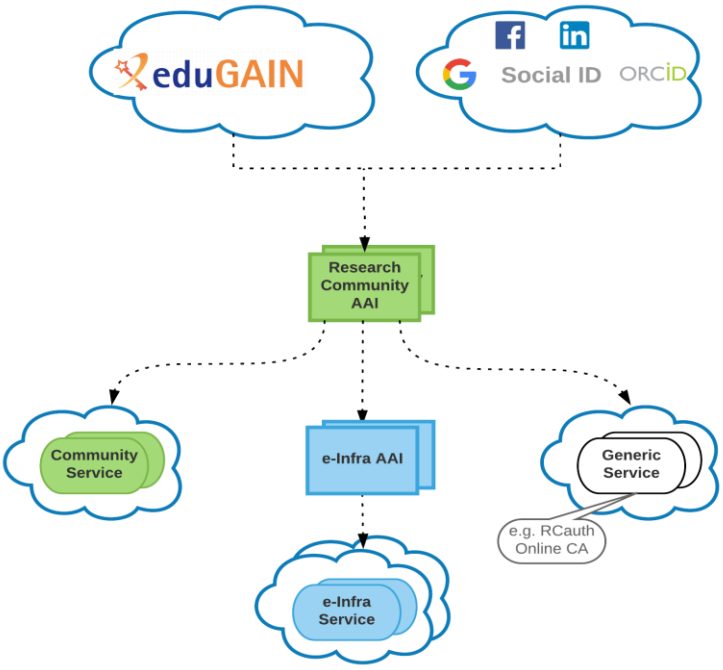

Figure 4 Integrated EOSC-hub AAI

AAI services provided by e-Infrastructures can act as both community proxy, for communities willing to leverage on e-Infrastructures to operate their own AAI, and e-Infra AAI.

The architecture above allows researchers to use their existing credentials from the Identity Provider of their Home Organisation to register once with their Community AAI and then to be able to access any of the services behind the different e-Infrastructure SP Proxies.

2.2.1 B2ACCESS - Check-in interconnection

This activity will enable the interconnection of EGI Check-in with EUDAT B2ACCESS to allow for the uniform authentication and access control of users in different community and service domains. The interconnection requires the harmonisation of user attributes, the alignment of the levels of

assurance, and the consistent representation of group membership and role information. Specifically, the following steps have been identified:

- B2ACCESS unique user identifiers will be expressed as eduPersonUniqueIds.
- Attributes expressing Check-in/B2ACCESS VO/group membership and role information will follow AARC guidelines [R19].
- Assurance information, including identifier uniqueness and the identity, authentication and attribute assurance, will be expressed based on the REFEDS Assurance Framework [R20] and AARC guidelines [R21], [R22].
- Users' affiliation within their home organisation and infrastructure will be expressed based on AARC guidelines.
- B2ACCESS/Check-in Acceptable Use Policies (AUPs) will be aligned based on AARC policy recommendations.
- SAML attributes for which there is no standard OpenID Connect claim will follow the mapping rules [R23] recommended by REFEDS/AARC.
- Check-in will be registered as an Identity Provider in B2ACCESS.
- B2ACCESS will be registered as an Identity Provider in Check-in.
- The IdP discovery and other user interfaces will be further developed to allow customisation based on the individual needs of different types of services and communities.

2.2.2 (De-)provisioning and continuous update of user account information

Many services require accounts to be provisioned before the users access the service. Even for services, which can provision accounts at the time of the first user access, the account information needs to be kept up to date (e.g. VO/groups/roles) and the services needs to be notified to deprovision the accounts when they become inactive.

This activity will enable services that require it, to be notified for account provisioning, deprovisioning and updates using standards-based solutions:

- VOMS (de-)provisioning (for users without a personal certificate or users whose VO is not managed by VOMS): for those services that require VOMS proxy certificates, Check-in needs to be able translate SAML assertions or OIDC claims to VOMS proxy extensions. Having this capability, users without a personal certificate or users whose VO is not managed by VOMS, will be able to use certificate-based services.
- Check-in needs to automatically (de-)provision users and groups to any application or identity store that is fronted by a web service with the interface defined in the System for Cross-Domain Identity Management (SCIM) 2.0 protocol specification.

2.2.3 WATTS - MasterPortal

As already stated, the primary usage of WaTTS within EOSC-hub will be via the rcauth-plugin. In this combination, WaTTS acts as one implementation of a so-called MasterPortal for the RCAUTH Online CA. Thus WaTTS will be integrated into the EOSC-hub environment to support the translation of federated credentials into tokens required to support access to non-web services, e.g. those relying on X509 certificates.

Development plans for both implementations:

- Fault tolerant High Availability (HA) deployment, so that the outage of one instance does not affect the operation of the infrastructure
- Closer Integration between WaTTS and the MasterPortal software of Nikhef. It needs to be investigated whether the two implementations can be merged, since they provide slightly different functionality: WaTTS provides a user-friendly web interface, the MasterPortal also functions as an OIDC provider.
- Additional plugins (SSH, SSH-CA, S3, ...) may be integrated into WaTTS to satisfy emerging users' requirements. For example WaTTS can manage SSH-Keys for access to VMs of a potential EOSC IAAS infrastructure.

2.2.4 RCauth - Online CA

Options to support non-web services, which traditionally relied on X509 certificates, are based around the concept of online authorities with attached credential stores, such as the RCauth.eu Online CA. Such techniques allow science gateways to obtain credentials on behalf of the end-user that can be used to directly authenticate to services.

RCauth currently does not support a HA installation. It is being investigated how this can be optimally done for each component of the RCauth service which is architecturally centralised, e.g. front end, database, key repository, etc. Obviously, components of RCauth that don't need high availability will not be considered as part of this process. Code which is currently run by NIKHEF will be redeployed and reconfigured at the target hosting sites (namely, STFC and GRNET) in a HA deployment. It may also be necessary to package code, possibly for different operating systems.

The technical changes need to support monitoring of operational and service parameters by the operational partners that run the actual service.

3 Marketplace and Order Management tools

3.1 Overview of services

3.1.1 Marketplace

3.1.1.1 High-Level Service Description

EOSC-hub Marketplace (MP) is a user-facing platform where production 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 user with e-infrastructures.

Functionalities offered by the Marketplace are:

1. Service catalogue management: Creation, publishing and updating the services in the MP backoffice. All services are classified and presented as part of a 3-level (service category, service, service option) hierarchy implemented to enhance user experience in the system.
2. Authentication: The login procedure including the user registration during the first access.
3. Discover and order services: Finding and ordering services within the Marketplace. The users can customise their orders selecting available service options and attributes.
4. Check-Out: Submitting a service order together with a set of information to profile it.
5. Order handling: based on given information and procedures behind it a set of order requests is created, ready to be handled by the Operational Team.
6. SLA management: accepted user's order results in creating corporate or custom SLA for the user, available on the user dashboard.

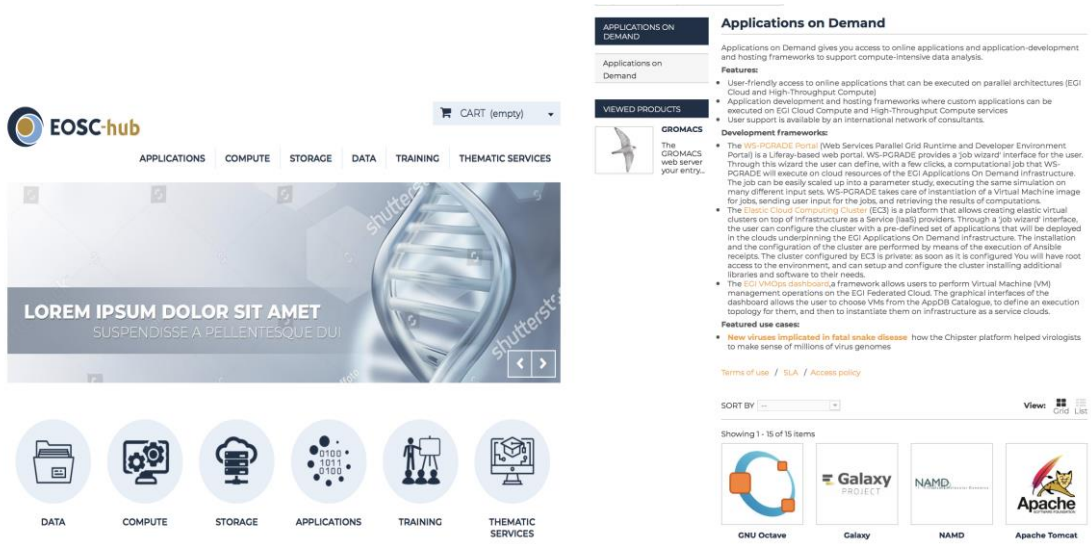

Figure 5 Service discovery in the Marketplace

The technological solution behind the Marketplace was delivered as a result of EGI Engage project by ACK Cyfronet AGH and the first deployed instance is currently supporting EGI infrastructure and its service catalogue (marketplace.egi.eu).

Current EOSC-hub Marketplace environment [R24] consists of following elements:

- Marketplace User Graphical Interface
- Marketplace Order Management component.

High level architecture of the Marketplace and its environment is shown in Figure 6.

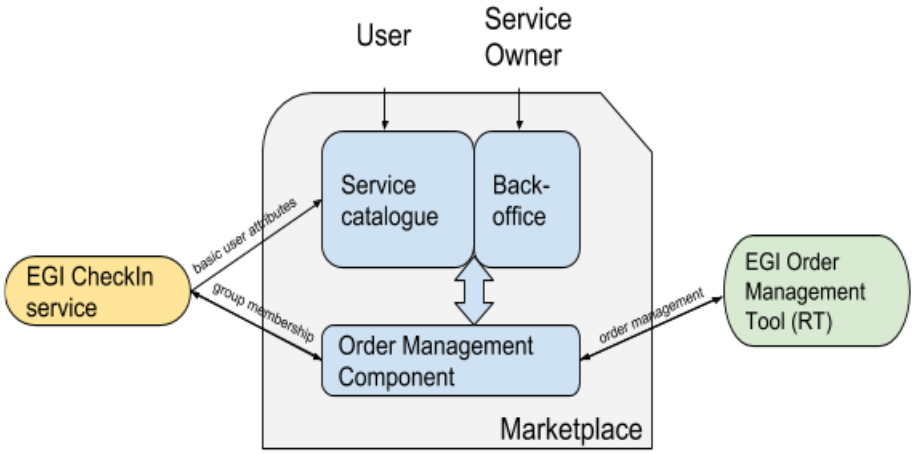

Figure 6 High Level Architecture of the Marketplace and Marketplace environment

User Graphical Interface - service catalogue:

Component responsible for direct User-System and Service Owner - System interaction. This component is a PrestaShop [R25] instance (PHP application using a MySQL database) enriched with the following plugins to extend its functionalities:

- Additional Product Attributes/Custom Product Fields Module: Allows to properly describe and introduce all order-relevant attributes to services and its' options.
- Custom Checkout and Customer and Address Fields manager Module: Needed to implement customer and service order profiles relevant in order handling.
- Dynamic Product Price Module: Allows to define dynamic prices based on the values that customers defined for the service options. Supports Pay-for-Use model that can be included in the platform
- Google Accounts login-in module for PrestaShop: Needed in order to extend the login functionality of PrestaShop, allows CheckIn service integration.

Additional customisations in PrestaShop were needed to implement the authentication and user enrolment, and the Check-Out workflows. In particular, to retrieve customer information from the Check-in service, to prevent the service order submission before the customer profile is completed

and to profile the service orders. Minor changes were also done to adjust the service options, the service list in the cart and the email templates. All the changes were applied to both the PrestaShop basic code and the extra modules listed above.

MINOS - service order management component:

Java based application closely integrated with PrestaShop database. It was developed to extend PrestaShop functionalities and introduce support for numerous workflows for order handling. It is the single point of integration with tools supporting the order handling activity and surrounding business processes. It implements some features for the order handling management and, in particular, the semi-automatic workflow to manage the Application on Demand service (AoDs) [R26] orders. Basing on external configuration files (yml format), it allows to define rules for request orders creation in EGI RT, attribute management in RT tickets (which order attributes should be mapped and how to TR ticket attributes), managing thresholds for semi-automatic resource provisioning for Application on Demand service, passing the information back and forth between PrestaShop and EGI Check-in service about users membership in AoDs group.

3.1.1.2 Maintenance plan

System configuration:

PRESTASHOP SERVER

Apache HTTP server version2.4.6 (CentOS)

PHP version 5.4.16

DATABASE INFORMATION

MySQL version 5.5.56-MariaDB

MINOS application

JAVA 8

Marketplace platform is available as a VM image consisting of mentioned components ready to be deployed and operational. Maintenance involves activities such as:

- code review
- components updates
- configurational adjustments

all to enhance platforms usability and operationality.

3.1.2 Service Portfolio Management Tool

3.1.2.1 High-Level Service Description

The Service Portfolio Management Tool (SPMT) is aimed at facilitating service management in IT service provision, including federated scenarios. SPMT presents a complete list of the services managed by a service provider; some of these services are visible to the customers, while others are internal. SPMT allows maintaining and managing the descriptions of services in the portfolio and it allows managing the transition from the portfolio to the catalogue.

The service management system has been designed to be compatible with the requirements for service portfolio management according to FitSM IT service management standards [27]. SPMT was developed by GRNET by gathering requirements from VI-SEEM [R28] and EUDAT2020. For EOSC-hub a production instance of SPMT is running at [R29], which is used to populate the EOSC-hub catalogue available at [R30]. The SPMT User Interface is shown in Figure 7.

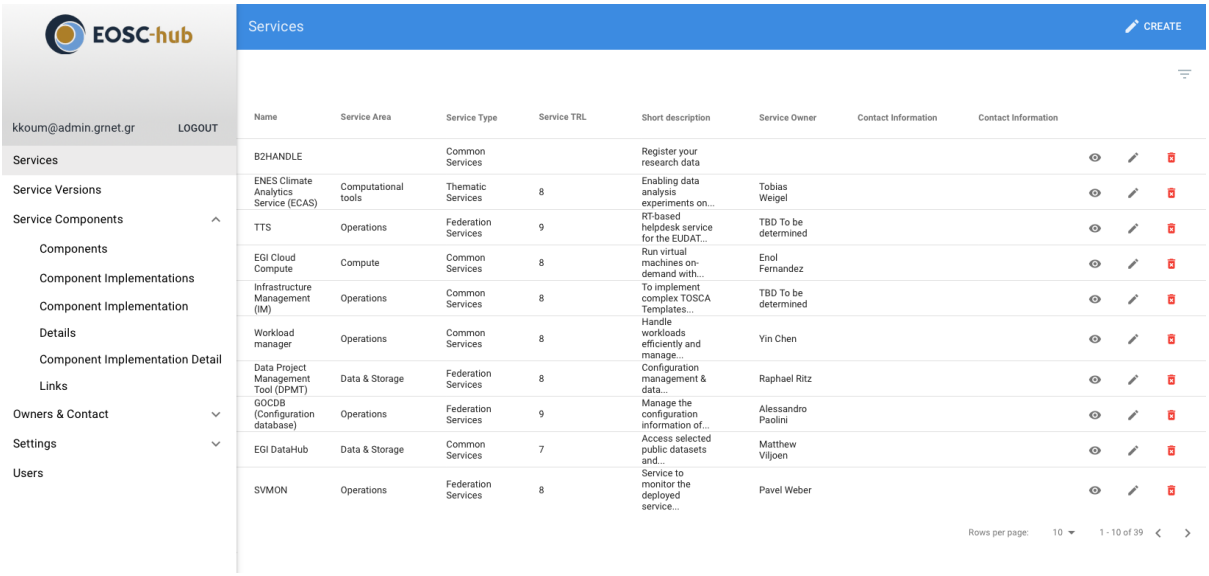

Figure 7 SPMT Write UI

A list of the main features and functionality of SPMT follows:

- The service catalogue contains only information about services that are offered to the potential customers. A subset of all the service information recorded in SPMT is published in the service catalogue for a subset of the services (e.g. production services) that are registered in the service portfolio.
- Each service can have multiple versions in the service portfolio. Each version can have a different readiness status i.e. “concept”, “under development”, “beta”, “production”, “retired” etc.
- The service portfolio is accessible via a RESTful API to accommodate 3rd party application building and creating custom views of the service catalogue according to the needs of the organization.
- Authentication: SPMT supports federated AAI handled via SAML.
- Authorization: SPMT supports role-based authorisation. The following categories are defined: admin, observer, service-owner.

A high level class diagram of SPMT data model is shown in Figure 8. It presents the slightly updated data model of the production version (SPMT version 1, released by February 2017 and used in the context of the EUDAT2020 project).

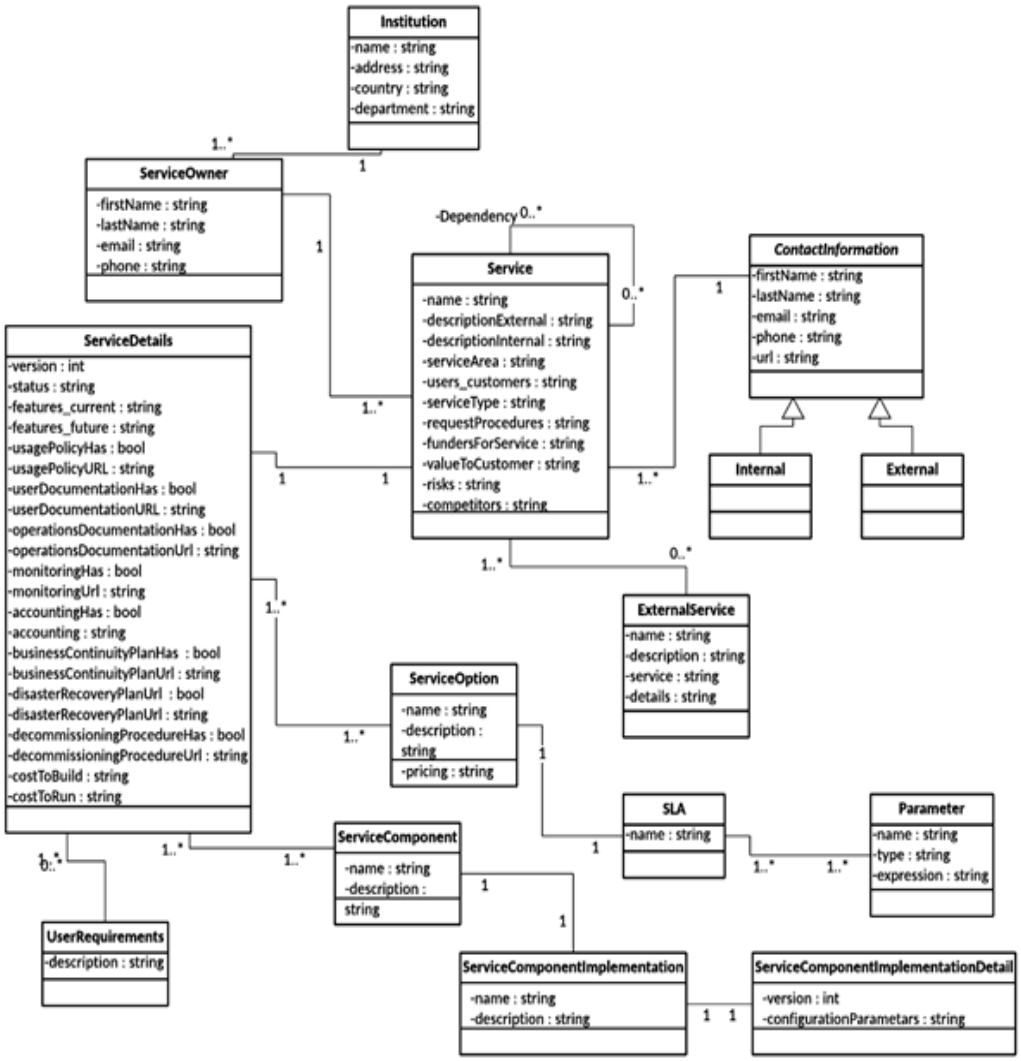

Figure 8 Data Model of the SPMT v1

3.1.2.2 Maintenance plan

We currently maintain two instances of SPMT: production instance at [R29] and development instance at [R31]. We use ansible scripts to automated deployment on VMs on GRNET Cloud Service ~Okeanos. In order to improve uptime, we roll out component features/updates/fixes initially in the devel instance to test and showcase the new features. After approval, they are rolled out to the production version.

Maintenance involves activities such as:

- code review

- components updates
- configurational adjustments
- regular security update

3.2 Integration Plan

3.2.1 Marketplace - SPMT

This integration between Marketplace and SPMT is crucial from the project point of view. The Service Portfolio Management Tool (SPMT) is a tool dedicated to provide a full list of services in the portfolio and allow managing service descriptions according to the service management guidelines of FitSM. It is meant to be a single point of entry for service providers to introduce their services and to reflect the life cycle of the service. When a service becomes production, it has to be published in the Marketplace, to make it available to the users. Marketplace must implement service portfolio related service data schema, whereas SPMT has to become aware of the hierarchy of the services and their internal organisation. To support proper access management in scope of service entry in the Marketplace for the service owners, SPMT needs to store service owner's persistent ID obtained from AAI system, which is used to manage user rights and assign corresponding role in the Marketplace.

To allow efficiency, a notification component needs to be delivered, which will be triggered on events in SPMT and give push notification to the Marketplace (MP) when a change in service entry will take place. MP must become aware of the service life cycle implemented in the SPMT, in this way it can properly filter services ready to be shown to the users (e.g. production services). Finally, as SPMT will not hold all the information relevant from the order point of view (e.g. service options and attributes to customise orders, they are not part of service portfolio entry), the two tools need to develop a common interface for service owners (connected to both SPMT's and MP's databases) which will allow them to enter all service attributes required by the system in one place.

3.2.2 Marketplace integration with operational tools

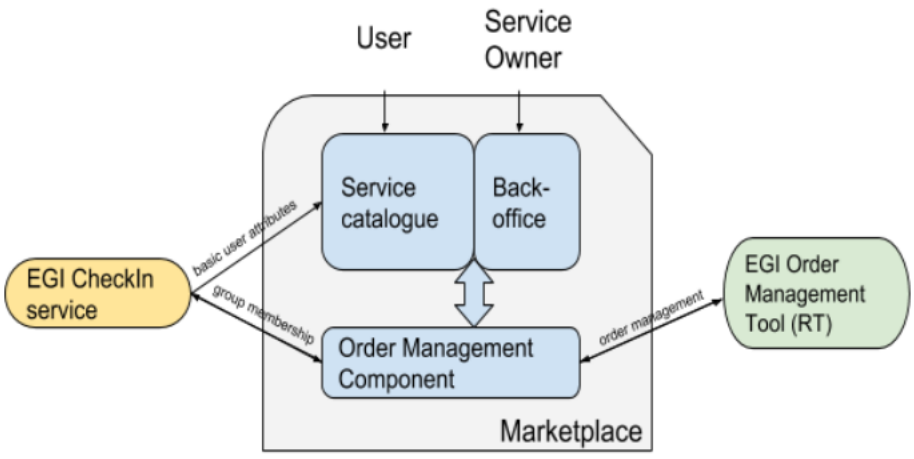

Figure 9 Integration of Marketplace with operational services

In order to properly support the user in the EOSC-hub environment and enable him easy service and resource ordering, a number of integrations is required. Apart from integration of Marketplace with SPMT to collect services with their business value attributes and Marketplace integration with AAI, which ensures crucial user attributes, Marketplace needs to enhance connections facilitating order management and complex order preparation which is an input for it.
To facilitate expanded and more accurate description of user needs, technical information about the service is needed. A source for technical data are EOSC-hub CMDBs (GOCDB and DPMT are shown in Figure 9) and due to theirs heterogeneous architecture each of them will integrate with a common API, passing attributes relevant from the Marketplace point of view in a format expected from the Marketplace. It will be a one -way integration, Marketplace will not be writing to any of the tools, only pulling data. Precise requirements are still to be discussed and will depend on the MP data schema which is now under construction.
Order management will be realised through the integration of Marketplace with Operations Portal and with JIRA ticketing system as the intermediary component. The simplified data flow is shown in Figure 9. As now, Marketplace will be responsible for ticket creation logic and storing relevant information about the order in JIRA, where the order management will take place. After an order is processed it will be captured by Operational Portal that will manage it until an SLA is agreed. At the same time, the Operations Portal will provide information about the order status to the Marketplace that will show it to the user along with the underpinned SLA(s).

3.2.3 Marketplace - Check-in service Integration

The current MP - Check-in integration will require further development to improve user attributes in the scope of order management. At the moment, information retrieved from the Check In includes user persistent ID, name and surname and user's email, all of them essential to process the order. Improved integration requires check point on the Check-in side and user's intervention to ensure existence of attributes crucial from the MP point of view. The minimum set of attributes identified, as follows:

- name
- surname
- email
- affiliation type (user profile)
- group memberships

The set of attributes that Check-in provides to the Marketplace could be enriched to satisfy emerging requirements.

4 Integrated Business and Operations Support Systems

4.1 Overview of services

4.1.1 Operations Portal

4.1.1.1 High-Level Service Description

The Operations Portal provides VO management functions and other capabilities which support the daily operations of the EGI infrastructure. It is a central portal for the operations community that offers a bundle of different capabilities, such as the broadcast tool, VO management facilities, a security dashboard and an operations dashboard that is used to display information about failing monitoring probes and to open tickets to the resource centres affected. It is fully interfaced with different helpdesks and the monitoring system through messaging. It is a critical component as it is used by all EGI Operations Centres to provide support to the respective resource centres. The Operations Portal provides tools supporting the daily running of operations of the entire infrastructure: Infrastructure oversight, security operations, VO management, broadcast, availability reporting.

More recently a new module called VAPOR has been added to the portal. VAPOR is a set of tools, scripts, web services collecting information about computing and storage resources and their statuses. VAPOR provides different web interfaces and API to consult this information.

- The aim of this set of tools is:
  - Easily identify the available resources for a given VOs
  - Check the status of these resources
  - Access in an easy way to the information published by sites
  - Provide summaries/reports for sites, Operations Center or communities
  - Provide unified and simplified views of GLUE2 information

The Operations Portal has been designed as an integration platform, allowing for strong interaction among existing tools with similar scope, but also filling up gaps wherever functionality has been lacking. The displayed information is retrieved from several distributed static and dynamic sources – databases, Grid Information System, web services, etc. – and gathered within the portal.

The architecture of the portal is composed of three modules:

- A database – to store information related to the users or the VO;
- A web module – graphical user interface – which is currently integrated into the Symfony

framework;

- A Data Aggregation and Unification Service named Lavoisier.

Lavoisier is the component used to store, consolidate and “feed” data into the web application.

The global information from the primary and heterogeneous data sources (e.g. VOMSES, GOCDB, NAGIOS, AppDB, ARGO, etc.) is retrieved by means of the use of the different plug-ins. The collected information is structured and organized within configuration files in Lavoisier and, finally, made available to the web application without the need for any further computations. This modular architecture is conceived to add easily new data source in this model and use the cached information, if a primary source is unavailable. The data sources are refreshed only as needed and only when an action has been triggered. In addition, it is very easy to add a new data source in this model, as depicted in Fig. 10 and Fig. 11. Nevertheless, two critical dependencies are remaining on the helpdesk systems: Global Grid User Support (GGUS) and Request Tracker for Incident Response (RTIR), as shown by red arrows in Fig. 10). These dependencies are due to the communication via web services between the Operations Portal and GGUS/RTIR for the creation or the update of tickets.

In case of disruptions of the GGUS or RT services, a part of the features of the Operations Portal will be affected: the creation and the update of tickets into the dashboards. For the rest of data sources, the cache mechanism of Lavoisier permits us to ensure the integrity of the application in case of failures of third parties providers.

For the VAPOR application, we use the same architecture with a dedicated instance of Lavoisier. Information is aggregated from several top BDII objects and from a monitoring tool based on JSAGA (JobMonitor) and local scripts in python and shell developed specifically to ease the VO support. VAPOR is fully integrated in the Operations Portal and is presented to the users as an additional feature available.

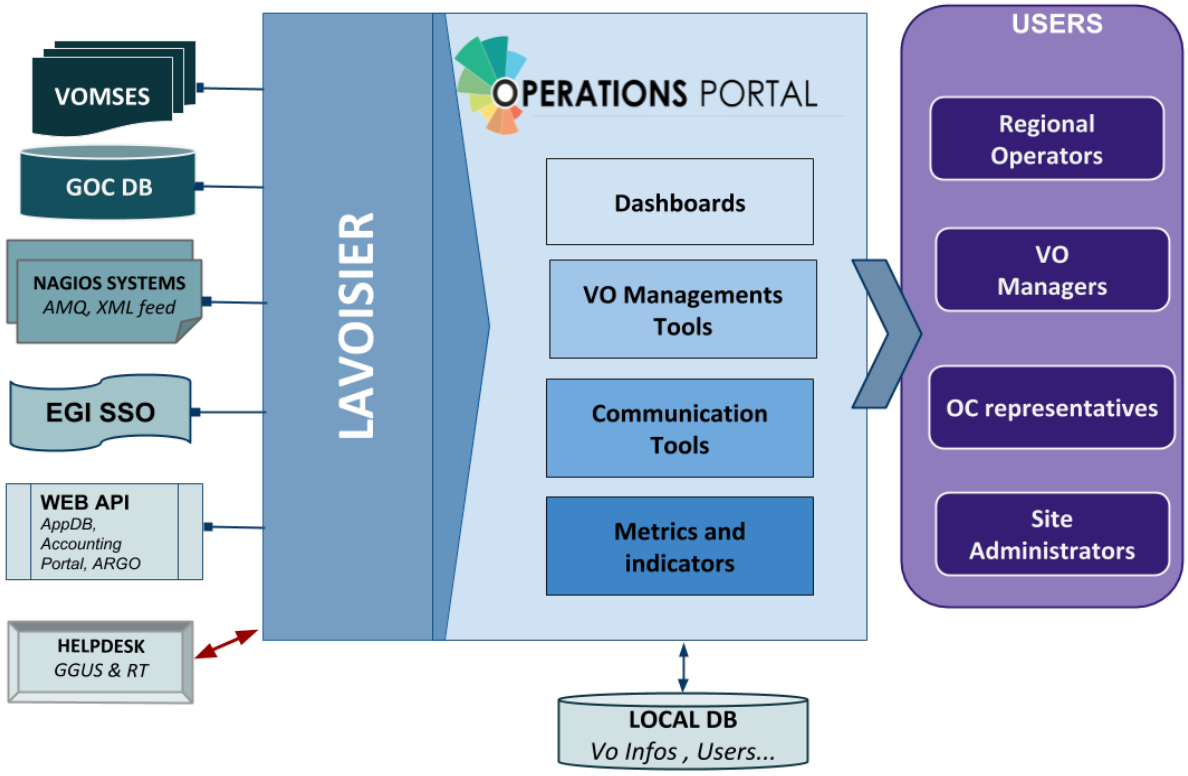

Figure 10 Operations portal

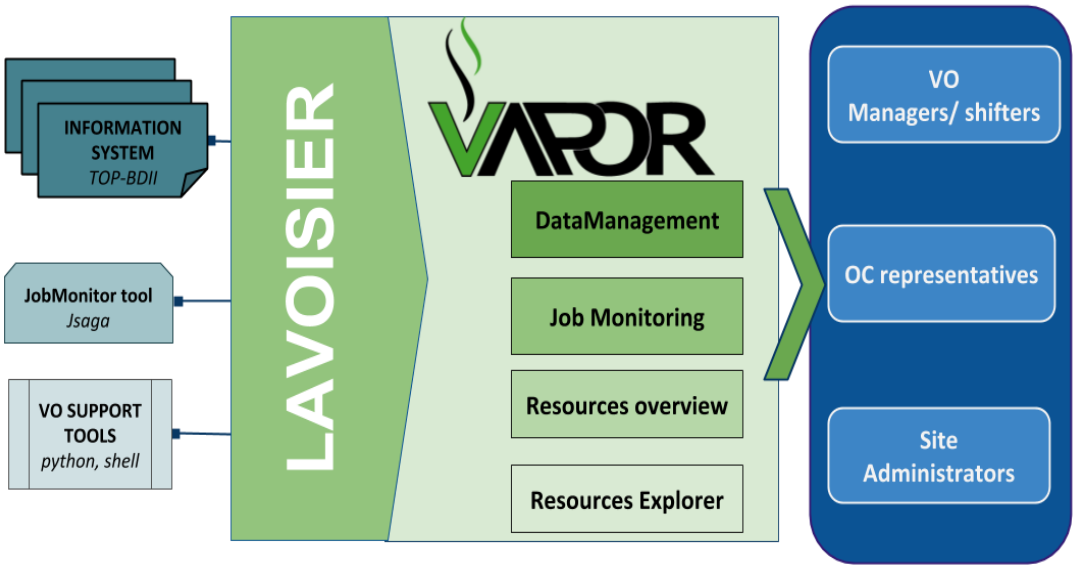

Figure 11 VAPOR application

4.1.1.2 Maintenance plan

Regular Maintenance

We perform regular upgrades on the software part to improve the performances of the applications.

It could be upgrade of the third parties libraries (javascript libraries, css framework or php framework). We work actively also on regular improvements on the data aggregation framework.

And we also put in place code reviews with the help of dedicated tools (SonarQube):

- to ensure the code quality
- to improve the efficiency of the code
- to ensure a good maintainability of the application

SLA/OLA management tool

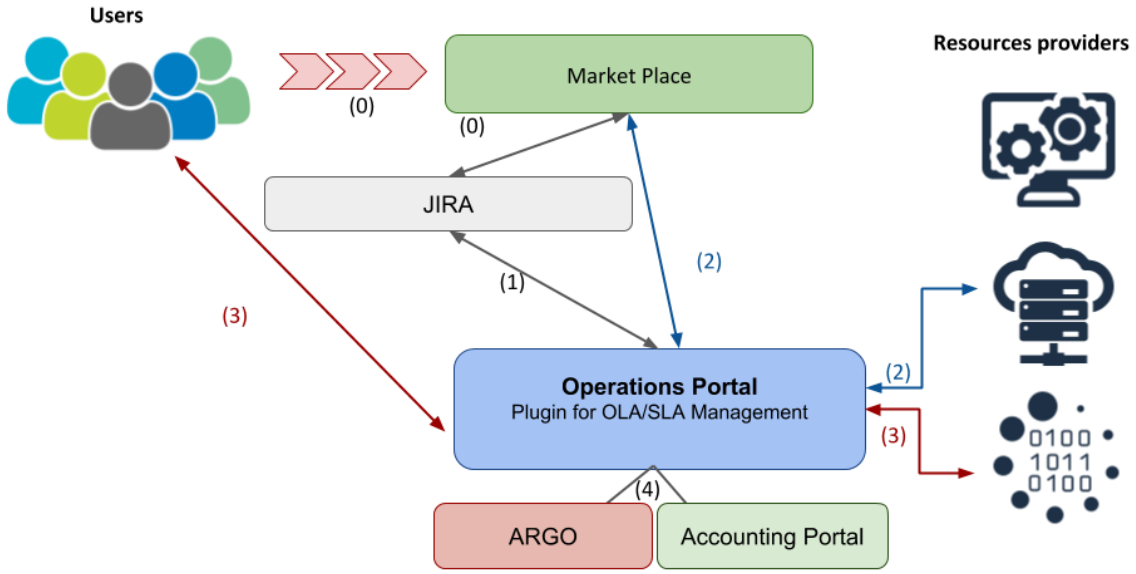

Figure 12 SLA/OLA management workflow

Workflow description

The Operations Portal will be mainly interfaced with the helpdesk instance (JIRA).
The workflow will be the following:

0. Individual users or communities will request resources through the Marketplace. These orders will generate tickets in JIRA containing the information about the nature of the request and the identity of the requestor.

1. The Operations Portal will be interfaced with JIRA and will present in a dashboard the different registered requests.

Depending from the nature of the request:

2. The Operations Portal could apply some automated orders either to the resource providers or to the Marketplace through the API.
3. Or the Operations Portal will start the negotiations between the resources providers and the users.
4. Then the Operations Portal will provide metrics with the integration of ARGO and the Accounting Portal about the usage / quality level of the resources.

SLA Management Interface

So depending from the nature of the request an SLA can be associated automatically or after a negotiation:

- Computing/Data/Storage services → Negotiation
- AoDs → default SLA
- Training infrastructure → default SLA
- FitSM trainings → default SLA

Information about SLA could be retrieved automatically via a REST API.

The main features of this plug-in will be oriented around the SLA management with 2 parts:

1. **Operator Dashboard**

This part will be restricted to the SLA Operators (Representatives of EGI, EUDAT and INDIGO-DataCloud) and will be used to manage the service requests :

- For SLA that is negotiated, keep a trace of the amount of allocated resources.
- Send automatic mail to the customer about the status of the negotiation.
- When the SLA is approved (information manually stored by the operator in the dashboard), a button to generate SLA and OLAs document is made available. Documents should be automatically stored on a document server.
- Provide templates for the SLA reports

2. **Service provider dashboard**

This part will be accessible for the resources providers and will allow to:

- See all my OLAs
- See open resource requests
- Approve OLA
- Download OLA and service report

- Provide my offer (low priority)

Reporting

Then the interface will provide also additional information:

- Resources usage with information coming from Accounting Portal
- Availability/Reliability of the resources with information coming from ARGO
- Any additional information about the quality of the service (e.g. survey results)

For each provider involved in SLA, the plugin should report metrics described previously and send them to the customer.

Once an order is approved, this plugin will act as orchestrator of the workflow to enable users to access the ordered services with the aim to automate as much as possible the configuration of the underlying infrastructure.

Dashboards

One historical module of the Operations Portal is the dashboard module in which a security dashboard and different operations dashboards provide information about failing monitoring probes and allow opening tickets to the affected resource centres. The dashboard also supports the central grid oversight activities. It is fully interfaced with the EGI Helpdesk and RTIR (helpdesk used by security team) and the monitoring system through messaging.

This module is based on old version of the Symfony framework and has been developed following the procedures and workflows set up during the EGEE project. The procedures have evolved and it should be reflected on the dashboard. We will provide a new dashboard more flexible and more customizable with the use of a recent version of Symfony.

Main objectives

- Replace the current dashboards (rod dashboard, security dashboard, VO dashboard, cod dashboard) by one dashboard:
    - To facilitate the maintenance
    - To upgrade Symfony version
    - To remove specificities and historical workflows
    - To propose a customisable service with users settings

4.1.2 GOCDB

4.1.2.1 High-Level Service Description

GOCDB is a central information repository providing a web portal interface for CRUD operations, and a REST API for data queries. It is a key tool for the configuration management of the EGI

Federation and it will become the configuration information service for the EOSC-hub behind the EOSC-hub marketplace.

It is a definitive information source where data is directly populated and managed in the system. Because GOCDB is a primary data-input source, the portal applies a range of business rules and data-validations to control input. It applies a comprehensive Role-based authorization model that enables different actions over different target resources. The Role model allows communities to manage their own resources where users with existing roles can approve or reject new role-requests.

It is intentionally designed to have no dependencies on other operational tools (other than the EGI Check-in service described below). For example, it does not query other systems to populate its core data model. The underling Oracle DB is hosted by the STFC DB Services Team with nightly tape backups. An additional failover instance is hosted at a second STFC site (Daresbury Labs). The failover instance is synchronized hourly against the production data.

The previous release, introduced a new dependency on the EGI Check-in service in order to provide federated access to GOCDB for users without client certificates. However, for users with certificates there continues to be no dependencies on other operational tools. Other than the extensions to the capability to the write API, this release brings no major alterations to the architecture.

4.1.2.2 Maintenance plan

Early in the project, GOCDB has been partially rearchitected to allow IPV6 access to the production instance. Over the course of the project, the failover instance will need to be moved to new infrastructure. The production instance architecture will be examined with the aim of increasing resilience and reliability. The functionality of the Write API will be expanded to meet evolving use cases. In order to meet the requirements of our new privacy policy (which has been reviewed in light of GDPR requirements), we will be consulting our user base on slight changes to access requirements to user contact details and making the associated changes.

4.1.3 Data Project Management Tool

4.1.3.1 High-Level Service Description

The Data Project Management Tool (DPMT) [R32] is a content management system that (a) supports the implementation of data management plans (or data project requests) during an enabling process and that (b) serves as a configuration management database. The DPMT registers service providers, their service and resource offers, the actual provided service instances and linked resources, the multi-entrant service components and data project requests and machine readable descriptions with specific states. The tool pulls the service descriptions from EUDAT's SPMT, is linked to the SVMON and renders the information from the EUDAT CDI accounting repository.

The code base of the DPMT, the information object types (content types) and the connectors to the GOCDB (for pulling information about sites, service endpoints and service groups) is openly accessible from [R33].

The Figure 13 below shows the view of the project panel of the DPMT. On the left side all the available projects are listed. The main part of the panel is a table of projects with some characteristics such as the project state, the group of service instances and the used storage space. The table rows can be filtered according to selection criteria and lexically sorted. The information can be exported in various formats (PDF, excel, json) and it can be pulled and further processed by any external client that is entitled via B2ACCESS.

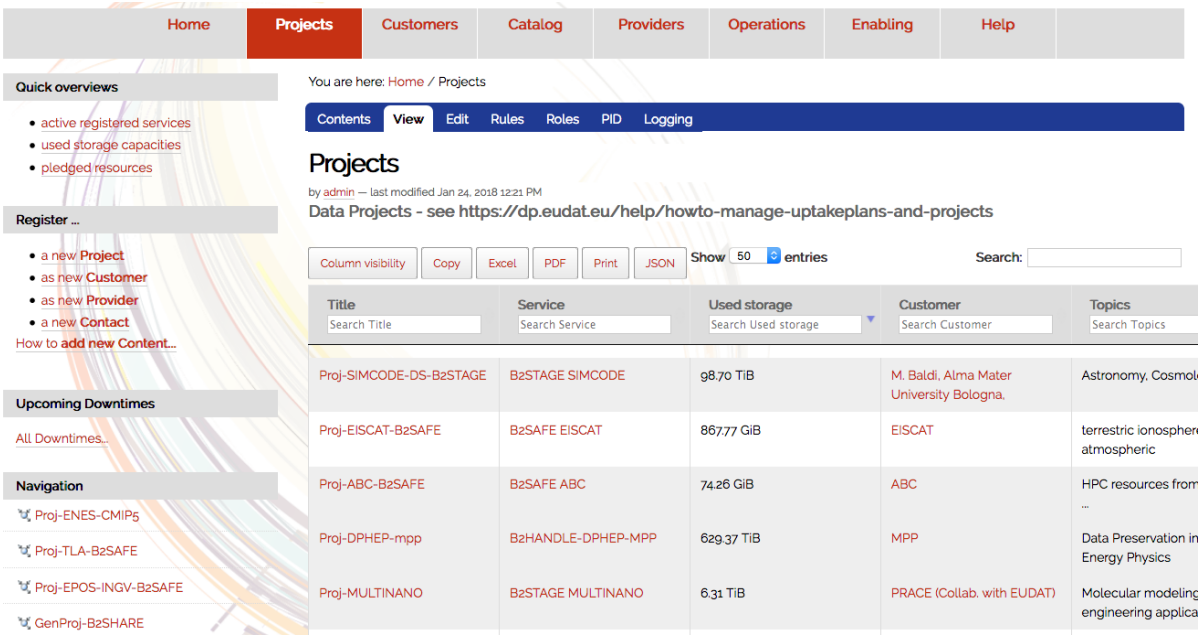

Figure 13 DPMT user interface

The DPMT allows the Providers to register provider information. A provider principal assigns roles to collaborators and manages them as far as it concerns the provider's administrative domain, service, provided services components or resources. The provider is in charge to validate the states of the services and he/she is checking the accounting status concerning the related services and particularly concerning the resources that are provided by the provider. The provider is checking for and replying to service (component) requests or resource requests. The provider is providing information about scheduled downtimes. The provider has to check whether the software is up-to-date or he should react on messages that are indicated that the software version will become deprecated. He/she adds or updates the information about the provided service components and resources.

The DPMT is the tool for the Service Enabler to manage information about the project and the enabled service in the course of the enabling process.

The DPMT provides status reports for *management bodies* (configuration, enabling status, offered vs. allocated vs used resources).

The configuration information can be exported in form of XML. This is useful for clients that are normally using the GOCDB Programmatic Interface [R34]. The DPMT has implemented functions of this programmatic interface that are relevant for the ARGO monitoring connector. These functions

include *get_site_list*, *get_site, get_service_types, get_service_group*, and *get_service*. Resource usage information is also available in StAR format.

4.1.3.2 Maintenance plan

The DPMT is hosted and operated by MPCDF and well integrated into its infrastructure. This includes proper maintenance of all underlying components (from the hardware via the operating system to the software libraries and frameworks used) as well as local monitoring and backup.

The DPMT application specifically will continue to be adapted to the needs of its users, including schema evolution where appropriate, additional or extended summary pages, as well as additional views in specific formats to improve interoperability with other components of the infrastructure (such as those realized using GOCDB or ARGO).

4.1.4 Data Management Planning Tool

4.1.4.1 High-Level Service Description

Research projects that intend to create or use data need to describe how the data will be managed during the data's lifetime. The data may live long after the project has completed. Funding agencies have developed guidelines following the Findable, Accessible, Interoperable and Reusable (FAIR) principles that projects should follow. The goal of the Data Management Planning (DMP) tool is to provide a way for researchers to create data management plans that can then be used by the project to manage their research data and by the funding agency and other stakeholders to mine the plan for information necessary for planning or other uses.

The DMP tool follows a distributed architecture and is a service consisting of a number of components that has been jointly conceptualized by EUDAT and OpenAIRE.

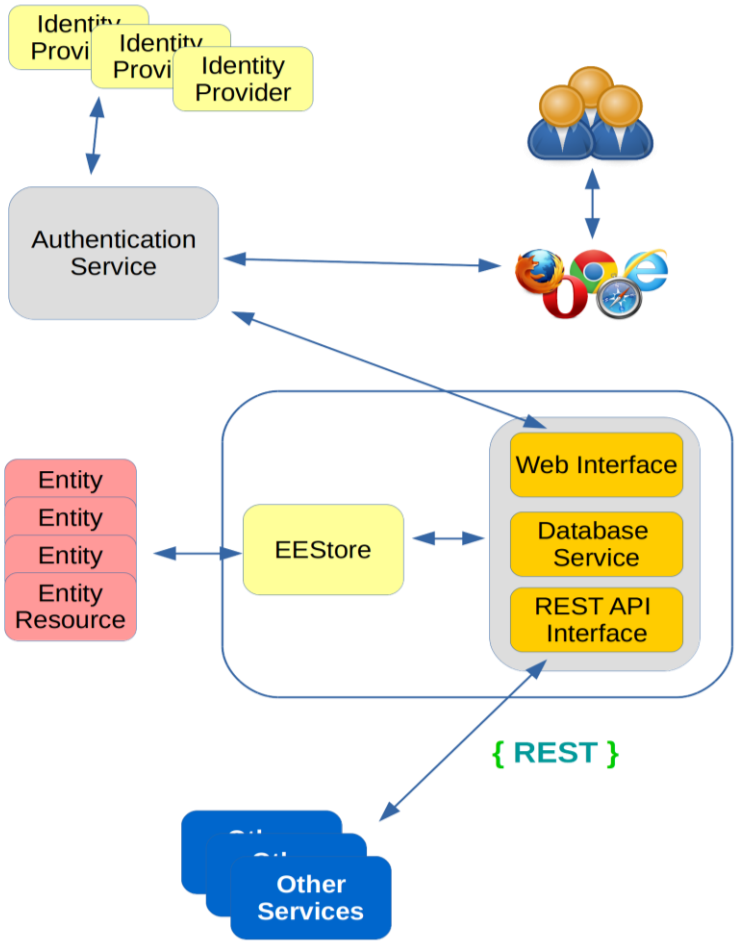

Figure 14 Illustration of the Data Management Planning Tool

A simplified illustration of the Data Management Planning Tool is shown in Figure 14. In EOSC-HUB we are currently working on the EEStore which is the uniform interface to the external registries such as the re3data and Zenodo etc. The DMP tool produces structured data management plans that can be consumed by other services through the REST API. The EEStore communicates via REST with other services.

The data model was developed from the viewpoint that the dataset is central to the data management plan. A core set of properties common to all datasets form the basis of the plan and properties that apply to categories of datasets (such as sensitive data) appear as dataset profiles that are in addition to the core properties. Some of the properties (such as metadata standard, or data repository) may be harvested from registries requiring the researcher to type in less information. This approach is particularly useful for plans for existing datasets where a significant portion of plan information can be extracted from external sources (e.g. information on the project, metadata, repository etc.). The model enables funding agencies to extract information from plans to facilitate funding and resource planning.

Researchers manage their Data Management Plans through the web interface and consumers of the plans can use REST APIs to extract plans.

The goal of the tool is to provide a flexible architecture that can accommodate services interested in consuming a plan or elements of a plan. The architecture can support many different types of data management plan template and the Horizon 2020 FAIR data management guidelines is the first template to be incorporated into the tool.

4.1.4.2 Maintenance plan

The DMP tool follows a distributed architecture. The DMP External Entity Referencing Service (eeStore in Fig 14.) is currently hosted on the UNINETT Sigma2 infrastructure in Norway and is integrated into the EUDAT infrastructure through B2ACCESS. The service will continue to be operated by Sigma2 and will be updated according to the needs of the developing DMP architecture. The DMP web interface and backend will be maintained by OpenAIRE-Advance project. Further services arising from the needs integration of the DMP tool within the joint architecture may also need to be supported.

4.1.5 Service Versions Monitoring Tool

4.1.5.1 High-Level Service Description

The software version monitoring framework SVMON collects the information on software versions of EUDAT services and their components installed in EUDAT CDI. The framework allows the monitoring of the following information:

- Operating system flavour and version of the endpoint.
- EUDAT service component software versions.
- Any service software version installed on the endpoint.
- History of changes performed for the given service.

The software version monitoring framework consists of the web-based central portal which collects the information on software versions, stores it in the database and displays it in a compact overview table and the agent software running on the service instance and reporting the attributes collected from service as shown in Figure 15. SVMON aims to provide information at least on software version and a few attributes for the services, which are not included into BDII information system.

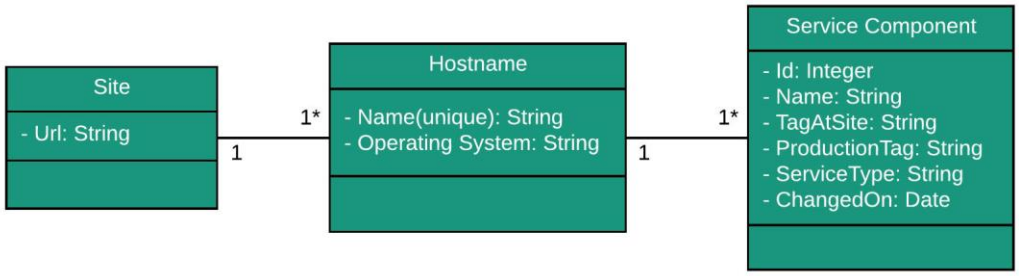

Figure 15 Attributes collected by SVMON from service provider

4.1.5.2 Maintenance plan

SVMON service hosted at KIT. The production instance of the SVMON is running in highly available virtual cluster, which provides full backup and efficient recovery procedure for the service.

The SVMON is divided into two different components: a client and a core server. The client implemented using Angular framework [R35] provides the user interface of the SVMON, while the core server, which is done with Java Spring Framework [R36] provides an API Rest interface to populate the user interface.

Both of them, client and core server, fetch the latest version configuration information from Gitlab at start up. The source code is divided into two branches available on GitLab: development and production. All the changes are made on the development branch that is merged to the production branch, once the functional tests are passed.

4.2 Integration plan

4.2.1 Integration of DPMT and GOCDB and Marketplace

As outlined in Section 3.2.2 The GOCDB and DPMT services will be integrated into Marketplace via the unified API interface. This integration step implies that both GOCDB and DPMT will provide any configuration information stored upon request coming from the Marketplace. No additional integration between GOCDB and DPMT are planned so far. The work plan for both GOCDB and DPMT will be focused on implementation of API interface according to the requirements defined by the Marketplace development team.

4.2.2 Integration of Operations Portal with Check-in

The Operations Portal is currently partially integrated with EGI Check-in.

It retrieves the user certificate DN from the EGI Check-in and provisions the user's roles by querying GOCDB and local DB with this DN.

The next step is to provision directly the user's role from EGI Check-in. So, it should be able to retrieve all attributes related to GOCDB (site/ngi administrators), but also for EUDAT and VO Membership.

4.2.3 Integration of SVMON with Pakiti

The integration work plan between SVMON and Pakiti clients has been adopted to provide unified installation and monitoring of software versions from service instances to facilitate release and deployment management process for EOSC-hub ecosystem. SVMON is integrated with Pakitii as shown in Figure 16.

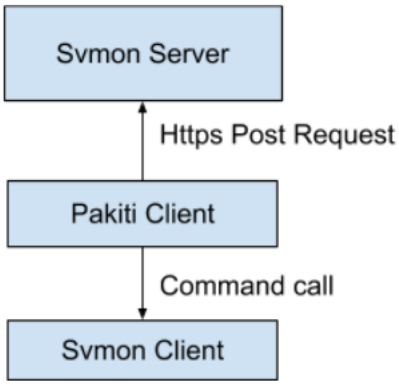

Figure 16 SVMON integration with Pakiti

First, the Pakiti client calls the SVMON client and the SVMON client sends the information that he has collected to Pakiti. The Pakiti client generates a report of the collected data and sends it via HTTP request to the SVMON server.

4.2.4 Integration of SVMON with DPMT and GOCDB

The integration of SVMON with EOSC-hub CMDB provided by DPMT and GOCDB implies the collection of all actual configuration information provided by DPMT and GOCDB like endpoints in "production" state deployed at sites etc. and map this information to the actual service versions installed at endpoints. The SVMON will provide two views of the configuration information separately for GOCDB and DMPT.

4.2.5 Integration of SVMON with B2ACCESS

The software version monitoring framework SVMON will be integrated with EUDAT B2ACCESS. As initial step only authentication layer will be provided in SVMON that users could access the configuration information. The authorization layer based on the role of the user retrieved from the AAI attributes will be implemented upon the further development of the SVMON tool.

5 Monitoring, Accounting, Messaging and Security Tools

5.1 Overview of services

5.1.1 Accounting Repository

5.1.1.1 High-Level Service Description

APEL is an accounting tool that collects accounting data from sites participating in the EGI and WLCG infrastructures as well as from sites belonging to other Grid organisations that are collaborating with EGI, including OSG and NorduGrid. 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. Figure 17 shows the current APEL system.

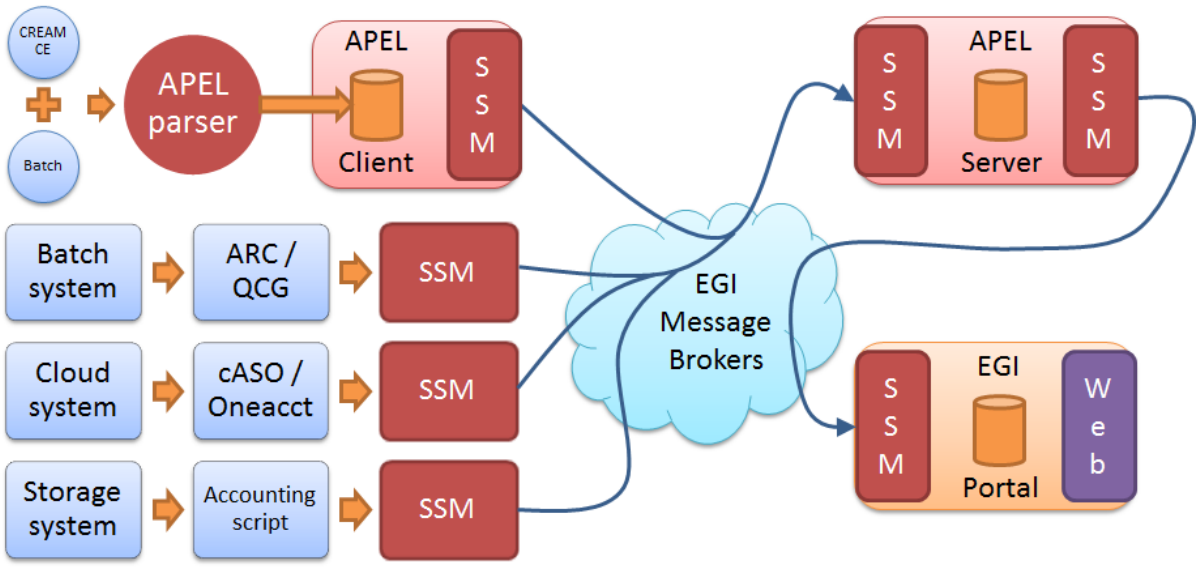

Figure 17 How multiple sources of different types of accounting information are integrated currently within the APEL central repository and accounting portal

APEL collects accounting information for compute, cloud and storage resources. Typically a site will deploy some form of accounting collector which will interact with the underlying resource provider and produce an accounting record in a supported format which is then sent to the APEL central repository via the EGI message broker network and our Secure STOMP Messenger (SSM). However, APEL is apathetic to the exact source of accounting data, so it is possible to set up regional APEL servers which receive the accounting data from national sites before sending a copy of the information on to the central server. A similar model exists for OSG, whereby sites

participating in that infrastructure send data to an OSG accounting server which then sends a copy of the accounting information to the APEL central repository.

In order to ensure the accounting service continues to run, it will need to evolve to meet new requirements.

5.1.1.2 Maintenance plan

Migration to the Argo Messaging Service (AMS)

APEL relies upon the existing STOMP message broker network for the sending of accounting data to the central Accounting Repository and the Portal. As the STOMP broker will eventually be decommissioned in favour of the AMS, support for the AMS will need to be developed and added to APEL messaging software.

Improvements to BAU processes

In order to reduce the support effort of the Accounting service, it shall be consolidated onto a smaller number of hosts and additional Icinga alerts will be added to monitor the backup and summariser processes.

5.1.2 Accounting Portal

5.1.2.1 High-Level Service Description

The Accounting Portal is a web application which has as its primary function to provide users, like VO managers, Site Admins, non-privileged users and other stakeholders, with customized accounting reports, containing tables and graphs, as web pages. It also offers RESTful web services to allow external entities to gather accounting data.

The basic architecture of the portal includes:

- A backend, which aggregates both data and metadata in a MySQL database, using the APEL SSM messaging system to interact with the Accounting Repository and several scripts, which periodically gather the data, and metadata described below.
- A Model represented by database schemas both external and internal which define database tables for several types of accounting (grid, cloud, storage, multicore, user statistics etc.) and metadata (topology, geographical data, site status, nodes, VO users and admins, site admins etc.) and a series of parameterized queries,
- A set of views that expose the data to the user. These views contain a form to set the parameters and metric of the report, a number of tables showing the data parametrized by two selectable dimensions and filtered by several parameters, a line graph showing the table data, and pie charts showing the percentage distribution on each dimension.

Statistics are available for view in different detail by users, Virtual Organisation (VO) managers, site administrators and anonymous users according to well-defined access rights.

5.1.2.2 Maintenance plan

The portal is structured into a production instance [R37], a development instance [R38] and a testing instance [R39]. Their contents have a daily backup with an rsync to an external machine. This rsync is monitored to detect problems with the backup procedure.

The instances are virtual machines instances inside the CESGA virtualization infrastructure. This means that physical hardware failures can be alleviated moving the VM from one node to another. Recently, improvements on our VM infrastructure allowed doubling the disk space allocated to them in order to avoid problems due to free space running out.

The SSM and DBLoader APEL import processes in the production instance are monitored with NAGIOS, so that interruptions in the data updates can be detected, this update are a backend process, so these interruptions are a minor problem instead of causing loss of service.

5.1.3 Monitoring

5.1.3.1 High-Level Service Description

ARGO is a flexible and scalable framework for monitoring status, availability and reliability of services provided by infrastructures with medium to high complexity. It can generate multiple reports using customer defined profiles (e.g. for SLA management, operations etc.) and has built-in multi-tenant support in the core framework.

ARGO supports flexible deployment models and its modular design enables ARGO to integrate with external systems (such as CMDBs, Service Catalogs etc.). During the report generation, ARGO can take into account custom factors such as the importance of a specific service endpoint, scheduled or unscheduled downtimes etc.

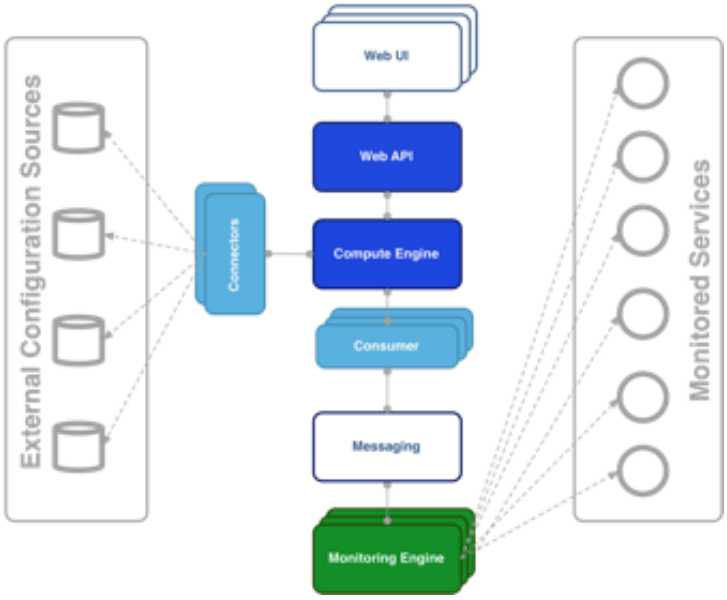

Figure 18 Argo monitoring system

For the Availability & Reliability monitoring, ARGO relies on a modular architecture as shown in Figure 18 and comprised of the several components, as described in the next subsections.

The ARGO Monitoring Service

For status monitoring, ARGO relies on Nagios. All probes developed for ARGO follow the Nagios conventions and can run on any stock Nagios box. ARGO provides an optional set of addons for the stock Nagios that provide features such as auto-configuration from external information sources [R40], publishing results to external Message Brokers etc.

In order to use the new messaging service, the monitoring engine also supports the new AMS Publisher. The AMS publisher is a new component acting as bridge from Nagios to ARGO Messaging system. It's integral part of software stack running on ARGO monitoring instance and is responsible for forming and dispatching messages that are results of Nagios tests. Ready and running on devel infrastructure. It is running as a UNIX daemon and it consists of two subsystems:

- queueing mechanism
- publishing/dispatching part

Messages are cached in local queue with the help of OCSP Nagios calls and each queue is being monitored by the daemon. After configurable amount of accumulated messages, publisher that is associated to queue sends them to ARGO Messaging system and drains the queue. argo-nagios-ams-publisher is written in multiprocessing manner so there is support for multiple queue/publish pairs where for each, new worker process will be spawned.

The ARGO Connectors

Through the use of custom connectors [R41], ARGO can connect to multiple external Configuration Management Databases and Service Catalogs. Already there are connectors for the EGI and EUDAT e-Infrastructures.

The ARGO Consumer

The ARGO Consumer [R42] is ingesting monitoring results in real-time from external Message Brokers. The consumer is responsible for the initial pre-filtering of the monitoring results and encodes them using AVRO serialization format before passing to the Compute Engine.

The ARGO Compute Engine

A powerful and scalable analytics engine built on top of Hadoop and HDFS. The Compute Engine [R43] is responsible for the aggregation of the status results and the computation of availability and reliability of composite services using customer defined algorithms. The reorganization of the Compute Engine to support stream processing in real time is one of the key new factors. A new streaming layer is introduced. Monitoring results flow through the AMS, to the streaming layer (in parallel to the HDFS). The streaming layer is used in order to push raw metric results to the metric result store and to compute status results and push them to the status store in real-time.

The ARGO Web API

The ARGO Web API [R44] provides the Serving Layer of ARGO. It is comprised of a high performance and scalable datastore and a multi-tenant REST HTTP API, which is used for retrieving the Status, Availability and Reliability reports and the actual raw metric results.

The ARGO Web UI

The default web UI [R45] is based on the Lavoisier Data Aggregation Framework.

5.1.3.2 Maintenance plan

We currently maintain a devel instance of every component in parallel to our production instance. This allows testing each update in a production like environment before it is rolled out. We use Ansible Scripts controlled by Jenkins and we usually roll out updates in the 1st week of each month. Our regular maintenance tasks include:

- Re-computations of Availability / Reliability
- Security updates
- IGTF CA updates
- Probe updates
- Product updates
- Log rotation and management

5.1.4 Argo Messaging

5.1.4.1 High-Level Service Description

The ARGO Messaging Service is a Publish/Subscribe Service, which implements the Google PubSub protocol. It provides an HTTP API that enables Users/Systems to implement message oriented service using the Publish/Subscribe Model over plain HTTP. 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. In the current deployment a haproxy server acts as load balancer for the 3 AMS servers running in the backend.

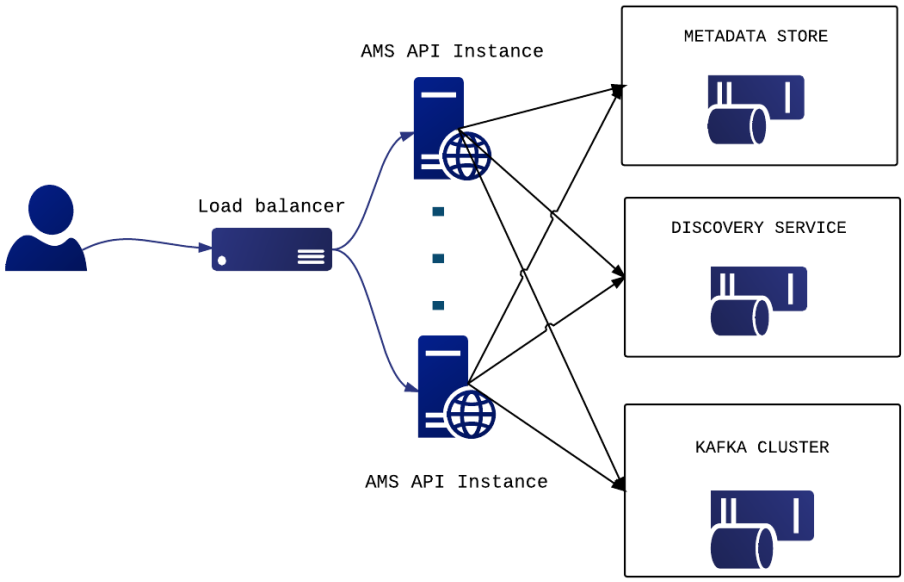

Figure 19 ARGO messaging service

5.1.4.2 Maintenance plan

We currently maintain a devel instance of every component in parallel to our production instance. This allows testing each update in a production like environment before it is rolled out. We use Ansible Scripts controlled by Jenkins and we usually roll out updates in the 1st week of each month.

Our regular maintenance tasks include:

1. Security updates
2. IGTF CA updates
3. Product updates
4. Log rotation and management

5.1.5 Security Tools

5.1.5.1 High-Level Service Description

Secant

Current utilization of cloud introduces new possibilities on how cloud facilities can be put at risk. A commonly seen pattern is an image of a virtual machine that exposes vulnerabilities which can be easily exploited once the virtual machine is instantiated and connected to the Internet. In order to detect common vulnerabilities, the Secant framework for image assessment has been developed.

Secant provides a set of services that obtain VM images available from a repository and performs their security assessment. The assessment is designed to detect typical vulnerabilities that are introduced either by configuration errors, weak credentials, insufficient patch management, etc. The checks are performed automatically, which enables straightforward integration with existing

image management. The goal, however, is not to replace detailed security audits done by human experts.

Secant performs dynamic analysis of images, which makes it possible to check their behavior and services exposed when the VM is running. After the analysis has started, Secant instantiates a virtual machine from the appliance that is being verified and mounts the security checks. During the first phase, Secant launches a series of external scans that try to detect vulnerabilities exposed by the machine to the Internet. Following these tests, and if the machine supports it, Secant runs a series of internal probes on the virtual machine which checks security properties of the installed software. Both internal and external probes are modular, and new tests can be easily added when needed. After the probes are executed, Secant processes the results and generates the assessment report.

Secant was enabled on the Application Database and can check images that are stored there. AppDB and Secant communicate using the EGI Argo Messaging service, upon which a message flow was implemented to pass information about available appliances and their status after the analysis.

Pakiti

Pakiti provides a tool to monitor and evaluate patch management of Linux systems.

Pakiti is using 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 an analysis done by the server. The Pakiti server compares the versions against 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.

Pakiti has a web-based GUI which provides a list of the registered systems. The overview helps system administrators keep multiple machines up-to-date and prevent unpatched machines to be kept silently on the network. The information processed is also available via programmatic interfaces.

The Pakiti client is part of the standard security probe set and is used to send reports from the monitored sites to the EGI Pakiti servers. The services are used by the EGI CSIRT and security managers of sites and NGIs.

5.1.5.2 Maintenance plan

Secant

Secant runs as a service operated by CESNET, utilizing the local cloud facilities for assessments. Given the changes planned on the CESNET infrastructure, we foresee the need to move from OpenNebula to another solution, which will require changes to a Secant module.

Secant will be maintained and adapted mainly to follow the requirement of the cloud operators and security teams. The security checks will be adapted to reflect current risks and findings from security incidents. Secant will need to improve checks performed with different contextualization settings.

Pakiti

The production Pakiti server will be updated to use the new implementation of the services, which delivers significant improvements over current implementation. The service will continue to support new types of vulnerabilities, including those specific to EGI. Integration with EOSC AAI will ease access for the users.

5.2 Integration Plan

5.2.1 Accounting Repository - EUDAT accounting service

Figure 20 shows the current EGI storage accounting system, which EUDAT accounting information will be integrated into.

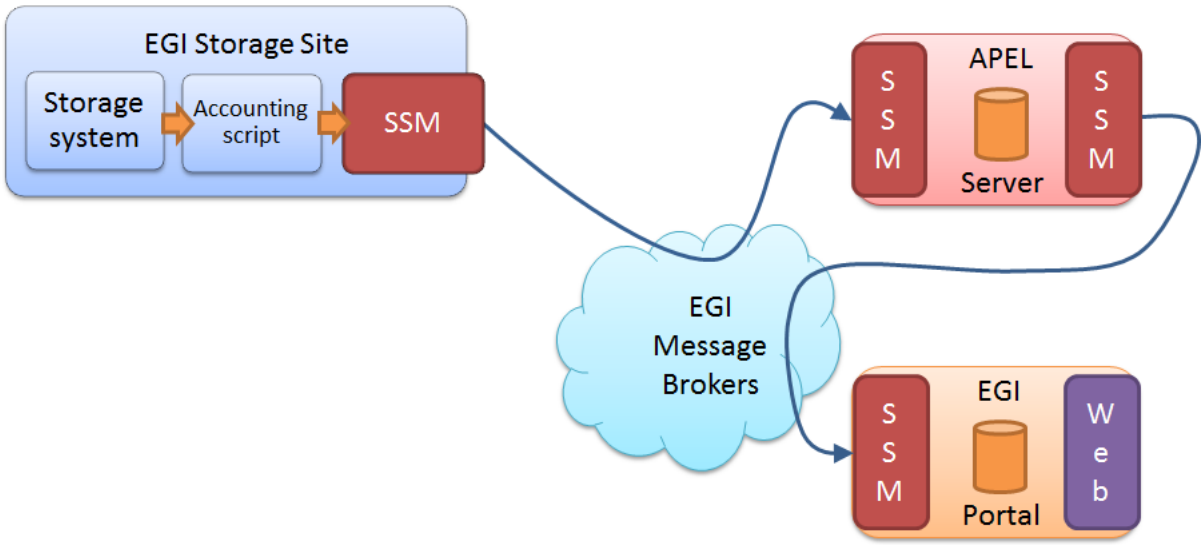

Figure 20 The current EGI storage accounting system

Because the APEL central repository is apathetic to the source of accounting data, new sources and storage systems can be added with ease, only requiring they produce accounting records in the correct format. Accounting data from EUDAT storage sites are already collected by the DPMT component within EUDAT's infrastructure. To integrate these data with the APEL central repository, an accounting collector will be developed to convert the data stored within the DPMT component into an extended StAR format [R46]. Once the records are produced, they will be sent to the APEL central repository via the SSM and the EGI Message Brokers where it will be combined with Storage data from EGI and other infrastructures, before being sent to the accounting portal for visualization. The proposed storage accounting system is displayed in Figure 21.

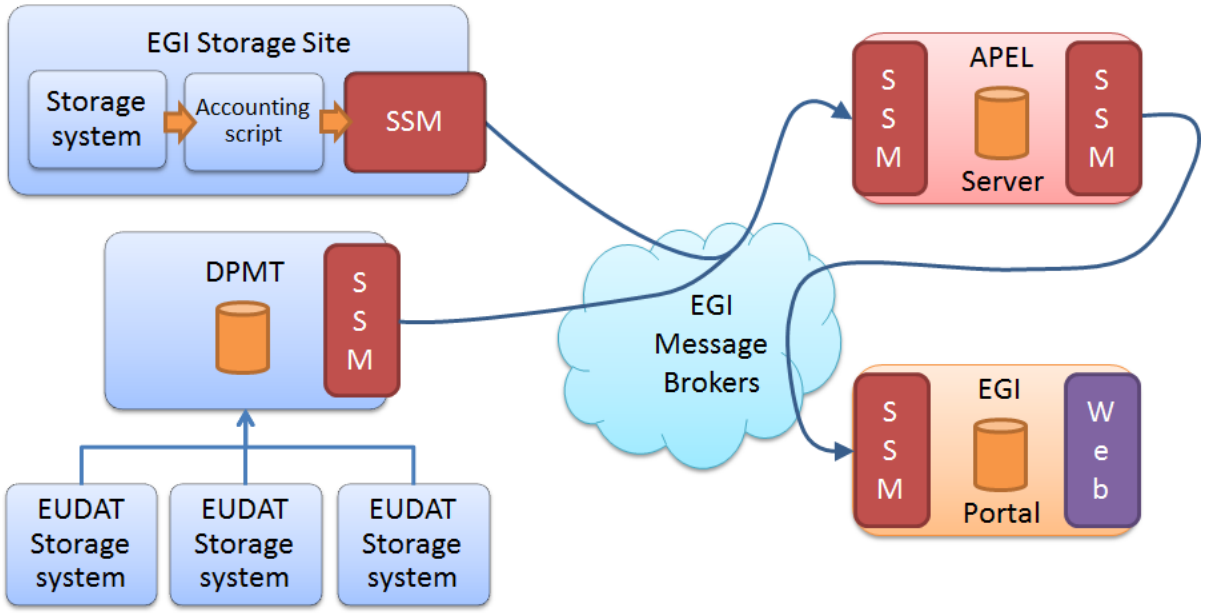

Figure 21 How EUDAT accounting System can be integrated with the APEL central repository and accounting portal

Under this system, individual EUDAT storage sites continue to report to DPMT using the current method, so do not need to upgrade or edit the local accounting scripts. The SSM also remains the same. The APEL central server will need to be updated to support the extended StAR format.

Accounting Portal

After repository level integration is complete, it may be necessary to produce custom views in the portal for the EUDAT communities, in a similar fashion to what is done now with WLCG views. The storage views are “Average” based, so each site can have several measurements along the day and the portal will average them, it will show also minimal and maximum values, it lets the user select min/max/avg Logical Terabytes consumed and min/max/avg files used. The two dimension of the view table can be date, region. site, vo, storage system, storage share, directory path, subgroup and role.

In addition to this approach, any special requirement from EUDAT community on storage view can be implemented upon request.

5.2.2 Monitoring

In order to provide an integrated view of the monitoring done for all EOSC-hub services the Argo Monitoring Framework will add support for the following.

- **Harmonization of the user facing web interfaces**
  These include the ARGO A/R and Status web interface and the POEM web interface.

Currently these web interfaces have different look and feel and they do not share user sessions. In this task, we will create one unified theme for all the ARGO web user interfaces (Fig. 22). Furthermore, the entire ARGO web UIs will be enabled to share the same user sessions, so that users do not have to authentication on each of them separately.

The Web UI will be built on the top of the API. So we will provide a uniform access but depending from the profile selected by the user we will use a different key to access to the API and consequently provide access to the monitoring information in relation with the request of the user.

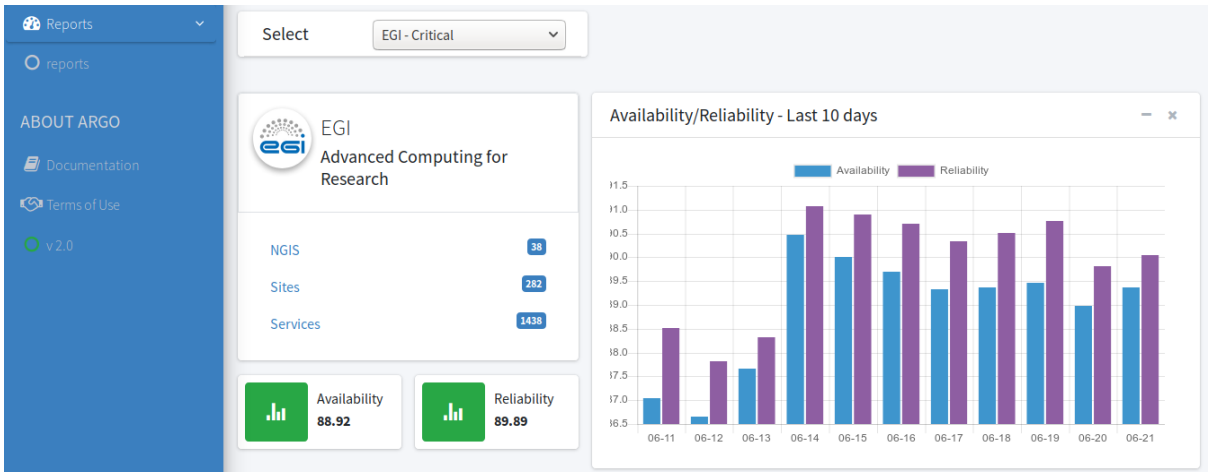

Figure 22 Unified theme for all ARGO web user interfaces

- **Single stop shop for service enablement and configuration**

Currently, the addition of a new customer on the ARGO Monitoring Service is a manual process that has to be performed by the ARGO team. With this task we want to implement a service management web interface through which customers (e.g. VO managers, Infrastructure Managers etc.) will be able to configure the monitoring service to their liking. We have already taken the first towards this direction with the addition of full life-cycle management of the probes by the product team through the POEM web service. Of course this is not enough. We want to give the ability to the customers to sign up for the service and configure it for their infrastructure through just a few clicks through a web flow.

- **Customer defined thresholds**

The ARGO Monitoring Service is generating Status and A/R reports based on the metric results that it gathers from the execution of the monitoring probes. Each metric result includes a status and performance data that typically contain values related to the provided status. Currently the ARGO Monitoring Service relies solely on the statuses returned by the probes in order to generate the status and A/R reports. Each probe has a hard-coded built-in static logic in order to compute the probe status. Although this have been proven sufficient for the purposes of infrastructure monitoring up to now, it does not

give us any flexibility in providing different SLA targets to customers. For example, let's say that a probe checks the average response time of the tickets in GGUS. With the current implementation, the acceptable response time is part of the probe configuration. If we want to have different acceptable response time for a specific customer, then we will have to create a new probe configuration and execute a new test. What we propose here, is to move the metric status computation to the ARGO Compute Engine, so the monitoring probes when executed will return the actual data (e.g. the average response time) and then on the ARGO Compute Engine we can have multiple profiles, which will be used in order to generate reports based on customer defined threshold.

5.2.3 Messaging

The plans for the ARGO messaging service are:

- Phase out the Broker Network based Service and migrate users to the Argo Messaging Service (Pub/Sub)
- Add support for alternate authentication methods such as OIDC and x509 certs, which will allow a number of services such as APEL Accounting and EGI Fedcloud to use the service.
- On-board thematic services in need of a messaging service.

5.2.4 Security Tools

- The integration of Secant and AppDb will be finalized and verified on images maintained by the users.
- Pakiti client will integrate support for packages processed by SVMON so there is a single client usable by both the services.

6 Helpdesk Services and Tools

The helpdesk is a basic service which is required in any IT infrastructure to provide an efficient support for its services. For EUDAT and EGI infrastructure, the tools used up to now are GGUS for EGI and RT for EUDAT. These two mature helpdesk tools provide a complete helpdesk service for both infrastructures, used in the incident and service request management and also in the problem management processes.

6.1 Overview of services

6.1.1 GGUS

6.1.1.1 High-Level Service Description

GGUS is the central helpdesk service for the EGI, WLCG e-Infrastructures and more than 40 other VOs. In collaboration of worldwide 2000 experts, grouped in 120 second level support teams, more than 130000 tickets have been solved meanwhile.

GGUS is synchronized with 16 other grid related helpdesk systems and interfaced with existing EGI tools like the GOCDB or the Operations Portal to exchange system relevant information. Service availability rates of more than 99% have been achieved all the years. 6 releases per year are planned to keep the system maintained and to ensure the implementation of required changes. The GGUS system is developed, maintained and hosted at KIT.

From the technical point of view the GGUS system is divided into three environments: development, test and production environment. Each environment includes three layers:

Presentation - web frontend to provide the entry point for the user interface.

Logic - AR Server (BMC) which executes the workflow rules and performs the main tasks. AR Server is providing the communication interface between external systems and is accompanied by the email-engine to provide the additional mail-based interface into the helpdesk system.

Backend - Oracle DBMS

GGUS is integrated into an on-call duty service. In case of a service incident the on call engineer (OCE) will fix the problem according to instructions described in the on-call duty service wiki. If the OCE does not succeed to fix the problem, the GGUS experts can be called.

6.1.1.2 Maintenance Plan

xGUS/GGUS consists of a three-stage system. A development system exclusively accessible by developers at KIT, a pre-production system to test functionalities within the project partners and the production system. All instances are maintained every two months. During the downtime, system updates and security patches are installed and the system can be equipped with requested and approved features. In addition, servers on VM with daily data backup, two frontend servers at

different locations, connected to a load balancer and the integration into a 24x7 on-call service contribute to the security and availability of xGUS/GGUS.

6.1.2 EUDAT-RT

6.1.2.1 High-Level Service Description

The EUDAT Trouble Ticketing System (TTS) is an important building block of the operational infrastructure of EUDAT CDI. The TTS is used to track internal support requests and the problem solving process. The TTS system at EUDAT uses the Request Tracker software from Best Practical Solutions LLC, the current version used is 4.2.12 and it is currently ported to support the B2ACCESS authentication from EUDAT.

The EUDAT helpdesk service uses the RT system to provide the 1st and the 2nd level support for all the EUDAT services. The different levels of support are managed by dedicated teams, which are using the trouble ticketing system at https://helpdesk.eudat.eu to manage and resolve the received tickets. The TTS system is currently including more than 41 teams managing different queues, each one operating part of the EUDAT infrastructure, within these 41 teams, it is included each of the sites providing resources and the 2nd level support for all the EUDAT B2 services family (B2ACCESS, B2SHARE, B2NOTE, B2DROP, B2HANDLE, B2STAGE, B2FIND,B2SAFE and B2GETHER).

The helpdesk system is accessible by all EUDAT partners. Tickets can be opened via the web form available at https://www.eudat.eu/contact-support-request, which is publicly open to assure a easy contact way for the current users or any person interested to have more information about EUDAT CDI infrastructure.

In addition, the TTS provides the required information to fulfill the helpdesk requirements according to OLAs and project-specific SLAs, providing Time to resolve and Time to answer statistics for all the queues and for any specific ticket.

The 1st level support team of EUDAT is in charge of answer any request received and if required, to forward the request to the specific 2nd level support in charge of the topic of the request/incident.

6.1.2.2 Maintenance Plan

The EUDAT-RT service is part of the core services of the EUDAT-CDI. The service is currently migrated from CINECA to BSC site, where it is installed a pre-production version of the service. At the BSC site the service will have a production version and a development version, in order to test new functionalities and check new updates.

The service is upgraded with the latest security patches provided by the software. In addition, the tool is running under a VM with daily backup and with the option to a fast migration to new hardware if required.

6.2 Integration plan

6.2.1 Integration of the helpdesk tools

The incident and problem management process relies on the current EGI and EUDAT mature helpdesk systems. However, this complexity will be hidden to the end users that could submit their requests through a single-entry point, a unified system that will act as first level of support forwarding, when necessary, the tickets to the appropriate underlying support system. The unified ticketing system used for EOSC-hub project will be xGUS, a developed lightweight clone of GGUS that permits a basic level of interoperability with GGUS and RT through a SOAP interface. The unified ticketing system, as shown in Figure 23, will provide a centralized place to manage the first level support tickets without losing the information of the tickets forwarded to the EUDAT or EGI helpdesk systems and providing a central contact point for all users independently of the final service/infrastructure finally used. The first level support of all the tickets received in the unified ticketing system will be managed by a 1st level support team from EOSC-hub project, this team will be in charge of answer the requests or incidents received from the unified ticket system and to forward them to the final service provider (EUDAT helpdesk or EGI helpdesk) in case of complex tickets or specific to a service therefore cannot be resolved by the 1st level support team.

The current work plan is scheduled to have the production version of the xGUS helpdesk on M10 when the Milestone “M5.5 Consistent Helpdesk system available with 1st and 2nd line support” is defined. The development version of xGUS to test the integration between xGUS, EUDAT-RT and EGI-GGUS is expected on M7 in order to have 3 months for the integration of the helpdesk systems and the final testing of the platform.

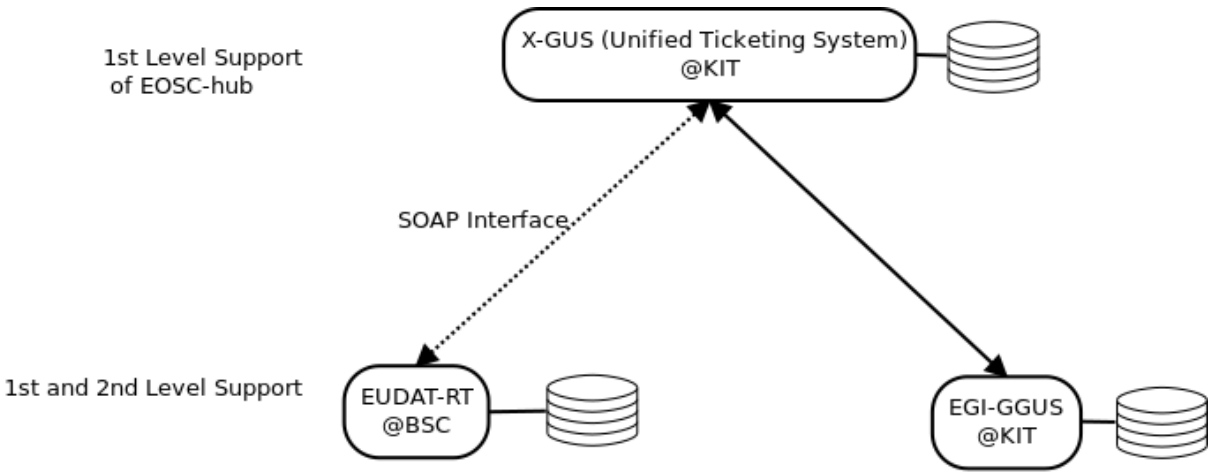

Figure 23 Integration diagram of EUDAT and EGI helpdesk systems

6.2.2 Integration for the helpdesk tools using xGUS

xGUS was developed as a lightweight clone of GGUS with the goal to provide a basic ticket system that can easily be set up and adapted to the requirements of user communities within e-

infrastructures. Figure 24 shows the xGUS submission form and Figure 25 demonstrates ticket search interface.

It contains all the expected features of a helpdesk system needed to provide effective user support such as a form to submit a ticket, a ticket search (see table below) and the sending of email notifications at changed ticket status.

Support staff is grouped in support units. Through a system internal user administration permissions can be granted which allow for various support work (Support staff, user administration, news administration, portal administration).

The xGUS system can be accessed either by x509 grid certificates, via the EGI CheckIn or by login with username and password.

xGUS runs on the same technical platform as GGUS, therefore ticket exchange with GGUS is enabled by default - but it can also be used as an independent helpdesk system.

For synchronizing tickets with other ticketing systems e.g. the EUDAT RT system, xGUS provides SOAP web services for creating and modifying tickets.

A detailed documentation about the GGUS SOAP web services, which also applies to xGUS, can be found here [R47].

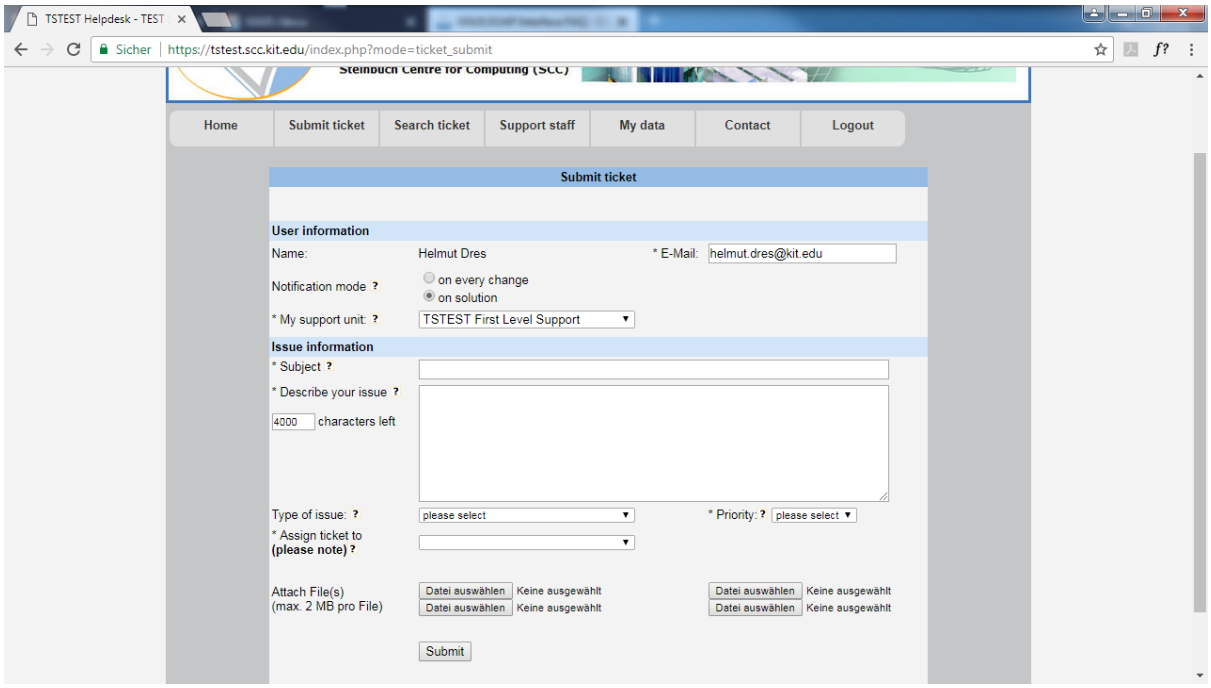

Figure 24 xGUS ticket submission form

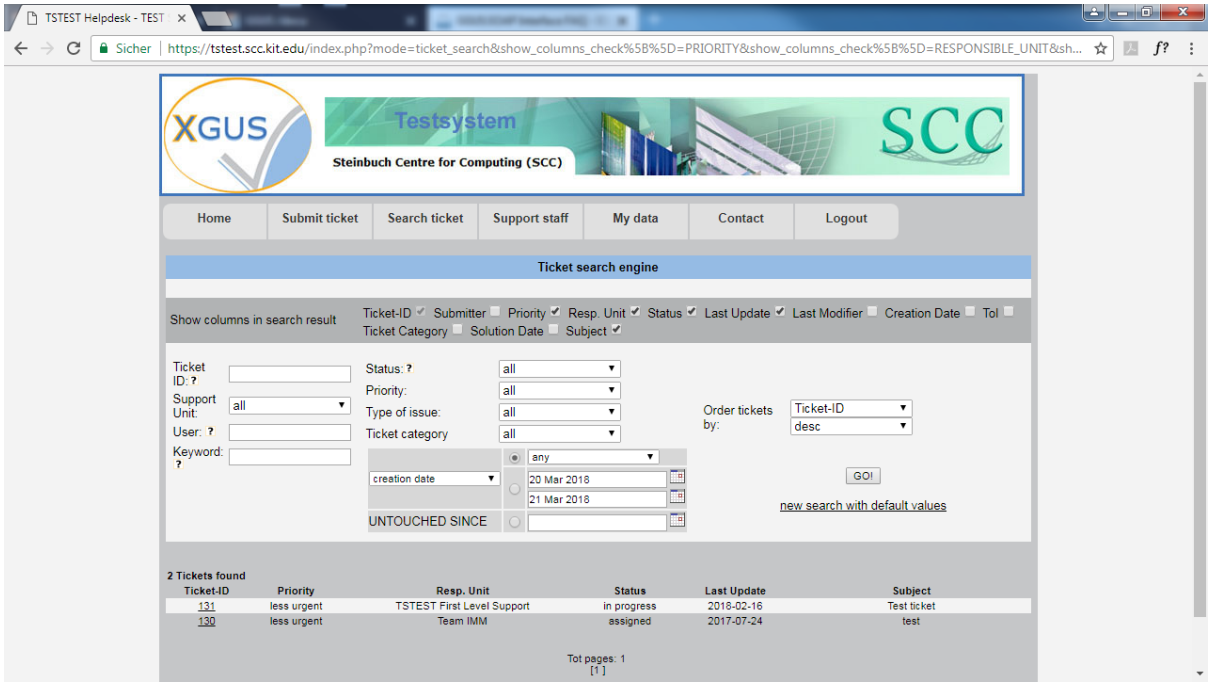

Figure 25 xGUS ticket search interface

6.2.3 Integration of EUDAT-RT with xGUS

The main idea at this point is to synchronize the operation of EUDAT-RT with xGUS via SOAP function calls and messages. Therefore, we will use as a starting point the existing GGUS-RT integration defined in https://wiki.egi.eu/wiki/GGUS-RT_Interface_Task_Force. In the previous webpage, there is a description of all the web-services which are already available in GGUS, for instance:

- TicketCreate
- TicketModify
- AddTicketAttachment

Concerning the ticket identifiers, we plan to add a table in the EUDAT-RT database in order to match the ticket identifiers between EUDAT-RT and xGUS.

The main workflow would be the following:

1. The user login in the xGUS helpdesk
2. The user submits a ticket in the xGUS helpdesk
3. A First level Support member of EOSC-hub check the ticket and depending on the topic, it is directly answered or assigned to the right infrastructure (EGI or EUDAT)
4. If the ticket is for EUDAT-RT, the xGUS systems create a new ticket in the EUDAT-RT. The xGUS id and the EUDAT-RT id is saved in the EUDAT-RT database.
5. A First or Second level Support member from EUDAT will take the ticket.

6. The owner of the ticket in the EUDAT-RT will reply to the ticket and the user will receive the reply via email. This action will generate an update in the EUDAT-RT database and in the xGUS system.
7. The user will reply to the ticket via email. The EUDAT-RT system will process this action and update the ticket both in EUDAT-RT and xGUS systems.

The implementation of these SOAP interface will start on M4 in order to have the first version on M7, as we expect the first development version of xGUS on this month and it will permit the testing of the SOAP calls in order to assure the correct integration of the EUDAT-RT with the xGUS helpdesk before the M10 when the service need to be in production.

7 Application store, Software Repositories and other Collaboration Tools

7.1 Overview of services

7.1.1 Applications Database

7.1.1.1 High-level service description

The Applications Database (AppDB) [R48] is a central service that stores and provides to the public information about:

- software solutions in the form of native software products and virtual or software appliances,
- the programmers and the scientists who are involved, and
- publications derived from the registered solutions.

In addition, AppDB is the responsible unit within the EOSC-hub ecosystem, for:

- distributing the registered VM images to the resource providers and
- enabling users to deploy and manage Virtual Machines to the EGI Cloud infrastructure through AppDB's dashboard services.

7.1.1.2 Maintenance plan

The AppDB is hosted and operated by IASA. The service is structured into a production instance and a development instance [R49]. Their dbs and contents are backed-up on a daily to an external storage. The release flow consists first to push the new version to the development instance, testing the new features and if everything is as expected the release is pushed into production.

7.1.2 GitLab

7.1.2.1 High-level service architecture

GitLab is a web-based platform, operated under the open-source license [R50], which provides an integrated environment for software development including Git-repository, issue tracking system, wiki, continuous integration module etc. GitLab is used as integrated solution for full software development cycle and provides rich APIs for integration with other services. Fully automated workflows for software testing and deployment implemented in GitLab can be used for efficient release and deployment management for EOSC-hub distributed services.

GitLab instance deployed at KIT is integrated with Container Registry [R51], which allows storing Docker images. The federated access to the GItLab is provided by the EUDAT AAI solution B2ACCESS, thus GitLab resources and Git-repositories are available for many research communities and scientific organisations.

7.1.2.2 Maintenance plan

The GitLab is hosted and operated by KIT. The production instance [R52] is running on virtual ESX cluster. The image of virtual machine is backed-up hourly using the ESX cluster backup technology with depth of two weeks. In addition the database and configuration of GitLab is backed-up daily.

The release cycle of GitLab instance is corresponding to official release cycle of GitLab Inc., which produces the new release monthly. Before deployment to production, the new release and its features are tested on development GitLab instance.

7.1.3 EGI software repository

7.1.3.1 High-level service architecture

The EGI Software Repository [R53] implements all necessary management workflows for the UMD & CMD Middleware Distributions and the Community Software, while providing a unified point of access to these resources, which are described below:

- the Unified Middleware Distribution (UMD) is the integrated set of software components contributed by Technology Providers and packaged for deployment as production quality services in EOSC-hub.
- the Cloud Middleware Distribution (CMD) distributes OpenStack and OpenNebula integration components, developed by Cloud Technology Providers. Two different distributions are technically available, CMD-OS for OpenStack and CMD-ONE for OpenNebula.
- the Community Software is strongly integrated with the AppDB system and consists of repositories of binary artifacts, provided by communities and individuals affiliated with the EOSC-hub project.

7.1.3.2 Maintenance plan

The EGI Software Repository is hosted and operated by IASA and well integrated into its infrastructure. This includes proper maintenance of all underlying components including the hardware, the operating system, up to the software libraries and frameworks used. Local monitoring and backup is performed on a daily basis.

7.2 Integration plan

7.2.1 Integration of the AppDB VMOps with the GGUS

Although AppDB VMOps [R54] controls the lifecycle of deployed VMs, it does not control the cloud providers that host them. Occasionally, providers might not offer certain functionalities such as networking or storage, due to resource constraints, or they might not be able to restore some VMs after a scheduled or unscheduled downtime occurs. Such incidents may result to malfunctioning VMs or deployment failures which, in some cases, can only be perceived by the VM's owner. Currently, there are no means for users to address such issues.

Integration with the GGUS service will provide a channel for users to communicate their issues with cloud provider administrators and resolve them. A graphical interface will be available to let users create a ticket within the GGUS system, addressing a specific VM. The AppDB VMOps portal will enrich the ticket with all the necessary information related to the specific VM, in order to help site administrators to locate and resolve the issue. Moreover, users will be able to visit and review progress on all tickets they have created for each VM, by means of the same interface, at any given time.

7.2.2 Integration of the AppDB with the EOSC-hub GitLab

The integration of AppDB with the EOSC-hub GitLab service will provide source code repository access for any registered software item. The creation and availability of such repositories will be automated upon software registration, without the need of user intervention. This will allow developers to store and maintain their source code by using the well-known Git toolset. The rest of the users will be able to either view the activity of a registered software item repository from the AppDB web portal, or inspect the code itself from within the EOSC-hub GitLab public web interface. Moreover, this integration could be extended in the future in order to include other types of registered items, such as virtual appliances, so that contextualization and configuration files may be stored and maintained.
The integration of AppDB with GitLab implies the implementation of common authorisation and authentication system, which will allow AppDB user to seamless access the GitLab source code repository for any registered software item in AppDB.
One of the possible scenarios of this implementation, to propagate the role information and group membership from AppDB to GitLab, is to use user attributes with one of the AAI solutions e.g. Check-in. This requires the translation of user attributes into GitLab project permissions. Another approach is to provide lookup APIs in AppDB with user information, which can be called by GitLab, when the permissions for the GitLab group or project should be defined. At the moment of writing both approaches are under investigation.

7.2.3 AppDB Information System extension

The AppDB Information System (AppDB-IS) [R55] is the unit responsible for collecting and correlating data from external services. The collected data are of significant importance for the smooth operation of the AppDB main system and its satellite services. Currently, the services the AppDB is integrated with are: GOCDB, the Argo monitoring system, and the Top-Level BDII.
With respect to the Top-Level BDII, harvested data currently follow the GLUE 2.0 schema, but this will be subject to change in the upcoming months, since the GLUE 2.1 specification is about to be released.

7.2.3.1 Adapt GLUE 2.1 schema

The AppDB-IS should evolve to support the GLUE 2.1 schema, which allows the representation of valuable information about service endpoints capacities and capabilities. Apart from AppDB

services, its implementation will also allow higher level services, such as brokers, take informed decisions on how to make better use of resources for a given workload.

7.2.3.2 Alternative information transport mechanism

The current setup consists of performing periodical ldap queries to the Top-Level BDII, in order to fetch the respective data. However, this information transport mechanism has been proven to be complicated and error-prone. Therefore, new transport mechanisms should be explored and implemented in order to avoid errors, data delays and unneeded complexity.

7.2.4 Enrich AppDB digital objects with PIDs

AppDB has long supported canonical URLs for every registered digital object, as a means to make finding and sharing information easier and more reliable. These are based on one or more canonicalized versions of the registered digital object's name, which serve as suffixes to AppDB's "store" URL prefix. As such, these canonical URLs inherently carry the object's name as metadata for the entry. Nevertheless, link rot, which could arise in our case from changing a digital objects record name, for example, cannot be prevented this way, nor can one perform comprehensive searches from outside AppDB, based on metadata other than the name.

In order to address these issues, AppDB plans to extend its platform with support for persistent identifiers (PIDs) for several registered digital object, through the use open standards such as the HANDLE system. These identifiers remain immutable within centralized indexes for the entire life-cycle of a digital object, thus preventing link rot, and may carry additional metadata about the object, which may enable sophisticated queries, otherwise not possible though standard web searches. The first digital objects to be included in this scheme are software, virtual appliances, and releases and versions thereof, respectively. AppDB will make use of EUDAT's B2HANDLE service in order to register HANDLE PIDs for these objects and will be responsible for maintaining them and keeping relevant metadata up-to-date. Moreover, AppDB will explore the possibility of integrating with other metadata repositories, such as OpenAIRE, through the use of these PIDs, as well as the extending their use to other registered digital objects such as datasets.

7.2.5 VM image list management migration

Currently, VO-wide image list management is realized through a separate section/panel within the AppDB portal, which makes its use, as well as its maintenance, cumbersome. The plan is to develop a dedicated dashboard, the *VO Image list dashboard*, which will offer the same functionality and be integrated with the VM security and endorsers dashboards. This way, important information such as VM security status and information related to VM endorsers will be available to VO managers while compiling VO-wide image lists they are responsible for.

7.2.6 Notification or push based image list distribution mechanisms

The current mechanism for VM image distribution relies on a polling model, where sites periodically query AppDB for changes in image lists. This causes significant delays in image propagation to sites. The plan is to explore and implement notification or push-based methods for

communicating changes in the image lists to the sites and to trigger the propagation of those changes as immediately as possible.

7.2.7 Development of the VM Security dashboard

A dashboard through which the security team will be able to:

- perform queries to get details on specific VM images, which are published by the AppDB system and are available within the infrastructure; such details may include the sites on the VM images in question are located, their author, the VOs that support them, etc.
- tag a VM image as insecure and propagate this information to the appropriate site administrators and VO managers through the GGUS system, so that they may perform further manual actions, such as banning the insecure VM image and suspending the respective VM instances from sites.

The VM Security dashboard will be integrated with the rest of the dashboards offered by the AppDB ecosystem:

- with the VMOps dashboard, in order to exclude insecure VM Images from the list of images available to end users (vm_operators),
- with the VM endorsers dashboard, in order to inform endorsers that a specific image has been characterized by the security team as insecure, and
- with the VO image list dashboard, in order to prevent VO managers from selecting insecure images when compiling the VO-wide image lists they are responsible for.

7.2.8 Development of the VM endorsers dashboard

A dedicated dashboard accessible by a special group of experts, through which they will be able to endorse or un-endorse VMs (VA versions) registered with the AppDB system. The “endorsement” details will be disseminated to VO managers, in order to assist them on deciding whether they should include a specific VA in the VO-wide image lists they are responsible for. Although this has not been clearly defined, yet, the plan is to have the endorsements be “scoped”, in the sense that an endorser should be able to endorse a registered VM either under the scope of a specific VO, or a specific project, a specific activity such as security or operations, etc.

7.2.9 Revise and enable datasets section of the AppDB

During the EGI-Engage project, the AppDB has been extended with new capabilities to manage and expose information about reference datasets and their replicas across EGI. Under this scope, the user is able to add/update/remove reference datasets in a versioned manner, defining if the registered entry is about primary or derived dataset (segment of a primary). This work is available at the AppDB development instance [R56] and during the EOSC-hub project it should be revised in order to enable it in the production or not.

8 OpenAIRE integration

8.1 Applications Database

8.1.1 Contribution on guidelines for software repositories and other products

The AppDB team will contribute to the definition of the software and other products guidelines that will eventually enable AppDB users to incorporate their research product into the OpenAIRE infrastructure for discoverability and utilizing value-added services provided by the OpenAIRE ecosystem.

8.1.2 Adoption of the guidelines by the AppDB

An AppDB sub-service will be developed exposing the metadata of the registered products to the OpenAIRE harvesting services, by following the OAI-PMH protocol. The registered items that will be populated through that sub-service will be of type, Software, Software release, Virtual Appliance and Virtual Appliance version. The metadata should follow the respective OpenAIRE guideline i.e. software-guideline for Software Items and Software releases and virtual-appliance-guideline or other-product-guideline for Virtual Appliance and Virtual Appliance versions.

8.1.3 Integration of the AppDB with the OpenAIRE Research Impact Dashboard

The necessary components will be developed for integrating the AppDB with the OpenAIRE Research Impact Dashboard. The overall aim of this activity is implementing a system that allows the AppDB service to be notified/triggered in case the AppDB records harvested by the OpenAIRE have been enriched with additional metadata or links to other products.

8.2 Integration with OpenAIRE AAI

Many research communities will be using resources and services provided by EOSC-hub and OpenAIRE. Both infrastructures operate AARC BPA-compliant AAI gateways to their services. Users should be able to share data, access services and roam across infrastructures in a seamless manner.

8.2.1 Documentation of use cases for AAI integration between OpenAIRE and EOSC-hub

This activity will investigate example use cases for the AAI integration in support of sharing data and accessing services between OpenAIRE and EOSC-hub. Service experts from both OpenAIRE and EOSC-hub will be involved in order to select such use cases.

8.2.2 Piloting of use cases for AAI integration between OpenAIRE and EOSC-hub

This task will pilot the identified integration use cases. As such, it will involve the AAI teams from OpenAIRE and EOSC-hub, as well as the related service experts. Both the OpenAIRE and the EOSC-hub AAI follow the architecture and policy guidelines from AARC, i.e. they are built on top of eduGAIN with support for the most common authentication and authorisation protocols (SAML & OpenID Connect). Thus, not much development activity is expected. Still, this integration task will

take some effort on enabling. Integration in the production environment is foreseen for PY2.

8.3 Data Management Planning Tool

The Data Management Planning (DMP) tool is a web-based service that has been jointly conceptualised by EUDAT and OpenAIRE as described in Section 4.1.4. The conceptualisation effort started already in mid-2017. Until the start of the EOSC-hub project, EUDAT worked on an implementation that allows project principal investigators (PIs) to specify their own data management plans based on a dynamic questionnaire that follows the H2020 DMP guidelines (easy.DMP). OpenAIRE developed another implementation (at the time of this writing called FAIR DMP) that focuses on complementary aspects compared to the easy. DMP tool (referring to data sets from repositories the first, to workspaces, storage and archiving services the second). In the context of EOSC-hub, the FAIR DMP tool will be further developed in collaboration with OpenAIRE Advance (collaboration agreement joint activity JA1), according to the activities defined in the EOSC-hub - OpenAIRE Advance collaboration agreement. The aim is to offer the FAIR DMP as the unified DMP tool interface.

The openAIRE-advanced and EOSC-HUB teams have discussed the activity and described in a Joint Activity document the broad goals of the effort. The goal of the first 18 months will be to pilot the DMP tool with user communities. The team is currently detailing the steps necessary to integrate and deploy the DMP so it can be used by the targeted user communities.

The web-based interface and the back-end database holding the plans will continue to be developed within openAIRE and EOSC-HUB will concentrate on the external entity store and related services. Further services needed for integration with interested consumers of the data will be identified during the operation of the service and possibly developed and integrated, depending on how critical they are and on the availability of the resources.

9 Conclusions

This deliverable outlines the initial integration and maintenance plans of the EOSC-hub federation and collaboration services for the first year. These plans have been prepared during initial 4 months of the project and allow to track the integration activities in the work package 5 with aim to establish a uniform and scalable environment which facilitates an efficient federated service management, integration of thematic services in EOSC eco system and enables seamless access for the research communities and individual scientists to the scientific data and services distributed across multiple e-infrastructures.

While the maintenance of the services is mainly responsibility of the corresponding service owner, the integration activities require accurate coordination at task level, package level and also project level. Integration and maintenance activities and plans also incorporate the requirements and feature requests from user communities and subject of frequent adjustments and changes. The results and implementation of the proposed here plans will be reported in the next deliverable, which will also provide further integration plans for the next project phase.

10 References

No	Description/Link
R1	https://aai.egi.eu/registry/
R2	https://edugain.org/
R3	https://simplesamlphp.org/
R4	http://mitreid-connect.github.io/
R5	https://www.internet2.edu/products-services/trust-identity/comanage/#service-overview
R6	https://b2access.eudat.eu/home/
R7	https://www.eudat.eu/eudat-collaborative-data-infrastructure-cdi
R8	http://www.unity-idm.eu/
R9	http://perun.cesnet.cz/web/
R10	https://github.com/CESNET/perun
R11	https://www.egi.eu/federation/egi-federated-cloud/
R12	https://watts-dev.data.kit.edu/
R13	https://github.com/indigo-dc/wattson
R14	https://aarc-project.eu/
R15	https://wiki.nikhef.nl/grid/AARC_Pilot_-_Architecture#Detailed_Architecture
R16	https://refeds.org/sirtfi
R17	https://rcauth.eu/
R18	https://www.igtf.net/ap/iota/
R19	https://aarc-project.eu/wp-content/uploads/2017/11/AARC-JRA1.4A-201710.pdf
R20	https://wiki.refeds.org/display/GROUPS/Assurance+Working+Group
R21	https://wiki.geant.org/download/attachments/92573909/AARC-G021-Exchange-of-specific-assurance-information-between-Infrastructures.pdf
R22	https://aarc-project.eu/wp-content/uploads/2018/03/AARC-G041-Expression-of-REFEDS-RAF-assurance-components-for-social-media-accounts.pdf
R23	https://docs.google.com/document/d/1TMvHEGAi4jTrOQ_fyFmBhJTZJ_jJIzB5hUXcZZHSW3E/edit
R24	https://marketplace.eosc-hub.eu

R25	https://www.prestashop.com/en
R26	https://wiki.egi.eu/wiki/Applications_on_Demand_Service_-_architecture
R27	https://fitsm.itemo.org/
R28	https://vi-seem.eu/
R29	http://eosc.agora.grnet.gr
R30	https://eosc-hub.eu/catalogue/
R31	https://esoc-hub-devel.grnet.gr
R32	https://dp.eudat.eu
R33	https://github.com/EUDAT-DPMT
R34	https://wiki.egi.eu/wiki/GOCDB/PI/Technical_Documentation
R35	https://angular.io/
R36	https://spring.io/
R37	https://accounting.egi.eu
R38	https://accounting-devel-next.egi.cesga.es
R39	https://accounting-pre.egi.cesga.es
R40	https://github.com/ARGOeu/argo-ncg
R41	https://github.com/ARGOeu/argo-egi-connectors
R42	https://github.com/ARGOeu/argo-egi-consumer
R43	https://github.com/ARGOeu/argo-compute-engine
R44	https://github.com/ARGOeu/argo-web-api
R45	https://github.com/ARGOeu/argo-egi-web
R46	http://cds.cern.ch/record/1452920/files/GFD.201.pdf
R47	https://wiki.egi.eu/wiki/GGUS:SOAP_Interface_FAQ
R48	https://appdb.egi.eu/
R49	https://appdb-dev.marie.hellasgrid.gr/
R50	https://about.gitlab.com/
R51	https://about.gitlab.com/2016/05/23/gitlab-container-registry/
R52	https://gitlab.eudat.eu/users/sign_in
R53	http://repository.egi.eu/
R54	https://dashboard.appdb.egi.eu/vmops

R55	http://is.marie.hellasgrid.gr/
R56	https://appdb-dev.marie.hellasgrid.gr/browse/datasets