EOSC-hub

D6.4 Second report on the maintenance and integration of common services

Lead Partner:	MPG
Version:	v1
Status:	FINAL
Dissemination Level:	Public
Document Link:	https://documents.egi.eu/document/3643

Deliverable Abstract

This deliverable provides an overview of the maintenance and integration of common services in the areas of all tasks of WP6 'Common Services'. It comprises the description of the driving and demanding use cases, the maintenance and integration for each of the common services, the integration activities performed and future plans.

COPYRIGHT NOTICE

DELIVERY SLIP

Date	Name	Partner/Activity	Date
From:	Olivier Rouchon Michele Carpene Heinrich Widmann	CINES CINECA DKRZ	
Moderated by:	Małgorzata Krakowian	EGI Foundation	
Reviewed by:	Johannes Reetz Shaun de Witt	MPG/AMB UKAEA/PMB	
Approved by:	AMB		

DOCUMENT LOG

Issue	Date	Comment	Author
v0.1	18/11/2019	Google doc draft generated	Michele Carpene (CINECA)
v0.2	02/12/2019	Started first adoptions (from D6.2. to D6.4) and to new deliverable template	Heinrich Widmann (DKRZ)
v0.3	13/03/2020	Combined contributions from partners	Olivier Rouchon (CINES), Heinrich Widmann (DKRZ), Michele Carpene (CINECA), Mattia D'Antonio (CINECA), Claudia Martens (DKRZ), Enol Fernandez (EGI), Jorge Gomes (LIP), Harri Hirvonsalo (CSC), Andrei Tsaregorodtsev (CNRS), Yann Le Franc (eSDF), German Molto (UPV), Marica Antonacci (INFN), Bartosz Kryza (CYFRONET), Bartosz Wilk (CYFRONET), Pablo Orviz (IFCA), Tomasz Zok (PCSS), Lukasz Dutka (CYFRONET), Mikael Karlsson (CSC), Hans van Piggelen (SURFsara), Tobias Weigel (DKRZ).
v0.4	05/05/2020	Final preparation for Review	John Kennedy(MPCDF), Olivier Rouchon(CINES)
v1	16/07/2020	Reviewer comment/suggestions incorporated	Reviewers: Johannes Reetz(MPCDF), Shaun de Witt(UKAEA).

TERMINOLOGY

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

Terminology/Acronym	Definition
API	Application Programming Interface
CKAN	Comprehensive Knowledge Archive Network
CMD	Cloud Middleware Distribution
DOI	Digital Object Identifier
IaaS	Infrastructure as a Service
IdP	Identity Provider
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
PaaS	Platform as a Service
CaaS	Computing as a Service
PAM	Pluggable Authentication Module
PID	Persistent Identifier
RCD	Research Community Dashboard
REST	REpresentational State Transfer
SAML	Security Assertion Markup Language
VM	Virtual Machine
VO	Virtual Organization
WebDAV	Web Distributed Authoring and Versioning
WMS	Workload Management System
YAML	Yet Another Markup Language

Contents

Executive summary

This document contains a report on the work plan of integration and maintenance of common services of the EOSC Hub Service Catalogue [1]for the second year of the project. The work described within this document, focusing on service provisioning and integration contributes to the EOSC-hub Key Exploitable Results 4,5 and 8.

To meet the needs of users, we first analyse the use cases to identify the requirements for the integration of the common services. In addition, the individual services are examined for their supported protocols, interfaces, and potentially useful combinations with other services. Based on this assessment, the work plan was created, identifying, and describing the integration activities that have been carried out or are planned. The document covers several categories, reflecting the different tasks of WP6 i.e. Discovery and Access (T6.1), Federated Compute (T6.2), Processing and Orchestration (T6.3), Data and Metadata Management (T6.4) and Sensitive Data (T6.6). The task Preservation (T6.5) is not included as its integration with other services has not started yet. Where relevant, information on plans in the scope of EOSC-hub project is included

---

[1] For detailed description of the common services see the report of deliverable D6.1 First release of common services software
(https://confluence.egi.eu/display/EOSC/D6.1%09First+release+of+common+services+software)

1   Introduction

This document presents a report on the integration and maintenance of EOSC-hub services, focusing on the driving use cases, and the status and progress of integration activities.

The coordination of integration and maintenance activities of these services is the main goal of WP6 and is described in this document for the areas of data discovery, access and management, federated computing, and orchestration.

The main effort in the second year of the project was focused on solving the existing issues related to integration while simultaneously responding to the community requirements for new integration activities. Thus, this deliverable provides an overview of the continued integration activities in the second year of the EOSC-hub project along with description of developments and improvements planned for the remaining part of the project. Underpinning the integration activities are the needs to ensure that community use-cases are promptly addressed while also ensuring that integration activities are undertaken in a manner that ensures EOSC achieves a high level of service interoperability.

The document is organized as follows. Since the needs and applications of the users lead to the requirements for the integration of the EOSC services, we put the description of the driving use cases at the beginning, see section 2. The following paragraphs are dedicated to the individual WP6 task areas, covering 'Discovery and Access' in section 3, 'Federated Compute' including 'Sensitive Data' in section 4 , 'Processing and Orchestration' in section 4, and 'Data and Metadata Management' in section 5. Each of these area sections is in turn subdivided in subsections 'Maintenance, interfaces and integration options' explained for each service and 'Integration activities', reporting on the work done within the second project year, 'Future Integration Plans' and optionally 'Justification for delay'. The report concludes with a section 'Summary and Outlook'.

2 Use cases

The following use cases provided requirements for the common services. Those requirements have been collected according to a common template and stored in the EOSC-hub wiki (https://wiki.eosc-hub.eu/display/EOSC/Community+requirements+DB). Then the WP10 has analyzed them and helped the communities to talk with the WP6 developers, organizing meetings and opening tickets on the EOSC JIRA system.

The use cases related to data management activities are collected from mainly five sources:

- Thematic Services.
- Competence Centers.
- Communities already using EUDAT/EGI/Indigo services.
- New communities entering EOSC.
- Low hanging fruits identified by EOSC-hub service providers about the integration among common or federated services.

This last point is the only one not directly related to user requirements. It encompasses integration activities, which can offer new features, like the interoperability between two data services, potentially interesting for the users and achievable by the service developers with a limited effort.

2.1 ECAS : Perform analysis on remote large volume climate data

The ENES Climate Analytics Service (ECAS) offers scientific users a set of tools to perform data analysis experiments on large volumes of multidimensional data, using parallel processing workflows on remote systems without needing to download data.

2.1.1 Integration of B2DROP to store and exchange output of ECAS output

B2DROP can be used within different parts of ECAS.

The first integration aspect of ECAS with B2DROP concerns the Ophidia workflow framework. The framework was extended by a custom operator which stores the workflow output in the B2DROP account of the user. To use this operator, the user creates an app password within B2DROP and stores this in the ECAS user space. The credentials and the files to be uploaded to B2DROP are then configured as part of calling the operator. To upload the files to the B2DROP space of the user, the B2DROP space is mounted locally using the WebDAV protocol.

The second part of ECAS that is integrated with B2DROP is JupyterHub. JupyterHub is a web-based framework for execution of Jupyter notebooks on remote resources. The deployment of JupyterHub within ECAS was configured and extended to access two different kinds of B2DROP storage. The first one is a shared space for all ECAS users and does not need further authentication of the users, thus facilitating easy exchange of files between ECAS users. The second storage is the user's private B2DROP space. This space is only visible and accessible for the owner after authentication. To use the private B2DROP space, the user creates an app password within B2DROP and stores the credentials in the environment file of the notebooks. Akin to the Ophidia operator, the B2DROP space is mounted locally via the WebDAV protocol. For the usage of the B2DROP space the graphical

interface was extended with two buttons. The buttons are called "Share" and "Move". The "Share" button copies the notebook to the shared space and the notebook is shared with all other ECAS users. The "Move" button copies the notebook to the private or the shared B2DROP space, specified by the user. Instead of moving the notebooks, the users can create them directly within the B2DROP space, too.

For more information see deliverable D7.2 chapter 4.

Service: B2DROP, B2ACCESS and IAM

Resource Providers: DKRZ and CMCC

Resources: allocation of B2DROP storage depending on use case

Use case requirements:

- B2DROP account (optional, only if sharing with otherECAS users is desired)
- Input data must reside in any data source supported by ECAS, e.g., B2DROP or community store (OpenDAP)
- Output data must not exceed user quota of B2DROP

2.1.2 Integration of EGI FedCloud resources with ECAS

The EC3 AoD platform has been extended to enable exploiting the EGI Cloud Compute service to deploy on demand ECAS elastic clusters. In that way a user can deploy an ECAS cluster following a simple wizard where the main parameters (CPU and memory of the nodes, number of nodes of the cluster, etc.) can be selected. Finally, the actual interaction with the infrastructure is delegated to the Infrastructure Manager (IM).

The integration of ECAS in the cloud-based resources provided by EGI allows users to easily deploy a full ECAS elastic cluster (composed of multiple nodes) in the cloud resources of the EGI Federation customized to their requirements. The EC3 service will take care of automatically installing and configuring the whole ECAS environment stack, including services and tools such as JupyterHub, PyOphidia, a rich set of data science Python libraries, the Ophidia HPDA framework, as well as a comprehensive set of Jupyter Notebooks for training. Furthermore, the internal elasticity manager (CLUES) automatically grows or shrinks the size of the cluster based on its workload.

Service: EC3, IM and FedCloud

Resource Providers: CMCC and FedCloud partner

Resources:

- Allocation of a set of VMs (from 2 to 12) with at least 2 CPUs and 4 GB or RAM in FedCloud sites.

Use case requirements:
- Launch ECAS virtual elastic cluster over FedCloud resources

2.2 Marine use cases

The Marine CC shows interest in the integration of B2DROP and B2STAGE for the two use cases described below. The use cases are not dependent on each other, but rather complementary.

2.2.1 Processing measurement data and share processed data for collaborative analysis.

Regarding measurement data, coming from Argo floats. To establish a workflow consisting of the following phases: upload raw data, process the raw data, generate processed data, upload processed data, collaborate on the analysis of the processed data, possibly publish the final results and reports in relevant formats for open access.

The raw data is incrementally uploaded once a day, consisting of both new and corrected/curated data. Therefore, the raw data needs to be processed daily to distinguish any difference to the processed data. The range of the incremental raw data is one calendar month. The size of the monthly raw data is ~ 2 GB. The size of total raw data is ~ 300 GB.

Processing of the raw data is a time-consuming event and should be carried out in a batch or asynchronous manner. Therefore, the CC would like to have a notification when the process has finished.

The identified components for enabling these steps are: B2STAGE for uploading/transferring raw data to storage, B2SAFE for storage of raw data, Apache Spark for data intensive computation tasks, FedCloud for running the compute and analysis applications, B2DROP for collaborative analysis on shared data , Jupyter Notebooks for interactive analysis, B2ACCESS for authentication and B2SHARE for publishing the final result data and associated publications and reports.

Service: B2STAGE, B2SAFE, FedCloud, B2DROP, B2ACCESS and B2SHARE

Resource Providers: CSC, CINECA, Jülich and FedCloud partner

Resources:

- Allocation of 300 GB storage in B2SAFE
- Processed data to share must not exceed user quota of B2DROP

Use case requirements:

- B2DROP and B2SHARE accounts
- Output data must not exceed user's quota on B2DROP
- Optional: Notification of user when processing finished

2.2.2 User applications in a Virtual Research Environment

Regarding a virtual research environment to establish a web-based platform able to host a range of scientific applications. The application instances are launched per user on user demand. The scientific applications could be used to e.g. analyse the processed data in "*Use case 1*". Some applications are single-component, others are client-server where the server could be shared among users with the client being per user. Many of the applications are memory-bound, some requiring up to 8 CPU and 16 GB RAM per user. Yet other applications require as little as 1 CPU and 4 GB RAM. Most require no GPU-capabilities. The analysis is often interactive serial, rarely batch

parallel, therefore traditional HPC/HTC computing cluster are not relevant. The duration of the sessions/runs are often not known beforehand and the sessions should not be killed pre-emptively losing user data. Cloud-provided resources are most suitable for this kind of dynamic/elastic purposes.

The user's saved data should be accessible across applications in near real-time, as well as accessible from a central user interface for managing.

The identified components for enabling these could be: B2ACCESS for authentication, B2DROP for syncing the data between applications and collaborating on, FedCloud for providing the VRE infrastructure, Kubernetes for orchestrating the VRE system and application containers, and Jupyter Notebooks for common and interactive analysis.

For more information see deliverable D7.2 chapter 4.

Service: B2DROP, FedCloud, Kubernetes and B2ACCESS

Resource Providers: CSC, CINECA, Jülich and FedCloud partner

Resources: allocation of memory and compute power, depending on the application up to 8 CPUs and 4 GB RAM

Use case requirements:

- B2DROP account
- Data must be accessible across applications via a central user interface
- Output data must not exceed user's quota on B2DROP

2.3 ICEDIG/Herbadrop use case: Digitisation infrastructure test on EUDAT

 The Herbadrop use case comes from a data pilot in the EUDAT project aiming at 'an innovative approach to long-term preservation and analysis of digitised herbarium specimens'[2] and is now being treated in the EU-funded project ICEDIG[3]. The project milestone 'Digitisation infrastructure test on EUDAT' is described in detail in the pdf in the Appendix[4].

Herbadrop's archive comprises 27 TB of data volume on the B2SAFE instance at CINES. The objective of the ICEDIG data pilot is to develop the premise of the future ETDR (long-term European certified Trustworthy Digital Repository), in which CINES is involved. Thanks to services such as B2FIND or community portals in interaction with ETDR, FAIR data which is preserved and curated into the ETDR

---

[2]   https://www.eudat.eu/herbadrop-an-innovative-approach-to-long-term-preservation-and-analysis-of-digitised-herbarium
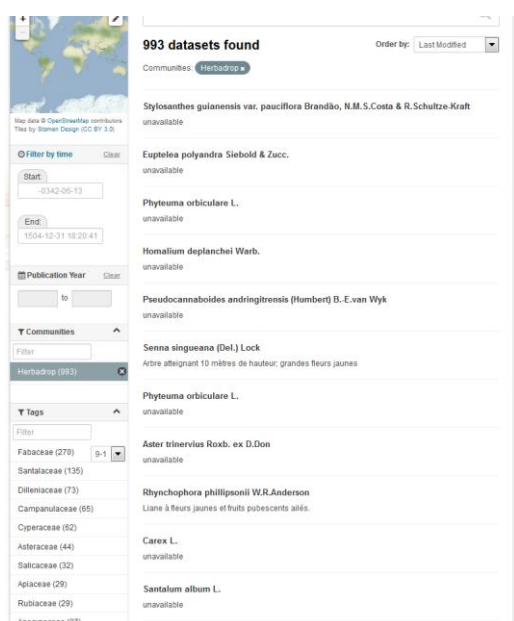
[3] ICEDIG stands for "Innovation and consolidation for large scale digitisation of natural heritage" https://www.icedig.eu/

[4] Appendix: Milestone MS39_ ICEDIG_Digitisation infrastructure test on EUDAT_v1.pdf, see at https://wiki.eosc-hub.eu/download/attachments/26416995/Milestone%20MS39_%20ICEDIG_Digitisation_infrastructure_test_on_EUDAT_v1.pdf?api=v2

infrastructure would be accessible for non-profit users in CINES open data portal as well as it would be searchable and accessible via EUDAT B2FIND.

The ICEDIG architecture is split into a sequence of functions that processes one-step of the workflow. The image replication operation uses the EUDAT B2SAFE service. B2HANDLE is required for PID (Persistent Identifier) generation and then to guarantee data access through the B2FIND portal. The B2FIND portal and API provide users with advanced search functionalities and allow access to the data resources associated to the metadata found in the catalogue. EUDAT retrieves the metadata in Elasticsearch with our HTTP-API and feeds in pull mode the B2FIND portal for each of the images of seagrass deposited on the ICEDIG platform. The access to data is then made possible through a webdav proceeding without authentication on the iRODS data node at CINES.

Herbadrop is visible in B2FIND as a Community, which means that a search request may start with showing all records that are offered by Herbadrop. To narrow down a search, e.g. for certain species, the facet <Tags> may be used (figure 1). The detailed search result page offers a direct link to the institution maintaining the digital objects as visualised in figures 2 and 3.

Service: B2SAFE, B2HANDLE, B2FIND

Resource Providers: CINES

Resources:

- Allocation of 27 TB of data volume on the B2SAFE instance at CINES
- B2HANDLE prefix to register PIDs

Use case requirements:

- B2SAFE instance at CINES
- Access to Elasticsearch repository via HTTP-API by B2FIND
- Workflow for generation of PIDs and WebDav-URLs refering the iRODs collections

2.4 WeNMR use case

WeNMR is a worldwide e-Infrastructure for NMR spectroscopy and Structural biology. It is the largest Virtual Organization in the Life sciences and is supported by EGI.

Through integration with EGI DataHub, WeNMR users will be able to access a data space provisioned through EGI DataHub and its underlying platform Onedata, through the West-Life Virtual Folder. "Virtual Folder" provides a unified access mechanism to files stored in a variety of locations including the local file system and cloud storage facilities.

Oneprovider enables several means for integration with other services including: REST API, CDMI (Cloud Data Management Interface) and POSIX. The integration with Virtual Folder will be based on the POSIX Fuse mountpoint enabled by Oneclient command line tool.

Service: EGI-DataHub, "Virtual Folder"

Resource Providers: EGI, CYFRONET

Resources:

- Onezone and Oneprovider EGI-DataHub instances deployed at CYFRONET

Use case requirements:

- POSIX Fuse mount point enabled by Oneclient
- Transparent POSIX access to files on remote storages

2.5 CompBioMed data replication use case

CompBioMed is a European commission H2020 funded Centre of Excellence focused on the use and development of computational methods for biomedical application. The data-intensive workflows and distributed international partners involved in the project urges the use of proper data management solutions for handling the data. Safe data replication and large data transfer is one of the major requirements within the community. In the past months, we have been working on a use case to replicate data from BSC (Barcelona Supercomputing Centre) to SURFsara (Netherlands) and EPCC (UK) using the EUDAT B2SAFE service. Once the replication service is setup and configured, we expect to replicate terabytes of data between the HPC centers which facilitates large data exchange and access to valuable data for researchers in this community.

Service: EUDAT B2SAFE service

Resource Providers: BSC, SURFsara, EPCC

Resources: allocation of at least 24 TB storage at each of the HPC centers

Use case requirements:

- Data to be replicated is 3D finite element mesh (file format can be. vtk, .txt).
- The maximum size per file is 1.2 TB.
- The total data to be replicated is 24 TB.
- Two copies of replicas are desired, one on the compute facilities to run simulations and one on tape.
- The data owner assesses the replicas.
- Data will be downloaded by researchers
- Full access control to the data (i.e. read/write/Exec access)
- Data needs to be findable Potentially after publication and/or after the 3-year quarantine

2.6 DODAS use case

The Dynamic On Demand Analysis Services (DODAS) is an open-source Platform-as-a-Service tool, developed and maintained by INFN, which allows to deploy software applications over heterogeneous and hybrid clouds. DODAS completely automates the process of provisioning, creating, managing and accessing a pool of heterogeneous computing and storage resources, thus drastically reducing the learning curve, as well as the operational cost of managing community-specific services running on distributed clouds. DODAS currently supports the on-demand deployment of:

- Batch system as a Service instances based on the HTCondor technology;
- Big Data analysis platforms providing Machine Learning as a service;

DODAS has already been integrated into the submission Infrastructure of the Compact Muon Solenoid (CMS), one of the two biggest and general purpose experiments at the CERN Large Hadron Collider (LHC), and into the Alpha Magnetic Spectrometer (AMS-02), an experiment hosted on the International Space Station, data analysis workflow.

One of the main architectural goals of DODAS Thematic Service is to provide a high level of modularity, a key to a generic applicability.

Being modular, the architecture provides the ability to easily customize the workflow depending on the community computational requirements. In this context the major EOSC-hub services adopted are:

- The PaaS Orchestrator which has the role of taking the requests related to application or service deployment coming from the user expressed using TOSCA, the OASIS standard to specify the topology of services provisioned in IT infrastructures. Based on the user requirements (typically expressed in the TOSCA template), the Orchestrator has the role to identify the best infrastructure (IaaS) for the deployment taking into account information about user's SLAs the availability and the health status of the IaaS services.
- The actual interaction with the infrastructure is delegated to the Infrastructure Manager (IM). This service is a key in the architecture as it is in charge to deploy complex and customized virtual infrastructures on different IaaS Cloud deployment, providing an abstraction layer to define and provision resources in different clouds and virtualization platforms. From the integration perspectives the TOSCA support provided by IM represent a key feature. Moreover, it eases the access and the usability of IaaS clouds by automating the VMI (Virtual Machine Image) provisioning including selection, deployment, configuration, software installation.
- The glue of the implemented flow is the Identity and Access Management service (IAM). IAM provides a layer where identities, enrolment, group membership, attributes and policies to access distributed resources, and mostly supports the federated authentication mechanisms. Identity and Access Management is provided through multiple methods (SAML, OpenID Connect and X.509) by leveraging on the credentials provided by the existing Identity Federations (i.e. IDEM, eduGAIN, EGI CheckIN). ESACO service is also part of the DODAS integrated service and this is responsible for guaranteeing Cloud providers (such as Openstack based providers) with support of multiple OAuth2 Authorization Servers.
- The support to Distributed Authorization Policies and Token Translation Service in DODAS is implemented thanks to the WaTTS service (https://github.com/indigo-dc/tts) which guarantees selected access to the resources as well as data protection and privacy.

Service: Indigo-IAM, PaaS Orchestrator, WaTTS

Resource Providers: INFN

Resources:

- Deployment of a dedicated IAM and WaTTS instance:

dodas-iam.cloud.cnaf.infn.it

dodas-tts.cloud.cnaf.infn.it

Use case requirements:

- Dedicated instances of security services (IAM and WaTTS)

2.7 DARIAH use case

The DARIAH (Digital Research Infrastructure for the Arts and Humanities) Thematic Service (TS) aims to enhance and improve the usage of the cloud-based services and technologies in the domain of the digital arts and humanities research. It will enable end-users coming from the digital arts and humanities domains to seamlessly store, describe (metadata) and share their datasets, discover, browse and reuse datasets shared by the others and to perform analysis on various data volumes.

The DARIAH TS is providing a set of services and in particular, among them, the "Invenio-based repository as a service" enables researchers and scholars to easily create, deploy and configure their own Invenio-based repository and host it on cloud infrastructures.

The service is built around a set of EOSC-hub services:

- The FutureGateway that provides a user-friendly web interface for requesting the deployment of the repository: the authenticated user can customize the deployment request using a simple form; through the web interface it is also possible to monitor the status of the deployment and get the endpoint to access the deployed system.
- The PaaS Orchestrator receives the deployment request submitted by the users through the FutureGateway and coordinates the provisioning and configuration of the needed cloud resources on the "best" cloud provider. The latter is selected taking into account information such as the SLAs signed with the users, the monitoring data about the health of the provider services.
- The Infrastructure Manager (IM) is steered by the Orchestrator to interact with the cloud sites (through the APIs provided by the different Cloud Management Frameworks) in order to provision the virtual resources (servers, block devices, etc.) needed by the deployment. The contextualization of the virtual machines is managed by IM as well exploiting ansible to automate the installation and configuration of the software components.
- The INDIGO IAM provides the authentication/authorization infrastructure: OIDC tokens issued by IAM are used to access and interact with the PaaS services and also with the cloud providers.

Service: FutureGateway, PaaS Orchestrator, Infrastructure Manager (IM), Indigo-IAM

Use case requirements:

- Automated deployment of Invenio-based repository on Cloud environment

3   Discovery and Access

The overall objective of the task T6.1 'Data Discovery and Access' is the establishment of the *Common Discovery and Access Interoperability Layer* through which end-users can find, localize, transfer and re-use data resources within EOSC-hub for their own scientific purposes.
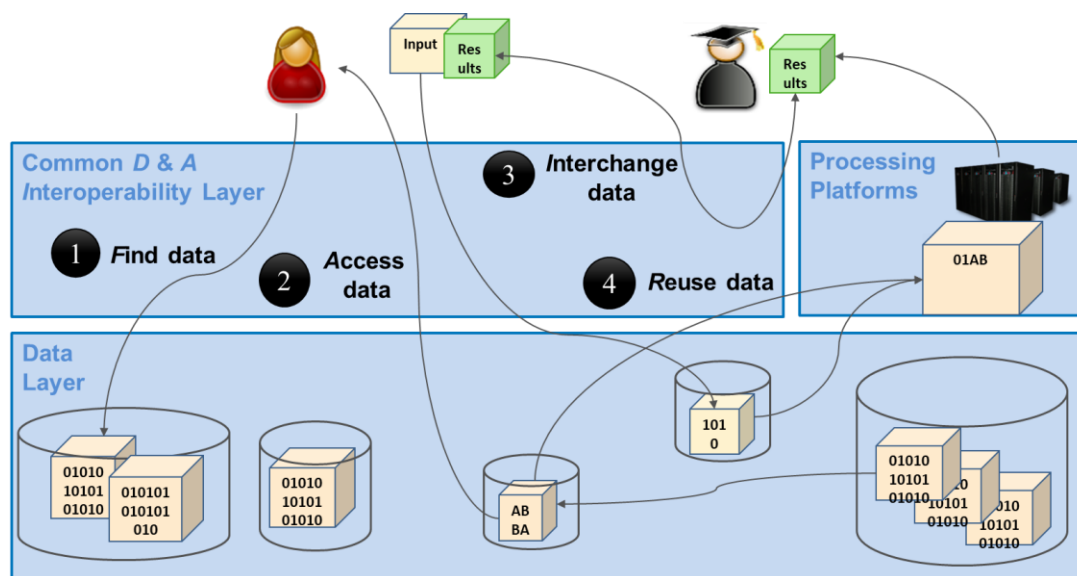
Figure 1. The Common Discovery & Access Interoperability Layer enabling FAIR data management

 As shown in figure 1 above, end-users should be enabled to manage and use research data in a FAIR way, which means in particular:

1. [F] search for distributed data in EOSC-hub and beyond
2. [A] seamless access to distributed data resources wherever they are located
3. [I] interoperable sharing and publishing of research output following open standards and domain agnostic tools and interfaces
4. [R] reuse, exchange and staging of data

Regarding data discovery the metadata service B2FIND is intended to play the role of the central search indexing tool of EOSC-hub. For this the service is extended and enhanced to cover data as well from storage services as EGI-datahub and B2SAFE and data archives within and beyond EOSC-hub. Regarding data access we elaborated and established seamless data access and transfer in and between the storage services EGI datahub and B2SAFE. Hereby the B2STAGE service and AAI proxies and tools from WP5 ('Federation Services') supports the data transfer between EGI-datahub and B2SAFE. Finally, B2DROP serves to easily exchange data with other researchers and to keep it synchronized and up to date.

The realisation and implementation of the Common Discovery and Access Interoperability Layer follows a roadmap comprising workpage activities (WPAs) describing the integration of pairs of common services from tasks from T6.1, T6.4 and WP5:

- WPA 6.1.3 & 4: Integration of EGI Datahub with B2FIND (indexing and discovery of EGI data resources)
- WPA 6.1.5: Get B2SAFE data collections indexed by B2FIND (see e.g. Herbadrop use case)
- WPA 6.1.6: Interoperability between EGI datahub and B2SAFE (data transfer between storage media of both services))
- WPA 6.1.7: B2STAGE integration with B2SHARE (retrieve processed data and store in B2SHARE)
- WPA 6.1.8: B2STAGE integration with B2DROP (prepare input data for B2STAGE / retrieve and store - small sized - data)
- WPA 6.1.9: B2DROP integration with EGI datahub

To achieve this, we defined an integration plan which identifies and specifies the work plan activities WPA 6.1.N in more detail. An overview of this integration plan is shown in the figure 2
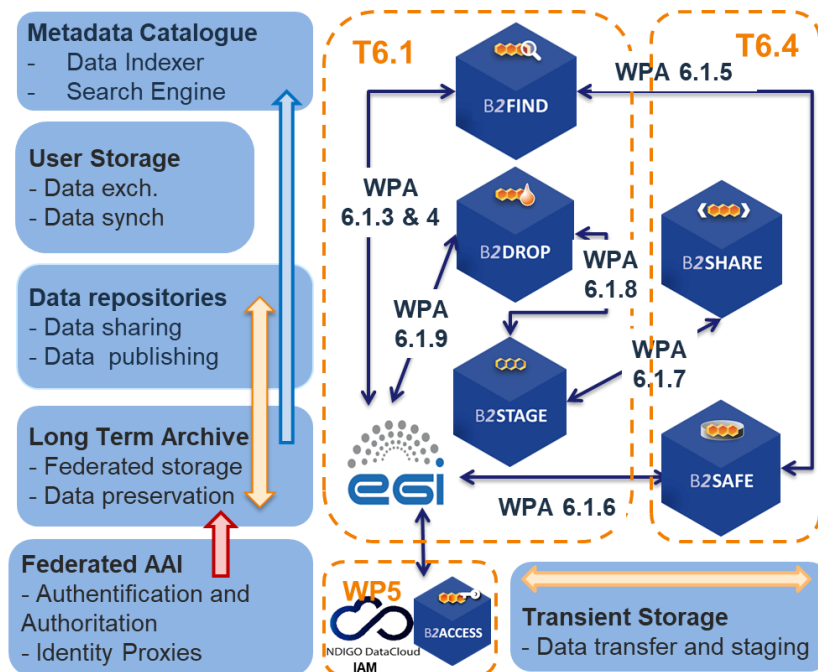
Figure 2. The updated integration plan of T6.1 - including services from T6.4 and AAI tools from WP5 - with the associated common services and Work Package Activities (WPA6.1.N)

The figure includes not only the integration of common services from T6.1, but as well the storage services B2SHARE and B2SAFE from T6.4 'Data and Metadata management' and the AAI tools from WP5. The latter are important for seamless authentication and authorisation for service and data access.

There are two additional WPAs that go beyond the common services of WP6 and are related to the cooperation with OpenAire:

- WPA 6.1.11: Integrate EGI DataHub with OpenAIRE Research Community Dashboard by adapting to OpenAIRE guidelines

- WPA 6.1.12: Integrate EUDAT B2FIND/B2SHARE with OpenAIRE Research Community Dashboard by adapting to OpenAIRE guidelines for data providers

In this report the progress and status of the WPAs are treated in detail in the section Integration Activities.

3.1 Maintenance, interfaces, and integration options of the services

We describe here for each service the work done w.r.t. maintenance like performed updates and upgrades of the software, the available interfaces like APIs and protocols used and the potential capabilities for integration with other services of the EOSC service catalogue.

3.1.1 B2FIND

B2FIND (http://b2find.eudat.eu) is an interdisciplinary discovery portal for research data that are stored within EOSC-hub and beyond. Therefore, metadata collected from heterogeneous sources are indexed in a comprehensive joint metadata catalogue and made searchable via an openly accessible web interface. B2FIND provides transparent access to the scientific data objects through the given references and identifiers in the metadata, thus supporting (at least) the first two pillows of FAIR data principles. For detailed description of the service see D6.1.

3.1.1.1 Maintenance

B2FIND maintenance includes both technical and content related issues. For ongoing Community integration, a typical uptake workflow consists of a test integration where metadata records from either a community specific repository or any other offered endpoint are harvested, mapped and uploaded to a test machine. As the semantic mapping of non-uniform, community specific metadata to homogenous structured datasets is a most subtle and challenging task, the mapping process is accompanied by iterative and intense exchange with community representatives and usage of controlled vocabularies and community specific ontologies in order to assure and improve metadata quality. Therefore, a new B2FIND_Community_template for Community communication has been created as well as new templates for several metadata prefixes that are supported (datacite3.1, datacite4.0, Dublincore, iso19139, ddi, cmdi, json-config).

As a prerequisite for an enhanced metadata ingestion workflow several changes have been made within the software stack, e.g. in order to simplify package management and deployment with conda, Miniconda was installed on all machines and the whole code was updated to Python 3.6. Currently some effort is put in the development of a generic Metadata Schema for EUDAT, which implies a complete revision of all communities already implemented in B2FIND but may lead to further harmonization and (hopefully) standardization of metadata records.

Concerning the common Classification for Research Areas[5], we are currently waiting for the Collaboration Partner to overcome some administrative obstacles in order to begin working on the content. clara.science was already presented (e.g. at RDA 13th Plenary) and has attracted great attention, which is reflected in several inquiries about possible participation and cooperation.

The integration of scientific communities is ongoing; both in terms of metadata ingestion from data providers and publishers as well as in terms of an integration of research infrastructures (such as ENVRIplus and PaNOSC) and other EOSC related projects, such as EOSC Nordic, EOSC Enhance and EOSC Pillar.

3.1.1.2 Service Interfaces

As the discovery portal is openly accessible there is no need for AAI integration.

B2FIND offers Guidelines for Data Providers, including research data management recommendations, references to FAIR data principles and technical requirements concerning harvesting methods as well as advices for aggregation levels and metadata quality in general:

- http://b2find.eudat.eu/guidelines/introduction.html
- http://b2find.eudat.eu/guidelines/providing.html
- http://b2find.eudat.eu/guidelines/harvesting.html

B2FIND offers a training module in GitHub:

- https://github.com/EUDAT-Training/B2FIND-Training

All B2FIND code is openly accessible and reusable in GitHub:

- https://github.com/EUDAT-B2FIND

3.1.1.3 Possible Integration Partner Services

B2FIND is integrated with B2SHARE as an incrementally harvesting daily for records in B2SHARE is implemented.

B2FIND is integrated with EGI DataHub via its OAI-PMH endpoint.

B2FIND can be integrated with B2SAFE as shown in Herbadrop Use Case.

3.1.2 EGI DataHub

EGI DataHub (https://datahub.egi.eu) is a service for provisioning large reference open data sets, based on Onedata distributed virtual filesystem platform, available to end users over standard POSIX interface. For detailed description of the service see [D6.1] or refer to Onedata documentation at https://onedata.org.

3.1.2.1 Maintenance

EGI DataHub maintenance includes scheduled service upgrades to the latest versions of Onedata, integration of external storage providers and fixing irregularities in service provisioning. During the reported period, the EGI DataHub has been upgraded 3 times to the consecutive stable Onedata versions: 19.0.1, 19.02.2 and 19.02.3. There was also one major unscheduled downtime in May 2020 related to a serious Cloud storage failure at Cyfronet, which required recovery of parts of database from backups, and caused a 2-day downtime of the service.

Furthermore, a cleanup of the open data set records published through the EGI DataHub's OAI-PMH interface has been performed, removing some redundant (mostly test data sets generated during

various demonstrations and tutorials), in order to provide more relevant search results in services which harvest metadata about open data sets from EGI DataHub (such as B2FIND).

3.1.2.2   Service interfaces

For service management and integration Onedata provides comprehensive REST API for each of its constituting services:

- **Onezone** - https://onedata.org/#/home/api/latest/onezone
- **Oneprovider** - https://onedata.org/#/home/api/latest/oneprovider
- **Onepanel** - https://onedata.org/#/home/api/latest/onepanel

For data access Onedata provides 3 main options:

- **POSIX** - available using Oneclient command line tool, which creates a mount point with a virtual filesystem based on data accessible to a given user
- **CDMI** - standard HTTP data access and management interface (https://www.snia.org/cdmi)
- **Web GUI** - easy to use web graphical interface enabling basic uploading and downloading of files

3.1.2.3   Integration Partner Services

EGI DataHub is integrated with B2FIND through its OAI-PMH endpoint, which exposes the metadata of published open data sets in EGI DataHub.

EGI DataHub is integrated with B2HANDLE enabling users to automatically mint a PID handle while publishing an open data set.

EGI DataHub is indirectly integrated with B2ACCESS, as it already supports login via EGI CheckIn which is integrated with B2ACCESS.

EGI DataHub is integrated with B2SAFE via WebDAV protocol enabling data transfers between the EGI and EUDAT users[6].

3.1.3   B2STAGE

B2STAGE is a suite of services aimed to transfer data into and out of EUDAT data nodes and exposes three protocols for data staging: GridFTP, WebDAV and HTTP-API. For detailed description of the service see D6.1, https://github.com/EUDAT-B2STAGE/B2STAGE-GridFTP and https://github.com/EUDAT-B2STAGE/http-api

3.1.3.1   Maintenance

B2STAGE maintenance includes periodic upgrades (every two to three months) of the underlying components, in particular the PostgreSQL database, the NGINX reverse proxy and the Flask server. During the current reporting period we had 5 new version of B2STAGE: 1.0.7, 1.0.8, 1.1.0, 1.1.1 and 1.1.2 (currently under development). PostgreSQL has been upgraded from version 10.7 (B2STAGE 1.0.7) to version 11.4 (with 1.0.8) to version 12.1 (B2STAGE 1.1.1). NGINX has been upgraded from version 1.15 to version 1.17 (with B2STAGE 1.0.8) to version 1.18 (with B2STAGE 1.1.2). Flask has

---

[6] https://docs.google.com/document/d/1UhnY-aqlMqVlxRk4eC0KY2aKlGMdxmq5qiuFlGRlPBk

been upgraded from version 1.0.2 to version 1.1.1 (with B2STAGE 1.0.8) to version 1.1.2 (with B2STAGE 1.1.2).

3.1.3.2 Service Interfaces

GridFTP (via the EUDAT Data Storage Interface) is a service aimed at large data transfer and a large number of files between HPC centers and EUDAT in order to store them, process them and, possibly, move the results back. GridFTP standard protocol has been extended by implementing a GridFTP iRODS Data Storage Interface (DSI), a set of C functions which can interact with iRODS through the iRODS C API. The main supported operations are get, put, delete and list. GridFTP iRODS DSI is available at https://github.com/EUDAT-B2STAGE/B2STAGE-GridFTP

WebDAV (Web Distributed Authoring and Versioning) is an HTTP extension supporting remote content editing. By leveraging WebDAV a common web server can be used as a file server, allowing authors to create, move, delete or copy files and folders. WebDAV is fully supporting dataobjects and collections from iRODS and thus is well integrated with B2SAFE instances.

The HTTP APIs service is a set of RESTful endpoints allowing the access to both data and metadata stored in a B2SAFE instance. This service is aimed for small to medium datasets and it offers programmatic access to data and thus allows for smooth integration of such data into other applications and data services. The primary goal is to allow users to ingest and retrieve data via a standard RESTful HTTP interface in order to hide the underlying technology from users, lower the entry barrier to using EUDAT services, simplify integration into existing workflows, allow direct access to data assets held with the EUDAT CDI. APIs implementation is available at https://github.com/EUDAT-B2STAGE/http-api

3.1.3.3 Possible Integration Partner Services

B2STAGE is integrated with EOSC-hub Service B2SAFE by supporting all the necessary authentication protocols (native, GSI and PAM). B2STAGE can expose data stored into B2SAFE by both referring to PIDs and data paths.

B2STAGE is integrated with EOSC-hub Service B2ACCESS by implementing the full OAuth2 authorization protocol and managing both access and refresh tokens provided by B2ACCESS.

B2STAGE is being integrated with EOSC-Hub Service B2SHARE by using B2ACCESS as a common authentication layer. This integration enables users of EOSC-Hub Service B2SHARE to access content of datasets, where metadata of the dataset is stored in EOSC-Hub Service B2SHARE and files of the dataset are stored in EOSC-Hub Service B2SAFE. B2STAGE is used as an adapter for transferring files from B2SAFE to user's computer.

3.1.4 B2DROP

B2DROP is a Sync & Share service offering researchers an easy way for collaborative working on documents and synchronisation of data across multiple devices. Beside of the common functionality of sync and share services, B2DROP is connected to other services, such as EUDAT B2SHARE or CLARIN Switchboard, and offers a one click file transfer to these services. For detailed description of the service see D6.1. and https://eudat.eu/services/userdoc/b2drop

3.1.4.1   Maintenance

The reporting period contains four big maintenances which contained a downtime of the service. There was one maintenance per quarter. Within each maintenance we updated the underlying software Nextcloud to the next major release and performed updates of the used database schema. During the whole reporting period we updated Nextcloud from version 13 to version 17. In the second maintenance we updated the whole software stack which includes the update of PHP from version 5 to version 7.2.

Beside the software updates, additional features within B2DROP were enabled. The announcement center allows an easy notification to all users about upcoming maintenance, planned service downtimes, temporary constraints or other services belongings. The right click option allows users to open the files context menu with a click on the right mouse button like it is in the normal desktop environment. The circles extension allows the users to create their own groups and share content with the groups. This simplifies recurring sharing with the same user group.

The B2SHAREbridge was enhanced in a more meaningful wording. Additionally, the restricted access of B2SHARE communities is already shown in the community selection within B2DROP. This extension should highlight the restriction before users try to create a B2SHARE record from B2DROP inside a restricted community.

3.1.4.2   Service Interfaces

● Web-frontend for interactive use, WebDAV API for Clients and connected services

3.1.4.3   Possible Integration Partner Services

● B2DROP is integrated with EOSC-hub Service B2ACCESS.
● B2DROP is integrated with EOSC-hub Service B2SHARE.
● B2DROP can be integrated with EOSC-hub Service CLARIN Switchboard.

3.2 Integration activities

This section presents the overview of new or improved features achieved by extending or integrating existing services, and their relevance for thematic and specialized services.

3.2.1   Discoverability of EGI DataHub datasets via B2FIND

EGI DataHub and B2FIND services have been integrated by means of the OAI-PMH endpoint exposed by EGI DataHub (http://datahub.egi.eu/oai_pmh?verb=ListRecords&metadataPrefix=oai_dc), which exposes metadata of all published open data sets in EGI DataHub.

From a user perspective, this works in the following way. In order to publish a dataset, users have to create a share from their selected directory in EGI DataHub. The share by default is not public but can be accessed using a public URL endpoint in the EGI DataHub. Once the share is created, users have the option to publish it as an open data set. This step requires that the user selects a handle registration service and provides relevant metadata in Dublin Core format. For the EOSC-hub users, EGI DataHub has been integrated with B2HANDLE, thus registration of PID handles is

automated. During publishing of the dataset using EGI DataHub, they will see the name of a PID minting service in the dropdown menu and EGI-DataHub will request generation of the PID for this dataset and from now on it will be included in the OAI-PMH endpoint listings including the provided DC metadata.
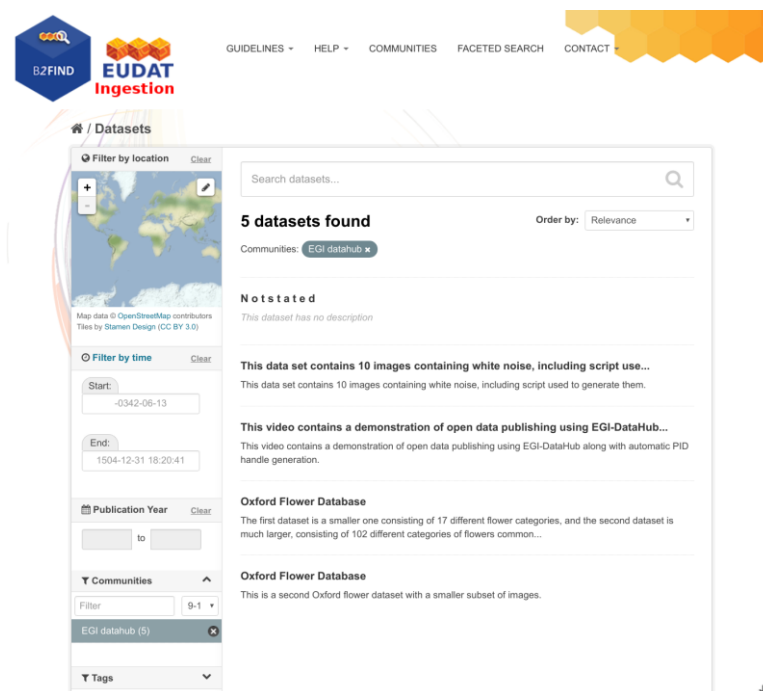

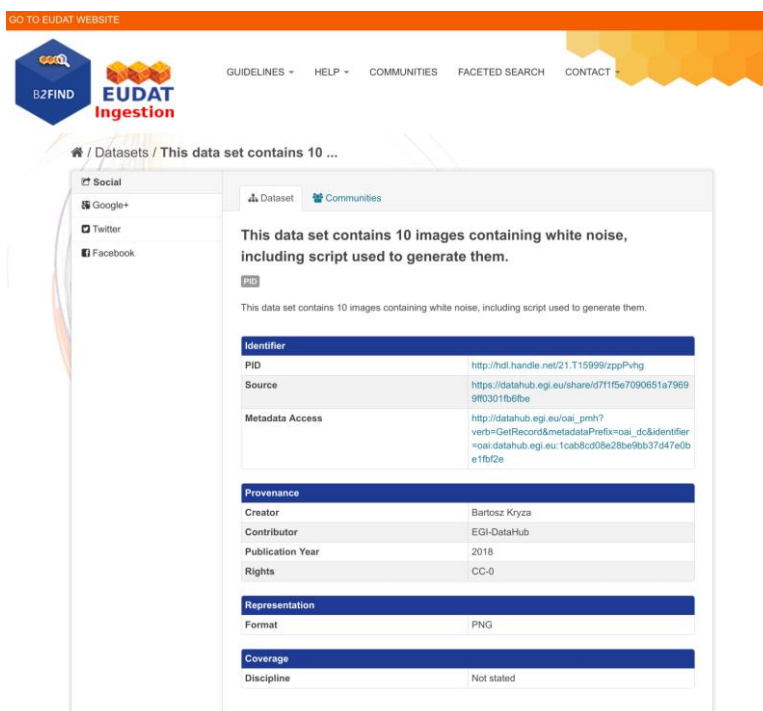*Figure 3. Test open data sets harvested from EGI DataHub by B2FIND*
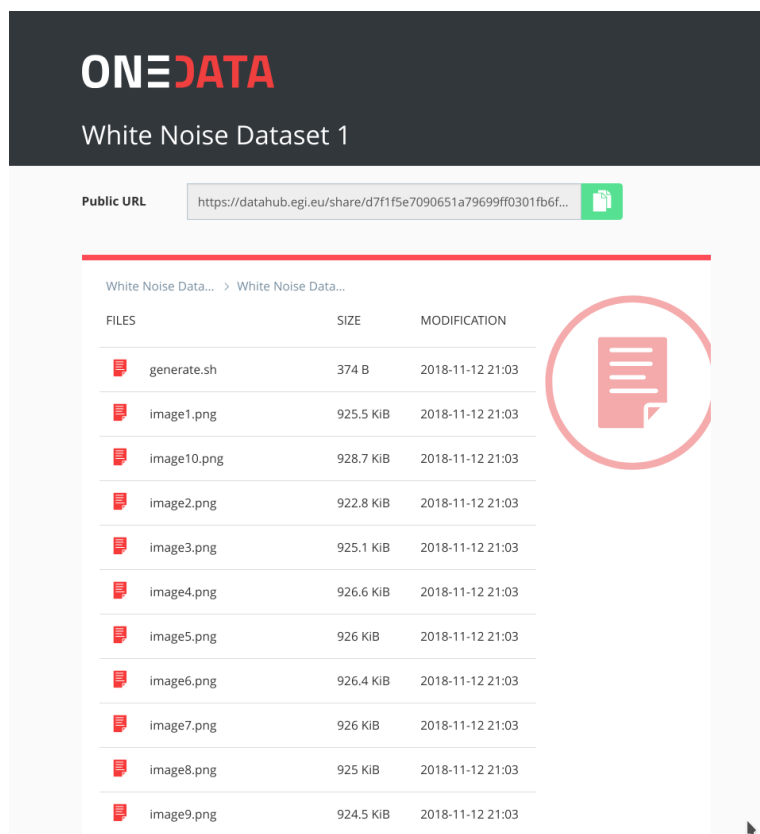
Figure 4. Metadata of a selected dataset harvested by B2FIND

Figure 5. Referenced data set can be accessed from web browser directly by following the handle

The remaining issues involve alignment of the metadata schemes between B2FIND and EGI DataHub, ensuring that fields relevant for specific communities will be provided when publishing the datasets via EGI DataHub.

In order to facilitate training and demonstration activities related to data discovery and open data publishing, a demo instance of EGI DataHub has been setup at https://datahub-demo.egi.eu, which allows creating test open data sets, which will be then discovered by a special instance of B2FIND without cluttering the official public B2FIND repository with test records. This feature will be fully enabled when EGI DataHub is upgraded to the latest currently published Onedata version - 19.02.1.

3.2.2   Staging data stored in EGI DataHub by B2STAGE for processing

During the former reporting period the integration between B2STAGE and EGI DataHub has been investigated, reaching the conclusion that the task was not feasible due to the lack of integration between B2STAGE and B2ACCESS. As a result, the activity has been diverted towards the integration between EGI DataHub and B2SAFE. To reach the goal EGI DataHub started to implement a WebDAV client to expose B2SAFE resources through the davrods interface. Furthermore, transfer tests have been performed.

In the scope of this activity, WebDAV driver has been added to Onedata which is the underlying platform for EGI DataHub, which allows to both import existing data from B2SAFE as well as storing data from DataHub in B2SAFE, using a WebDAV endpoint. We have validated the transfers using test instances of WebDAV endpoint provided by CINECA and a test instance of Onedata. However,

to enable it for production, we need to implement a token refresh mechanism in Onedata, which will allow registration of WebDAV endpoint permanently (by default access tokens generated using B2ACCESS expire after just a few hours).

Now the integration between B2STAGE and B2ACCESS has been completed, as described in detail in section 3.2.4. and exploiting this integration we will be able to proceed with the task to integrate EGI DataHub and B2STAGE, as originally planned.

This integration activity allows open to move data, from any configured endpoint, not only from B2SAFE. Detailed instructions can be found within the EGI wiki page 'Jointly exploit EGI and EUDAT services' [7] and in a step by step demonstration[8] shows in two scenarios how data can be staged from and into EGI datahub Onedata storage.

3.2.3 INDIGO-DataCloud IAM authentication integration with EGI DataHub

IAM service is already integrated with EGI DataHub. It can be enabled in the main EGI DataHub Onezone instance at https://datahub.egi.eu by means of its configuration file. Once enabled the users logging to EGI DataHub will be able to choose IAM as identity provider and then authenticate using any of the mechanisms supported at that IAM instance.

3.2.4 Integration between B2STAGE and B2ACCESS

B2STAGE HTTP-APIs fully implemented the OAuth2 workflow required to support the B2ACCESS authentication (described in the Service Integration Documentation from B2ACCESS) by exposing two different endpoints. The first (/auth/askauth) is intended to let B2STAGE manage the whole OAuth2 workflow and it requires that the user operates through a web browser. The second one (/auth/b2safeproxy) allows the authorization via B2ACCESS to be skipped by directly providing an access token, so that this endpoint can also be requested from command line interfaces and/or included in automated scripts.

When the user calls the /auth/askauth endpoint from a web browser the request is automatically redirected to the B2ACCESS website, where the user can log-in and authorize HTTP APIs to access the user profile. Then B2ACCESS redirects back to B2STAGE service by including two tokens: an access token (with a validity of a few hours) and a refresh token (that can be used to request for new access tokens). By using the access token, B2STAGE retrieves from B2ACCESS the user profile, in particular to obtain the email address used to map the request over B2SAFE users. The workflow described so far is not executed when the user calls the /auth/b2safeproxy endpoint, since in this case the B2ACCESS access token is provided by the user as input. In both cases B2STAGE creates, and provides to the user, a new JWT token linked to both B2ACCES tokens. That JWT token can be used to make further requests on restricted endpoints and it allows B2STAGE to transparently use B2ACCESS tokens. The access token is provided to B2SAFE to authenticate the user by adopting the PAM protocol. In case of authentication errors the access token is intended to be expired and B2STAGE uses the refresh token to ask B2ACCESS for a new access token.

---

[7] https://wiki.egi.eu/wiki/Jointly_exploit_EGI_and_EUDAT_services
[8] https://datahub.egi.eu/share/8c81c2dac4a2b3683b1727e49b5657f6

HTTP APIs are connected to B2SAFE by means of the [python-irodsclient](#) (PRC) but this library did not support the PAM protocol. The lack of this functionality delayed the completion of this activity and postponed other integration activities based on B2ACCESS common credentials. To be able to proceed with this task, we decided to directly contribute to the development of the python irods client and extend the required functionalities by providing a merge request with our implementation of the PAM protocol. The merge request has been accepted by the irods team and PAM is now officially supported by PRC.

3.2.5 Retrieve processed data from B2STAGE and share into B2SHARE Sharing processed data by B2SHARE

This task will implement a feature to let the user fetch files stored in an external storage system such as B2SAFE, within B2SHARE UI by utilizing B2STAGE for the actual transfer of bytes from B2SAFE to the user's computer.

In preparation for the common authentication layer between B2SHARE and B2STAGE, support for OAuth2 workflows and support for PAM protocol have been implemented to B2STAGE. Chapter 3.2.4 describes these activities more in detail.

Previously identified fixes to B2STAGE HTTP-API required to support B2SHARE integration have been implemented. Support for HTTP HEAD method was implemented, and it is now possible to provide JWT token in the URL instead of Authentication-header. This was required due to a technical constraint (HTTP GET request made by B2SHARE cannot provide an Authentication-header).

For B2SHARE, detailed implementation plans have been sketched and workflow defined in them has been fully tested. B2SHARE will require support for exchanging B2ACCESS userinfo tokens for B2STAGE JWT tokens and support for delivering this JWT token back to B2STAGE when the user requests to download a file stored in B2SAFE. Implementation of these features should be done with good usability in mind. Implementation of these features will continue during 2020.

3.2.6 Retrieve and store small data sets with B2DROP

Within the last reporting period the integration of B2SAFE with B2DROP was continued. For the operational workflow we recommend only an integration of a B2SAFE storage which is the ingestion point of a B2SAFE replication queue. If another server of this queue is connected, the B2SAFE service should only allow read access to the data. How B2SAFE deals with file changes by the B2DROP integration is up to the replication policies of the single B2SAFE services. Additionally, the recommendation was to use local B2SAFE credentials for the integration because with the B2ACCESS credentials, the users need to renew the credentials of B2SAFE, stored in B2DROP, once per hour.

The general possibility of deployment on the catch all B2DROP instance was rejected because the data transfer from B2SAFE through B2DROP would increase the network load on B2DROP by a factor of two and B2STAGE offers better performance for the file transfer.

3.2.7 EGI DataHub dataset discoverability in OpenAIRE Community Dashboard

EGI DataHub provides automated mechanism for publishing open data sets, which are then exposed via a standard OAI-PMH endpoint. Furthermore, EGI DataHub allows easy minting of data handles

(including DOI and PID), which enables assigning permanent identifiers to the published data sets which can be then referenced.

3.2.8 EUDAT dataset discoverability in OpenAIRE Community Dashboard

OpenAire and EUDAT-B2FIND enhanced the compliance of their guidelines for data providers[9]. This is particularly evident in the use of common standards (such as OAI-PMH) and the compatibility of the metadata schemas used.

We agreed furthermore to a cooperation between EOSC-Hub and OpenAIRE Advance[10] that comprises the provision of enriched metadata indexed in B2FIND to OpenAIRE. This will lead both to increased takeup and to an improved curation of meta data indexed in B2FIND by the services of OpenAire. In order to implement this, B2FIND planned in the last reporting period to offer its processed metadata in a format compatible with OpenAire via OAI-PMH.

Meanwhile an OAI-PMH API was built for datasets stored in B2FIND's metadata database. This is implemented based on the ckan extension for OAI-PMH exposure from CKAN[11]. The API returns data in DataCite format, which requires that the identifier for a digital object is given as DOI, so only the records that have a DOI are returned. The installation works technically, but before the API can be released to OpenAire and other interested aggregators for harvesting, some content issues and policies need to be clarified. Currently work is done to expose B2FIND metadata with OpenAire´s metadata schema, including metadata prefixes in OpenAire format, in order to expose all records with persistent identifiers, being they DOIs or e.g. Handles. This requires communication activities concerning standardization agreements, which is (and probably will always be) work in progress.

In this context we would also like to point out initiatives aiming to consolidate and unify metadata standards and schemes on the level of EOSC wide generic and FAIR metadata management. In this context we refer to the initiative of B2FIND and B2SHARE to establish a common metadata schema for EUDAT services and the cooperation with EOSCpilot 6.2 'Data Interoperability'[12] and the RDA's Data Discovery Paradigms IG[13].

3.3 Future Integration Plans

This section presents the overview of planned features we want achieve by extending or integrating existing services within the next project period, and enhance their relevance for thematic and specialized services.

- Normal software and service maintenance activities; e.g. upgrade and consolidate the metadata schema of B2FIND.
- Extend the uptake of metadata and indexing more data resources registered in EGI-datahub and B2SAFE

---

[9] compare guidelines of OpenAire (https://guidelines.openaire.eu/en/latest/data/index.html) and EUDAT-B2FIND (http://b2find.eudat.eu/guidelines/introduction.html)

[10] https://docs.google.com/document/d/1zXcDrrS2Ud8XL2lDFJcj2b1CQjMzmHKp7USBBJkKvVc

[11] https://github.com/openresearchdata/ckanext-oaipmh

[12] https://eoscpilot.eu/content/d66-2nd-report-data-interoperability

[13] https://www.rd-alliance.org/groups/data-discovery-paradigms-ig

- Set up an OAI endpoint on top of B2FIND from where OpenAire and other indexers can harvest metadata in a proper format
- Further development of B2DROP to enhance the allowed size of files and user storage space to support applications with big data volumes.
- Improve two-way integration with B2NOTE and B2FIND.
- Investigate the integration between B2STAGE and EGI DataHub and complete the data transfer tests between B2SAFE and DataHub.

4   Federated Compute

Federate Compute covers those services providing resources for the execution of user applications as virtual machines (Cloud Compute), as containers (Cloud Container Compute), or as jobs (High-Throughput Compute). Users needing tighter control on the resources and how these are allocated should use Cloud Compute, users with existing containerised applications following a cloud-native approach are better served with Cloud Container Compute, and for those users with the need to run parallel computing tasks at scale that can be modelled as traditional jobs in a batch system, High Throughput Compute will better meet their needs. The following table summarises the different options offered through these the services:

	Cloud Compute	Cloud Container Compute	High Throughput Compute
What is it?	Multi-cloud IaaS	Kubernetes on top of EGI Cloud Compute	The grid, a scalable batch system
What you run?	VMs	(Docker) Containers	Jobs
Typical workloads	Lift and shift existing applications Specific OS (kernel) requirements	Cloud-native containerised applications.	Execution of parallel computing tasks to analyse large datasets.
Pros / Cons	[+] Complete control on resources, run (almost) anything you'd like [-] Complex operation	[+] Industry standard [+] Hides complexity of Kubernetes setup [-] Some Kubernetes features not available	[+] No management of resources, just submit jobs [-] Legacy interfaces [-] Porting of applications

Table 1: Federated Compute services

These services are complemented and integrated with Workload Management, Online Storage, CVMFS and Advanced IaaS to provide advanced features on top of the basic computing power. Workload Manager provides users with an automated distribution of tasks across different computing services. Online Storage offers access to files and objects from the Virtual Machines, containers or jobs. CVMFS offers a software distribution system so the user applications are available in the distributed infrastructure. Advanced IaaS offers the possibility to easily execute containerised applications on systems without native docker support and without administrative privileges as available in the EGI High-Throughput Compute service.

4.1 Maintenance, interfaces and integration options of the services

4.1.1   EGI Cloud Compute

EGI Cloud Compute (https://www.egi.eu/services/cloud-compute/) provides users with a distributed computing service to deploy and scale virtual machines on-demand. It offers access to

API-controlled computational resources in a secure and isolated environment without the overhead of managing physical servers. For detailed description of the service see [D6.1] or refer to the service documentation at https://wiki.egi.eu/wiki/Federated_Cloud_user_support

4.1.1.1 Service Interfaces

- GUI access: AppDB VMOps https://dashboard.appdb.egi.eu/vmops
- API/CLI access:
  - Discovery: AppDB IS API (REST and GraphQL) https://wiki.egi.eu/wiki/Federated_Cloud_Discovery#AppDB
  - IaaS Federated Access Tools: https://wiki.egi.eu/wiki/Federated_Cloud_IaaS_Orchestration
  - Direct IaaS access, several APIs depending on the provider: https://wiki.egi.eu/wiki/Federated_Cloud_APIs_and_SDKs
  - Discovery: AppDB IS API (REST and GraphQL) https://wiki.egi.eu/wiki/Federated_Cloud_Discovery#AppDB

4.1.1.2 Possible Integration Partner Services

- Cloud Compute is integrated with EOSC-hub Service EGI Check-in
- Cloud Compute can be integrated with EOSC-hub Service B2DROP
- Cloud Compute can be integrated with EOSC-hub Service DataHub

4.1.2 EGI Cloud Container

Cloud Container Compute gives you the ability to deploy and scale Docker containers on-demand using Kubernetes technology. The service provides easy to provision Kubernetes clusters on EGI Cloud Compute resources that can be scaled and upgraded without the overhead of installing, managing and operating the nodes. For detailed description of the service see D6.1. and the user documentation at https://wiki.egi.eu/wiki/Federated_Cloud_Containers .

4.1.2.1 Service Interfaces

- Uses native Kubernetes API with OpenID Connect authentication: https://kubernetes.io/docs/reference/access-authn-authz/authentication/#authentication-strategies
- Kubernetes API: https://kubernetes.io/docs/concepts/overview/kubernetes-api/.

4.1.2.2 Possible Integration Partner Services

- EGI Cloud Container Compute is integrated with EOSC-hub Service Check-in.
- EGI Cloud Container Compute is integrated with EOSC-hub Service Cloud Compute.
- EGI Cloud Container Compute can be integrated with EOSC-hub Service B2DROP.
- EGI Cloud Container Compute can be integrated with EOSC-hub Service DataHub.

4.1.3 EGI Workload Management

EGI Workload Management allows users to manage and distribute your computing tasks in an efficient way while maximising the usage of computational resources. For detailed description of the service see D6.1. and https://www.egi.eu/services/workload-manager/.

4.1.3.1 Service Interfaces

- The Workload Manager service is based on DIRAC technology and is suitable for users that need to exploit distributed resources in a transparent way. The service has a user-friendly interface and also allows easy extensions for the needs of specific applications via APIs.
- DIRAC documentation is available at https://dirac.readthedocs.io/en/latest/index.html .

4.1.3.2 Possible Integration Partner Services

- Workload Management is integrated with EOSC-hub Service High Throughput Compute
- Workload Management is integrated with EOSC-hub Service Online Storage
- Workload Management can be integrated with EOSC-hub Service Check-in.
- Workload Management can be integrated with EOSC-hub Service Cloud Compute.

4.1.4 EGI Online Storage

EGI Online storage is a service that allows you to store data in a reliable and high-quality environment and share it across distributed teams. Your data can be accessed through different standard protocols and can be replicated across different providers to increase fault-tolerance. For detailed description of the service see D6.1. and https://www.egi.eu/services/online-storage/ .

4.1.4.1 Service Interfaces

Online Storage is offered via different APIs depending on the available providers and user needs:

- Block Storage offered via OCCI/OpenStack APIs: http://occi-wg.org/, https://api.openstack.org/
- Object Storage offered via Swift: https://api.openstack.org/
- File-based Storage offered via SRM: https://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html, WebDAV and GridFTP.

4.1.4.2 Possible Integration Partner Services

- Online Storage is integrated with EOSC-hub service Cloud Compute.
- Online Storage is integrated with EOSC-hub service High-Throughput Compute.

4.1.5 EGI High-Throughput Compute

EGI High-Throughput Compute allows users to run computational jobs at scale on the EGI infrastructure. It allows you to analyse large datasets and execute thousands of parallel computing tasks on a distributed network of computing centres, accessible via a standard interface. For detailed description of the service see D6.1. and https://www.egi.eu/services/high-throughput-compute/.

4.1.5.1   Service Interfaces

The service is offered via 3 different APIs, depending on the providers and users' preferences:

- CREAM: https://wiki.italiangrid.it/twiki/bin/view/CREAM/WebHome
- ARC: http://www.nordugrid.org/arc/, and
- QCG: http://www.qoscosgrid.org/qcg-now/en/

4.1.5.2   Possible Integration Partner Services

- EGI High Throughput Compute is integrated with EOSC-hub service Online Storage.

4.1.6   Advanced IaaS

uDocker allows the execution of applications and services within virtualized environments similar to Linux containers. It enables the execution of docker containers without requiring any privileges both for installation and execution, making it especially suitable to execute applications in batch systems or other environments where the end user does not have system administrator privileges. It is being used for high throughput computing, grid computing, GPU computing and high-performance computing. For detailed description of the service see D6.1 and https://github.com/indigo-dc/udocker.

4.1.6.1   Service Interfaces

uDocker is meant to be used from the command line interface and offers a docker like command syntax. uDocker supports the docker registry HTTP APIs V1 and V2. In addition, uDocker supports importing and exporting of container filesystems in tar format, and loading of containers in both docker and Open Containers Initiative (OCI) formats.

4.1.6.2   Possible Integration Partner Services

- uDocker is being used with the EOSC-hub thematic service OPENCoastS
- uDocker is being used with the EOSC-hub service EGI High-Throughput Compute
- uDocker is being used with the EOSC-hub service EGI Workload Management
- uDocker is being used with the EOSC-hub service EGI Cloud Compute.

4.1.7   CVMFS

The CernVM File System (CernVM-FS or CVMFS) is a read-only file system designed to deliver scientific software onto virtual machines and physical worker nodes in a fast, scalable, and reliable way.

CernVM-FS is a file system with a single source of data. This single source, the Stratum-0 repository, is maintained on a dedicated release manager machine or CVMFS Uploader.

4.1.7.1   Possible Integration Partner Services

- CVMFS can be integrated with EOSC-hub service Cloud Container Compute
- CVMFS can be integrated with EOSC-hub service Cloud Compute

● CVMFS is integrated with EOSC-hub service High-Throughput Compute

4.2 Integration activities

4.2.1 OIDC support in IaaS cloud management frameworks (OpenStack, OpenNebula, Synnefo)

The integration of the different cloud middleware options available on the provider of the EGI Cloud Compute was completed during the first year of EOSC-hub. The integration consists on the support of OpenID Connect at the providers by using the existing authentication services and documenting the needed deployment and configuration options available in the case of OpenStack, or by providing the Keystorm (https://github.com/the-rocci-project/keystorm) component that deployed alongside the cloud platform allows the support of OpenID Connect easily in the case of OpenNebula. Synnefo development was not continued as this middleware is currently under deprecation in the federation and the efforts have been shifted to improve the support for OpenStack.

The use of OpenID Connect allows to support both web-browser and API/CLI based-access seamlessly. The figure below shows the dashboard login page of IN2P3-IRES provider. In order to facilitate the access via APIs, a dedicated EGI Cloud client for obtaining and renewing credentials was created at https://aai.egi.eu/fedcloud.

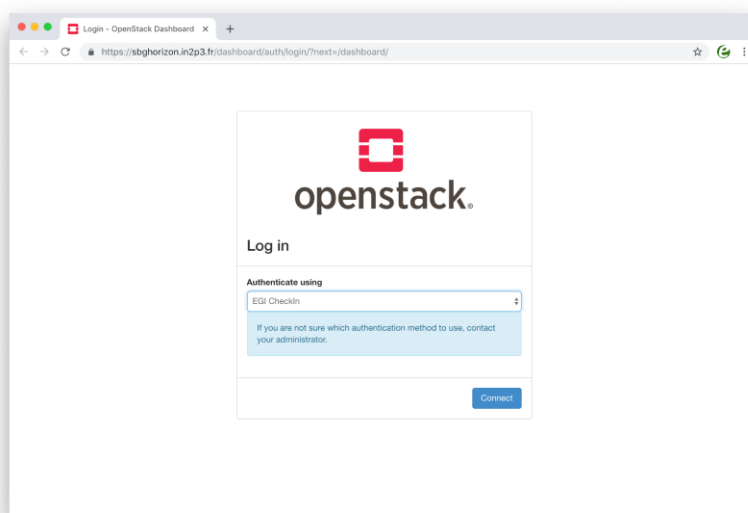*Figure 6. OpenStack Login dashboard with EGI Check-in*

Detailed documentation on how to integrate and configure providers is available through the EGI Cloud integration documentation and currently all providers in the federation are going through the integration process.

Besides the providers support, clients have been adapted or documentation provided to support this new authentication method:

- New monitoring probes using OpenID Connect were released and added to the ARGO service
- Use of OpenID Connect with OpenStack clients is described in the service documentation
- OCCI clients were updated to support OpenID Connect and documentation updated.

Other clients, out of the scope of the WP6.2 task such as IM and Terraform, were tested and a detailed description on how to use them with the new authentication is provided in the service documentation.

4.2.2 Application Database integrational activities

Although the AppDB VMOps dashboard controls the lifecycle of deployed VMs, it has no control over 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 in malfunctioning VMs or deployment failures which, in some cases, can only be noticed by the VM's owner. Currently, there are no means for users to address such issues. Integration with the GGUS service provides a channel for users to communicate their issues with cloud provider administrators and resolve them. A graphical interface lets users create a ticket within the GGUS system, addressing a specific VM. The AppDB VMOps portal automatically enriches the ticket with all the necessary information related to the specific VM, in order to help site administrators, locate and resolve the issue. Moreover, users are 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 (see figure 7).
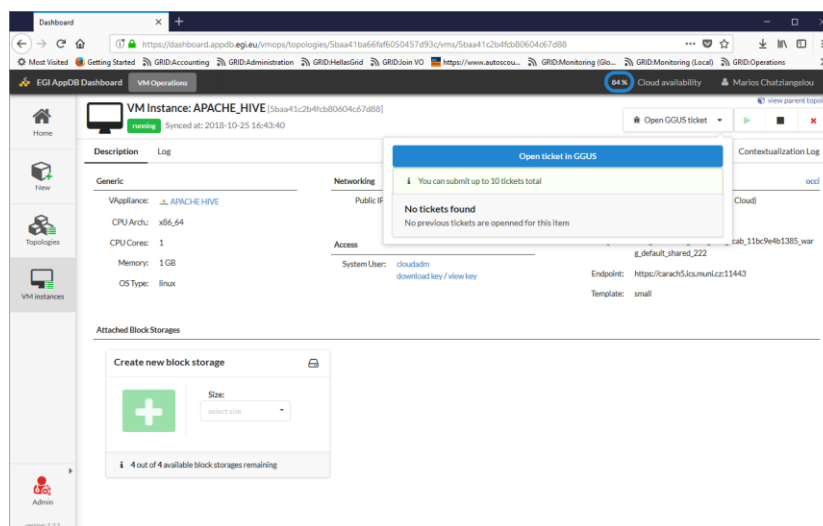
Figure 7. File a GGUS ticket directly to the site via the VMOps

A second integration that has been introduced is the one with the Security Cloud Assessment Tool (SECANT), developed by CESNET, which performs automated security checks for all new VA versions upon publishing. AppDB and Secant communicate using the EGI Argo Messaging service, upon which a message flow was implemented in order to pass information about available appliances and their status after the analysis. An initial list of checks performed by SECANT is provided in the following table.

Security check title	Short description
OPEN_PORTS	Detect services and ports available on the machine
NTP_AMP	Check the NTP server can't be abused for DDoS amplification attacks
SSH_AUTH	Check whether SSH forbids password-based authentication
SSH_PASSWD	Detect weak passwords available over SSH
LYNIS_TEST	Detect weaknesses of tools and libraries using the Lynis framework
PAKITI_TEST	Detect unpatched software packages

4.2.3   Advancements in CREAM/BDII information services

The CREAM (Computing Resource Execution And Management) Service is a simple, lightweight service that implements all the operations at the Computing Element (CE) level. The service is a basic component for a federated service-oriented architecture managing distributed processes (jobs). In order to guarantee the interoperability among different applications it implements a standard Web Service interface based on WS-I specification.

The CREAM service accesses and operates local resource management systems. The current version of the application supports the following management systems: TORQUE, LSF, SLURM, HTCondor, Grid Engine (partially).

The authentication to the CREAM service is based on X509 certificates and RFC 3820 proxy certificates; the authorization is attribute based and resorts to certified attributes published by Virtual Organizations. Attributes are embedded into user proxy certificates. Interaction with the federated authentication systems, like SAML or OpenID Connect, is possible through a Token Translation Service.

The CREAM service has been declared the main Computing Element for the grid environment by the European Grid Infrastructure (EGI). The software must be considered completely stable, no more features are planned. It has been deployed and operated in the grid environment for more than ten years. The maintenance in the recent past is mainly dedicated to the improvements of the security infrastructure, for example with the adoption of new cryptographic schemes, and changes required for keeping the compatibility with the latest releases of the batch systems.

For the Federated Compute package of EOSC-Hub a new installation and configuration tool has been developed for the CREAM and the resource BDII services. The tool is based on the puppet framework and replaces the old reference application (YAIM) for the grid environment and it is distributed as a puppet module through the puppetforge portal: https://forge.puppet.com/infnpd/creamce. The module provides a rough support to the common batch systems, besides the configuration of the core of the application. Such a support is meant to be improved by users' requirements that will be collected in the next future.

The main distribution channel for CREAM and BDII applications is through the Unified Middleware Distribution (UMD) coordinated by EGI. In order to be fully compatible with the process workflow

of UMD the build system and continuous integration of the software have been re-designed. The platform chosen for the continuous integration is Jenkins; the build system to take advantage of its advanced features, like pipeline processing and docker container support, for compiling and testing the code efficiently. The system at the moment makes use of the Jenkins platform hosted at INFN; it will be completely integrated into UMD workflow as soon as the compatibility will be certified.

All the improvements of the information system (BDII) are related to the adoption of the GLUE schema, version 2.1. The new version of the standard defines all the entities required for a complete model of cloud infrastructures and resources equipped with accelerator devices, like GPUs or MICs. The GLUE schema version 2.1 is still a draft document but, as a proof of concept, an experimental release of BDII information providers, publishing the aforementioned entities, is already available.

Since the beginning of January, the maintenance of the CREAM CE service covers only security updates. The CREAM CE service is deprecated and will not be supported beyond 2020.

4.2.4   cloudkeeper advancements

Cloudkeeper is a suite of tools that synchronize user-specific virtual machine images from a common source (typically AppDB) to all relevant cloud sites where the user can expect such images. The synchronization process includes not only transfer but also registration at the target cloud site and -- if required -- conversion to a format supported by the target infrastructure. Cloudkeeper does not cover only the initial upload but rather manages the whole life cycle of the virtual machine image, i.e., distribution, updates and end-of-life removal.

This Task was aiming at updating and streamlining Cloudkeeper's workflow, especially in view of the upcoming transition from a site-centric to a VO-centric operation of the cloud integration tool. This led to the recent release of Cloudkeeper 2.0, the main difference between the previous and new major version being that Cloudkeeper 2.0 is prepared to run on behalf of a VO, requiring no elevated rights, and synchronizing all the VO's images to all sites supporting that VO.

For cloud site administrators this means that they no longer need to run cloudkeeper themselves for every VO their site supports. For VO administrators this means that they do not need to negotiate individually with each site, especially in case of issues, which are now concentrated into a single cloudkeeper instance per their VO. It also makes it easy for EGI Operations to support selected (or all) VOs by running the integration tools for them, without having to act through potentially dozens of site administrators.

The Cloudkeeper-one backend for OpenNebula-based cloud sites within the EGI Federated Cloud platform has also already been updated for Cloudkeeper-2.0.
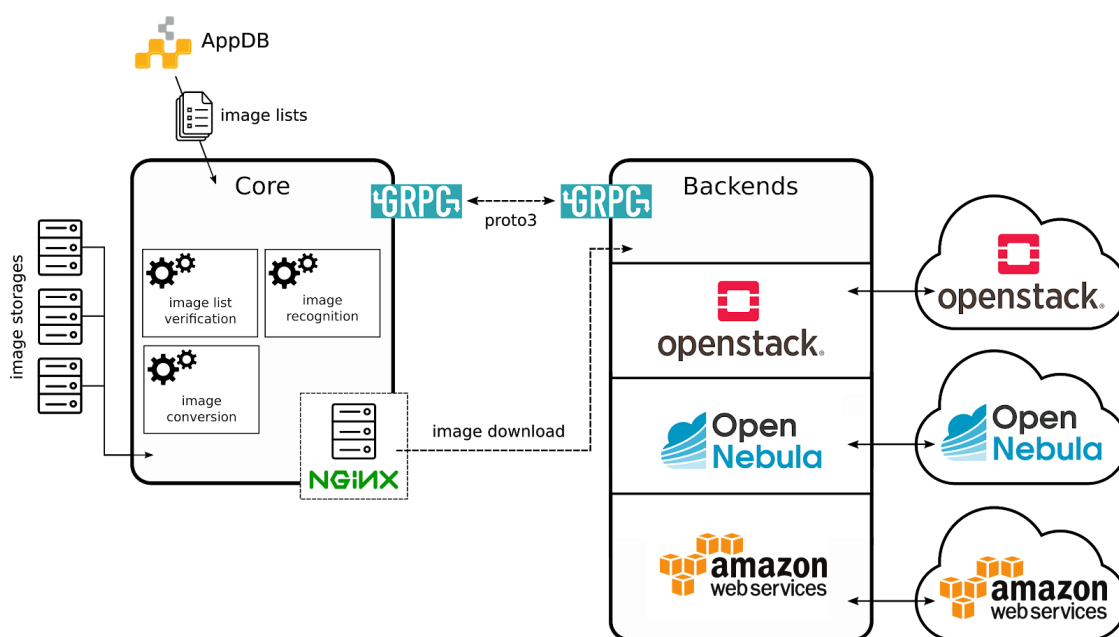
Figure 8. Cloudkeeper 2.0 ecosystem

During the reference period cloudkeeper has also received a major 3rd-party code contribution funded by project GN4-2, which took the form of a completely new and functional Cloudkeeper backend for Amazon Web Services. It allows VOs to synchronize their images not only to sites associated with the EGI Federated Cloud platform, but also to Amazon Web Services, where they are ready for cloud bursting. This contribution has been successfully tested and integrated within this Task.

Finally, updates required in the Cloudkeeper-os backend for OpenStack have been identified and implementation is ready to commence.

4.2.5 Elastic Kubernetes cluster support

In the first year of the project:

Kubernetes provides an ideal platform to run container-based workloads be it an application composed of several microservices or a High Throughput Computing application composed of loosely-coupled jobs. Indeed, Kubernetes supports autoscaling capabilities by means of the *Horizontal Pod autoscaler* that is in charge of determining the right number of pods inside a Kubernetes cluster depending on the varying workload. However, this does not affect the number of nodes of the Kubernetes cluster. To this aim, Kubernetes offers the *Cluster Autoscaler* which is responsible to scale the cluster nodes when the pods cannot be scheduled on nodes because there are no free resources available. However, this component is only functional for Amazon Web Services, Microsoft Azure and Google Cloud Platform.

It was the aim of this task to include auto-scaling support for Kubernetes clusters deployed by the users on any IaaS Cloud site. To this goal, first a set of Ansible roles for the automatic installation and configuration of a Kubernetes cluster have been created. It enables to provision a fully functional Kubernetes cluster and the addition/removal of new nodes on runtime. It includes the

configuration of the different network plugins (Flannel, Calico, Weave, etc.), the deployment of the dashboard and the installation of the Helm tools:

- https://github.com/grycap/ansible-role-kubernetes

Then a plugin for the CLUES elasticity system has been developed to enable the elastic management of the Kubernetes cluster adding/removing nodes based on the workload.

- https://github.com/grycap/clues/blob/master/cluesplugins/kubernetes.py

The EC3 templates has been created to enable launching the elastic Kubernetes cluster using the EC3 tool automatically.

- https://github.com/grycap/ec3/blob/master/templates/kubernetes.radl

In this second year of the project:

The main objective of the second year has been to maintain and update the kubernetes role to adapt it to the fast-developing process of the kubernetes project. The role has been updated to the 1.16 version that has introduced some major changes in the API that involved many changes in the role. It has also been updated to use the version 2.0 of the dashboard. Finally, support has been added for the NFS client provisioner to make it easier to create persistent volumes using an NFS backend. Currently we are addressing the migration to Helm v3 and the installation of the cert-manager to generate certificates using valid CA entities as Let's encrypt.

4.2.6   uDocker advancements

The main EOSC services relevant for uDocker integration are the EGI High-Throughput Compute, EGI Workload Management, and EGI Cloud Compute as well as the thematic services. In this context uDocker is ideal to encapsulate the execution applications in such heterogeneous environments. Its interface and integration code are written in Python and offers several execution engines to mimic chroot like behaviour. The four execution engines include a ptrace of system calls based on ptrace (Pn modes), sharable library preload to intercept library calls based on fakechroot (Fn modes), runC using namespaces where supported (Rn modes) and the possibility of using Singularity if already locally installed (Sn modes). The latest production release of udocker is v1.1.4 issued in January 2020 and available at:

> https://github.com/indigo-dc/udocker/releases/tag/v1.1.4

A new pre-release of uDocker for Python 3 was also released and is available at:

> https://github.com/indigo-dc/udocker/releases/tag/devel3_1.2.4

Several enhancements have been included to facilitate the integrated usage of uDocker with the mentioned EOSC services. These include:

- Enhancements to the command line interface and docker repository APIs for better interoperability:
    - support for image names in format host/repository:tag
    - listing of tags in docker container repositories
    - improved search in docker repositories

- ○ listing of local container attributes such as size and execution modes
- ○ added renaming of containers
- ○ added support for container manifests in v2 format
- ○ improved authentication and handling of redirects when accessing docker repositories
- Added new capabilities to facilitate interoperability and exchange of containers including:
  - ○ saving uDocker containers in docker format including metadata and layer.
  - ○ loading and handling of containers in Open Containers Interface (OCI) image format.
  - ○ enhanced container image verification for both docker and OCI containers.
- General execution enhancements:
  - ○ improved handling of mount points and cleanup
  - ○ add support for reading environment variables from file
  - ○ enable override via environment variables and configuration parameters of the libraries and executables that support the execution modes
  - ○ added capability of fixing container file permissions and ownership
- Fn execution modes using shared library interception:
  - ○ added support for new Linux system calls
  - ○ added interception libraries for newer Linux distributions of Ubuntu, Fedora, CentOS, Alpine, Debian, LinuxMint and RedHat.
  - ○ Improved pathname translation for mounted paths
- Sn execution modes based on Singularity:
  - ○ support for root emulation in Singularity based execution modes
  - ○ add support to enable transparent file binding
- Pn execution modes based on system call interception:
  - ○ additional fixes for the accelerated mode with SECCOMP
- Rn execution modes based on runC:
  - ○ add overlay execution mode enabling the use of system call interception mode Pn on top of Rn modes for better handling of non-privileged namespace limitations such as group handling.
- Security enhancements:
  - ○ create all processes without shell context
  - ○ safer validation and parsing of configuration files

4.2.7 Accounting

The work on improving the accounting of resource usage has been focused on three main features:

- Provide correct figures for long-running VMs in the infrastructure. A series of issues in both the APEL accounting repository and the accounting probes has been fixed during the second period of the project with releases of the cASO accounting component and a set of operational tickets to coordinate and follow the progress of the updates into the infrastructure.
- Provide accounting for public IP addresses. After the format of the record for IP addresses was agreed a first implementation for its generation is available for OpenNebula and the

OpenStack version is in progress. The APEL repository is also being adapted to capture this information.

- Similarly to the IP accounting, block storage devices record format is now agreed and the implementations are starting to be available. OpenNebula implementation is ready and OpenStack implementation is under heavy development. APEL repository is ready to accept the records.

4.2.8 GPGPU integration

One of the aims of the GPGPU integration is to ensure that all technologies necessary for operation of Accelerated computing within EOSC-Hub Federated Cloud, with different CMF (Cloud Middleware Framework) and integration tools (e.g. VMOps, image management, accounting, information services). That also provides computing services and support for user communities interested in using GPGPU for accelerating computation in their applications.

Two sites with GPGPU with different CMFs have been deployed and integrated into EOSC-Hub Federated Cloud: one with Openstack and the other with OpenNebula. The integration process was carried out in close cooperation with other integration tools and problems were reported to developers of the corresponding frameworks/tools. The GPGPU attributes for images have been defined in AppDB and propagated via cloudkeeper to sites. GLUE scheme 2.1 for describing GPGPU related information in BDII has been proposed and is being incorporated into the cloud-info-provider. Recently, the Openstack site with GPGPU is also integrated with EGI Check-in via OIDC.

For development and testing with GPGPU, EOSC-Hub users can use the dedicated virtual organization acc-comp.egi.eu. The VO is fully integrated to EGI CheckIn. Other user communities are also supported on the GPGPU sites, including enmr.eu, biomed. The provider also helped the users to solve different problems during deployment and execution of applications using GPU on the sites.

In order to make the use of cloud GPGPU resources easier for users the NVIDIA Docker Virtual Appliance Image was created and is provided through EGI Applications Database. NVIDIA Docker technology allows to build and run Docker containers leveraging NVIDIA GPUs. The VA Image allows users to run various pre-build GPU-ready containers (e.g. deep learning library TensorFlow) inside virtual machines with attached NVIDIA GPU accelerators.

4.2.9 Improved support for native API's of cloud stacks

Following the work of the previous period to allow cloud providers of the service to expose native APIs instead of or in addition to the OCCI API, the EGI Cloud Compute service has further ensured the different integration components are ready to work with different underlying technologies. In this period, we have also experienced a transition of most providers to OpenStack, hence while capable to use different APIs, the service has become more homogeneous and capable of leveraging the extensive toolset available for the OpenStack platform. Users are also provided with tools that help them to interact with the heterogenous providers without the need to care about the low-level details of each provider of the federation:

- The Infrastructure Manager (IM) is now the default orchestrator to manage deployments on the infrastructure. IM supports a wide range of IaaS cloud APIs with little to none changes of a deployment description.
- The AppDB VMOps provides a unified dashboard that allows users to manage VMs on the different providers without the need to log into each of them separately. AppDB relies on IM to access the endpoints.

The support of native APIs requires also support on the following integration components, that have been adapted and updated to enable the seamless integration of the providers into the infrastructure:

- Information discovery: the implementation of the GLUE 2.1 Schema standard for OpenStack and OCCI was finalized. The cloud-info-provider has been extended to support pluggable refresh of credentials (with a plugin implemented to support independent VO-level authentication) and pluggable output modules (with a plugin implemented to use the Argo Messaging System as output). These two extensions allow the operations of the tool in a centralized manner, thus removing the need to have providers operating the information discovery.
- AppDB: turning the system cloud provider API agnostic with the support for OpenStack and OCCI. This required the finalization of the support for the GLUE 2.1 Schema and usage of the Argo Messaging System for receiving provider information and ensuring the information can be properly used and displayed across the AppDB. AppDB has also completed the transition to OpenID Connect federated identity for interacting with the infrastructure, removing the need for certificates and the usage of PUSP (Per-User-Sub-Proxy) that presented some limitations in the control of access of users into the providers.
- Extension and implementation of the monitoring probes using native OpenStack APIs. Ensuring these resources are properly monitored in the infrastructure.

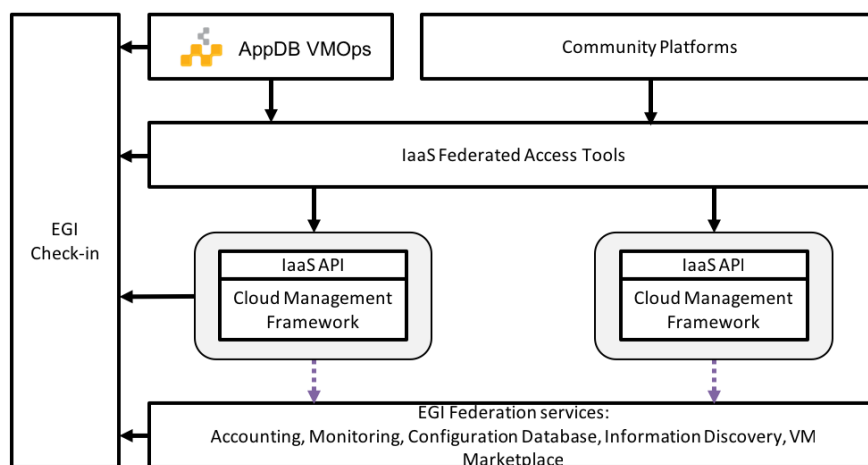The architecture of the EGI Cloud is now established as shown in the figure below:

Figure 9. Architecture of the EGI Cloud.

The documentation of the service was reviewed and updated to reflect this new architecture and is now available at https://egi-federated-cloud.readthedocs.io for users and at https://egi-federated-cloud-integration.readthedocs.io for providers.

4.2.10 Integration of EGI Cloud Compute and AAI services

EGI Cloud Compute resources can be integrated with other computing resources like EGI grid sites or even standalone computing centers. This is performed with the help of the EGI Workload Manager service built on the basis of the DIRAC software toolkit. In order to incorporate EGI Cloud Compute resources, the following developments are undertaken:

- OCCI Endpoint was developed as an implementation of the DIRAC abstract interface to cloud managers. This implementation is using the REST OCCI interface and has a much better performance compared to the previous implementation based on the rOCCI command line tool to access OCCI cloud managers.
- The OCCI interface is used to access EGI cloud Compute resources with the authentication mechanism based on the X509 proxy certificates. The X509 proxy certificates are managed by the ProxyManager subsystem of the EGI Workload Manager. This allows users to allocate cloud resources with a proper authentication whenever user payloads are available in the system.
- The OCCI interface was introduced in order to have a common, and X509 compatible, interface to various cloud managers (OpenStack, OpenNebula, etc). As of recently the OpenStack cloud manager, which is used in the majority of the EGI cloud sites, is enabled for authentication with the X509 VOMS certificates. Therefore, new cloud connectors were developed for the EGI Workload Manager that use directly the OpenStack and OpenNebula REST interfaces without the intermediate OCCI layer. This makes the system more efficient and prepares for the eventual withdrawal of the OCCI support.

Another important area of development concerns the use of the authentication mechanism based on the OpenID Connect technology utilised in the EGI Check-In service. The following activities are undertaken:

The EGI Workload Manager Web Portal interface was enabled for authentication using the EGI Check-In service.

The mechanism of registration of new users who are authenticated by the EGI Check-In service was developed. The work is ongoing on proper implementation of the compute resources usage policies by users of different Virtual Organisations as defined in the EGI Check-In user profiles.

Work is also ongoing to provide the EGI Check-In authentication for use with the command line tools to access the EGI Workload Manager.

4.2.11 Access demonstration from EGI Cloud Compute to EOSC services

EGI Cloud Compute can potentially be integrated with any other service as the VMs run on the cloud can execute any arbitrary software. Access to any service requires that the correct access credentials

are available at the running VM, which can be injected in the contextualisation phase of the VM or via user input during the VM life-time.

As a Proof of Concept of the access to other EOSC-hub service, we have deployed pilot installations of the EGI Notebooks service on EGI Cloud which integrates the following services:

- EGI Cloud Compute provides to underlying computing resources for the execution of the service
- EGI Online Storage is used to provide additional block storage for the VMs running the services so the permanent user space associated with each notebook can be managed independently of the VMs. This allows storage to be safely persisted even if the VMs fail and augments the available space for each user.
- EGI Cloud Container Compute is used to provide Kubernetes deployment on top of the VMs. This Kubernetes cluster is configured so it can be managed using authentication for EGI Check-in
- A new client for EGI Check-in is created for the Notebooks, so users can effectively authenticate into the service using their EGI credential.
- The Notebooks service was extended to refresh authentication credentials from users at any time they are needed for interaction with other services. This allows users of the notebook to access any other service using Check-in for authentication with their own credentials and without the need to re-login.

Furthermore, to show the capabilities of the integration with other EOSC-services, two use cases were implemented into these deployments:

4.2.11.1 Open Data Analysis

By integrating EGI Datahub with the notebooks, users are able to access files from open data repositories as if they were local and can also create new share spaces that can have PID minted in B2HANDLE and discovered via B2FIND. In this implementation, when users launch a new Notebook in the service, the service contacts EGI Check-in to obtain a valid access token and in turn use that access token to create a new EGI Datahub token that can be used to mount the user spaces. The new notebooks are launched with an extra file system managed with the oneclient that exposes all data accessible to the user in the DataHub.
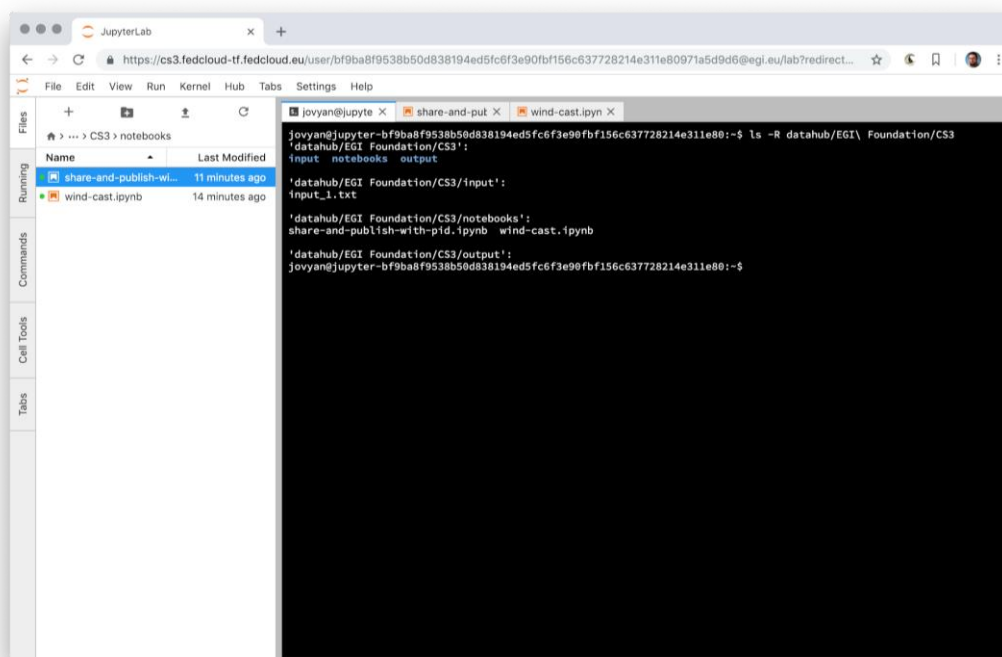
Figure 10. Notebooks environment with a terminal accessing the DataHub files.

As DataHub is already integrated with other EOSC-hub services, users can access these indirectly as well (e.g. minting PIDs for datasets with B2HANDLE and discovering those datasets). From the notebooks interface, users can easily share and mint PIDs for datasets using the provided APIs. The complete use case is shown in the following diagram:
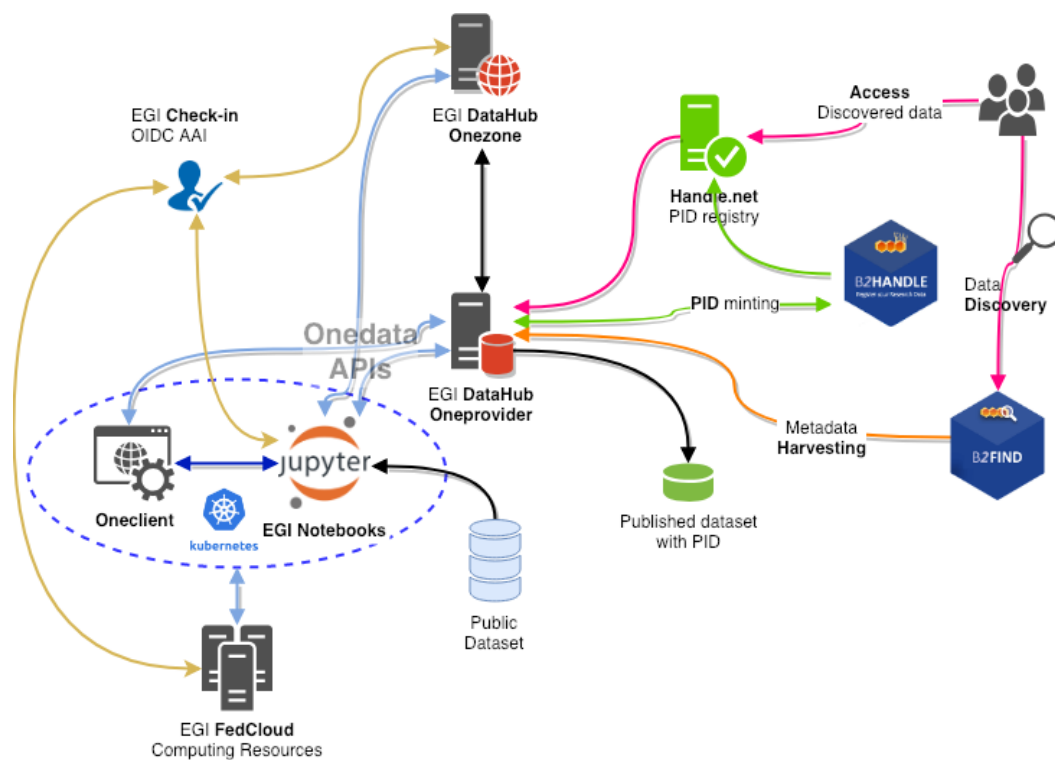
Figure 11. Notebooks with DataHub, B2HANDLE and B2FIND.

4.2.11.2 Access to B2DROP

B2DROP offers a WebDAV interface that can be mounted from the EGI Cloud VM. However, as the authentication is handled with B2ACCESS the integration cannot be performed automatically as with the DataHub and requires the user to provide her own credentials as input when launching the notebook. Similarly, to the Datahub case above, the files available in B2DROP will show up in the Notebook as a mounted file system that can be used transparently for any storage needs. The youtube video at https://www.youtube.com/watch?v=BYI3a_EOFJo shows a short demo of this feature.

4.2.11.3 uDocker in Sensitive data service

The EOSC-hub project is providing services for sensitive data through two partners: the Sigma2 / University of Oslo in Norway, and the CSC in Finland. CSC offers the ePouta secure cloud infrastructure, which provides customer organisations a virtual private cloud connected to the customer's infrastructure through a secure virtual private network. The University of Oslo provides TSD - the Norwegian e-Infrastructure for sensitive data storage and management. TSD provides sensitive data services directly to researchers and groups in the form of SaaS (Software as a Service) and PaaS (Platform as a Service).

Several container platforms have been evaluated, by both EOSC-hub T6.6 and PRACE WP 6.2.5, for support in sensitive data services. uDocker has been found the most secure, since it does not require admin privileges to install and/or use. uDocker therefore is recommended for ePouta users and is supported as part of the software stack in TSD.

4.2.12 Amnesia in Sensitive data services

As part of the collaboration between EOSC-hub and OpenAIRE projects, TSD - the Norwegian e-Infrastructure for sensitive data storage and management now supports data anonymization through Amnesia which is a flexible data anonymization tool that transforms relational and transactional databases to dataset where formal privacy guarantees hold. It allows users to remove any identifying information from sensitive data.

5   Processing and orchestration

This task focuses on the maintenance and integration of orchestration services with the Cloud Compute and Cloud Container services. This allows to build complex virtual computing infrastructures based on the OASIS TOSCA Simple Profile YAML standard[14] and integrate the INDIGO-DataCloud PaaS components as orchestrator for the EOSC-hub services.

Figure 11 below provides an overview of the architecture and interrelation of the different components that are part of task T6.3 "Processing and Orchestration". It also includes additional components that, even though they are not strictly included in T6.3 since they are not expected to be evolved in the context of EOSC-HUB, they are part of the PaaS Orchestration layer.

---
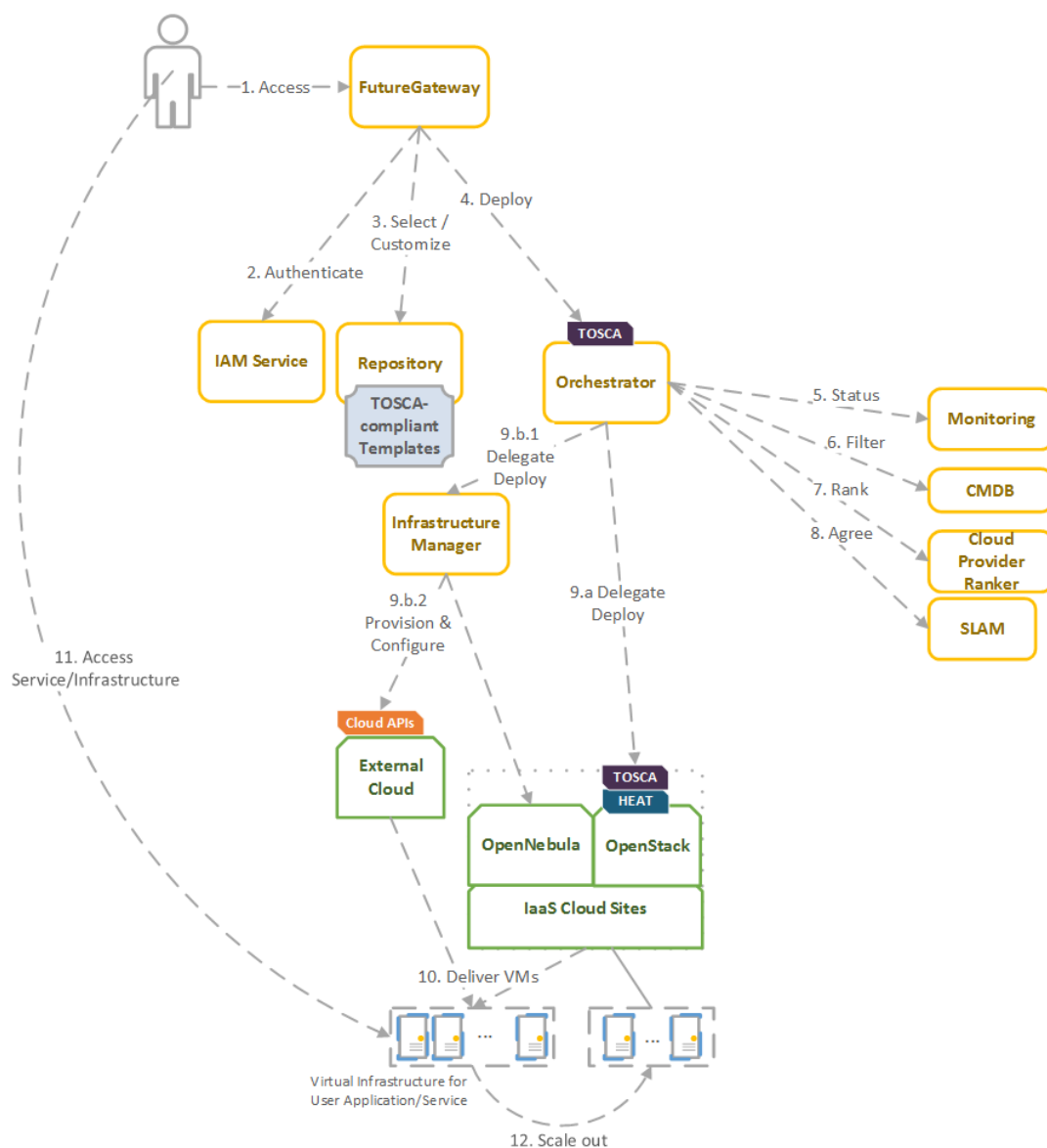
[14] Crandall, John, and Paul Lipton. "OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC." https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca.

Figure 11. Architecture and workflow among the Processing and Orchestration services in EOSC-hub.

End users are expected to access the PaaS Orchestration layer via high-level graphical user interfaces, such as the portlets provided by the *FutureGateway*. These are web-based components that are customized for specific user applications and are responsible for performing the authentication with the *IAM service* and interacting with the *Orchestrator* by submitting a TOSCA template. TOSCA stands for the Topology and Orchestration Specification for Cloud Applications and it is a YAML-based domain-specific language (DSL) to describe application architectures to be deployed on a Cloud. Advanced end-users could also interact with the Orchestrator via the authenticated REST APIs provided.

Once the Orchestrator receives the TOSCA template interacts with different services in order to identify the most appropriate IaaS Cloud site on which to perform the execution. This decision depends on the monitoring state of the underlying Cloud sites (information managed by the *Monitoring* service), the SLAs (Service Level Agreements) agreed between the user and the sites (information managed by the *SLAM* service) and the availability of the VMIs (Virtual Machine Images) on each site (information managed by the *CMDB* service). With all of this information, the *CloudProviderRanker* service is employed in order to apply a set of rules in order to obtain a ranked list of Cloud sites.

The Orchestrator delegates on the *Infrastructure Manager* to perform the deployment on public Cloud sites (Amazon Web Services, Microsoft Azure, Google Cloud Platform and Open Telekom Cloud) or on other external Clouds managed by popular Cloud Management Platform (CMPs) such as OpenNebula and OpenStack. The IM can also be configured with single-site mode in order to provide a TOSCA-enabled endpoint and support local-site orchestration of complex application architectures. In order to achieve a similar functionality in OpenStack, the *HEAT Translator* component can be used in order to translate from a TOSCA template into a HOT template, the native language employed by the HEAT service in OpenStack.

The following section provides an overview of the services involved in task T6.3 "Processing and Orchestration in EOSC-hub". For each service, a brief description is included and, in a separate section, the added value in EOSC-HUB for each service is identified.

5.1 Maintenance, interfaces and integration options of the services

5.1.1 TOSCA for Heat

TOSCA for Heat is a tool that translates TOSCA templates to Heat Orchestration Template (HOT) format. For detailed description of the service see D6.1. and https://github.com/indigo-dc/heat-translator.

5.1.1.1 Service Interfaces

- CLI

heat-translator can be used by TOSCA-aware services such as the EOSC hub services' IM and PaaS Orchestrator in order to deploy resource stacks in OpenStack Heat.

5.1.2   Infrastructure Manager

Infrastructure Manager (IM) is a tool that orchestrates the deployment of complex and customized virtual infrastructures on multiple Cloud providers. For detailed description of the service see D6.1. and https://www.grycap.upv.es/im/.

5.1.2.1   Service Interfaces

IM a web-based GUI, a XML-RPC API, a REST API and a command-line application.

5.1.2.2   Possible Integration Partner Services

IM is integrated with the following EOSC-hub Services: PaaS Orchestrator, EC3, EGI VMOps Dashboard, EGI AppDB and EGI Cloud Compute.

5.1.3   PaaS Orchestrator

The PaaS Orchestrator is a service that allows to 1) coordinate the provisioning of cloud resources and the deployment of virtual infrastructures on heterogeneous cloud environments like private clouds (OpenStack, OpenNebula) and public clouds (Amazon Web Services, Microsoft Azure); 2) manage the deployment of dockerized long-running services or the execution of dockerized batch-like jobs on top of Apache Mesos clusters.

For detailed description of the service see D6.1. and https://indigo-dc.gitbooks.io/indigo-paas-orchestrator/content/

5.1.3.1   Service Interfaces

The PaaS orchestrator exposes REST API endpoints documented at https://indigo-dc.github.io/orchestrator/restdocs/ ; request/response data are transferred in the compact and easy-to-use JSON data-interchange format.

The Orchestrator supports the TOSCA standard for describing the topology of the virtual infrastructures and the services to be deployed. The deployment requests submitted by the users to the orchestration tools must adhere to the TOSCA template syntax defined by the TOSCA's YAML Simple Profile that specifies a rendering of TOSCA providing a more accessible syntax as well as a more concise and incremental expressiveness of the TOSCA DSL (Domain Specific Language).

The adoption of the TOSCA standard ensures the portability of the deployment topology description across different cloud providers and the support of the cloud bursting use-case.

5.1.3.2   Possible Integration Partner Services

The PaaS Orchestrator is already integrated with the INDIGO IAM, the Infrastructure Manager, the CMDB and SLAM services and Onedata.

5.1.4   FutureGateway

The FutureGateway is a complete framework aiming to aid the creation of Science Gateways. It includes many components for installation and management. It provides a set of REST API calls to address final user interfaces. For detailed description of the service see D6.1. and GitHub main page at: https://github.com/FutureGatewayFramework

5.1.4.1   Service Interfaces

The most important part of FutureGateway consists of its RESTful API calls; they are intended to address distributed computing resources using three logical entities named: *Infrastructures*, *Applications* and *Tasks*. The Task element consists of application instances, running on top of a given distributed infrastructure.

FutureGateway provides services to install and maintain the system and encourages its customisation in order to best fit the adopter's needs.

In accordance with the FutureGateway architecture the current implementation foresees the following software components:

- **fgSetup**: Collect scripts and procedures to install and maintain FutureGateway.
- **fgAPIServer**: Python based implementation of the FutureGateway APIs.
- **fgAPIServerDaemon**: A daemon process that addresses physical distributed resources. This component uses a set of sub-component named: **ExecutorInterfaces**, that can be developed in order to target any kind of distributed environment. At the moment, the available executor interfaces are: '*ToscaIDC*' and '*Grid and Cloud Engine*'.

ToscaIDC is integrated with the EOSC-hub Service INDIGO PaaS Orchestrator.

The Grid and Cloud Engine provides a multi-infrastructure solution since it uses the jSAGA library to approach many computing infrastructures.

Potentially the FutureGateway framework can be adopted by any community requesting capabilities offered by a classic Science Gateway. The FG APIs can be also used for Mobile applications, IoT, Workflow engines and More recently the FG APIs have been also used in conjunction with OpenAccess repositories, demonstrating an infrastructure-agnostic solution to implement a Science Reproducibility and Reusability Platform.

The fgAPIServer can be integrated with EGI AAI, INDIGO IAM as it is configured in the EGI Science Software on Demand Service developed in the context of EOSC-hub T10.3 activity.

5.2   Integration activities

5.2.1   INDIGO Orchestrator improvements

The INDIGO Orchestrator is being extended in the framework of the EOSC-HUB project both for ensuring better performances and enhanced reliability and for supporting the new requirements coming from the thematic services.

In the first year of the project, in order to improve the stability and scalability of the Orchestrator service, a new Workflow Manager, Flowable, has been integrated in the Orchestrator replacing the old jBPM engine. Flowable (https://www.flowable.org/) provides a workflow and Business Process Management (BPM) platform that is faster and more reliable.

The definition and management of the workflows implemented in the Orchestrator for creating/updating/deleting deployments have been revised as well and enhanced in order to better address the possible failures.

A retry strategy has been implemented in order to recover from:

- Failures of a single step in the running workflow, e.g. a glitch in the communication with a specific service;
- Failures of the whole deployment process at the selected site: in this case, the deployment is retried using the other available sites following the order specified by the ranking algorithm.

Moreover, as requested by the DODAS thematic service, the resources created for a deployment can be preserved in case of failure. Before this modification, the resources allocated for the deployment were deleted in case of failure; this behaviour was not acceptable in some cases: DODAS can spawn virtual clusters that need huge amounts of compute resources; with the old approach, if a failure happens when almost all the resources have already been allocated and are up and running, the whole cluster is deleted. Whereas, with the newly implemented solution, the user will be able to keep the cluster with the available resources and decide about the next operations (deletion, update, etc.).

In order to exploit sites that provide accelerators, the Orchestrator is now able to deploy containers on top of Apache Mesos clusters that provide GPUs as cluster resources.

Examples of TOSCA templates for long-running services and batch-like jobs requiring GPUs are available on GitHub at https://github.com/indigo-dc/tosca-templates

In the second year of the project, the Orchestrator functionalities have been further improved and extended:

- The support for the deployment of Virtual Machines with GPU(s) has been added: the user can request one or more GPU for his virtual server and the Orchestrator will select the sites that provide GPU-enabled flavors in order to match the user requirements;
- The integration with Hashicorp Vault has been implemented in order to safely manage the user secrets for the services deployed on top of Marathon/Mesos clusters;
- A new plugin has been implemented in order to support the submission of batch jobs on HPC sites through the QCG Computing gateway: this first implementation provides a prototype for the PaaS integration with HPC environments; the goal is to facilitate the access to HPC computing resources using cloud-like interfaces;
- The support for hybrid deployments has been further improved exploiting the INDIGO vRouter component.
- Finally, the overall deployment workflow has been consolidated, improving the interfaces with the auxiliary services (CMDB, Monitoring, Cloud Provider Ranker, etc.).

5.2.2   Infrastructure Manager evolution

In the first year of the project:

The IM has been evolved in the framework of the EOSC-Hub project. The first step was to test the authentication systems provided in the project. IM was already integrated with the INDIGO IAM (based on OpenID) and it has been successfully tested with EGI Check-In system (also based on OpenID) without any code modification.

As requested by the DODAS Thematic service it has been added the support for EC2 spot instances. EC2 spot instances were already supported in the AWS IM connector, but TOSCA support has been added. This required adding the "preemptible_instance" property in the TOSCA compute node definition and the proper translation in the IM orchestrator core.

Also, the Exoscale provider (CloudStack API) has been added as requested by the EGI Applications on Demand service to access this platform as part of a collaboration with the project HNSciCloud.

Another extension made to the IM is the ability to use SSH reverse tunnels to connect with VMs that only feature private IP addressing in different Cloud providers. This enables to contextualize hybrid deployments, that is virtual infrastructures deployed across multiple IaaS Cloud sites, using only one public IP per infrastructure.

Regarding the extension of the TOSCA types, two main contributions have been made: Kubernetes and JupyterHub. A set of Kubernetes node types have been defined to allow the user to specify a TOSCA document describing a Kubernetes cluster. This enables the user, together with the Ansible roles commented in section 4.2.6, to automatically provision a fully functional Kubernetes cluster. This eases the Kubernetes cluster as a Service functionality required in the ELIXIR CC.

The deployment includes a NFS configuration to create persistent volumes. Furthermore, a JupyterHub node has been defined to launch this application as a standalone application or on top of a Kubernetes cluster to spawn the Jupyter notebooks as pods of the cluster.

A public highly available instance of the IM has been deployed at UPV. It is deployed on a dedicated Kubernetes cluster of three nodes (one master and two working nodes). It has deployed ten instances of the IM container using an HAProxy to balance the load among them. Also, the IM web portal has been deployed in this cluster using a nginx service to separate the application deployed. Furthermore, this endpoint has been published in the EOSC marketplace. The URLs are the following:

- URL REST API: https://appsgrycap.i3m.upv.es:31443/im/
- URL Web portal: https://appsgrycap.i3m.upv.es:31443/im-web/
- EOSC   Marketplace   entry: https://marketplace.eosc-portal.eu/services/infrastructure-manager-im

In this second year of the project:

EGI Cloud Compute is deprecating OCCI and migrating to OpenStack APIs. IM already supports OpenStack but some improvements have been made in the connector to add missing features to correctly manage VMs for tools such as EC3 AoD or EGI AppDB VMOps dashboard. Firstly, the support of AppDB URLs in the OpenStack connector (as already implemented in the OCCI connector)

has been added. This enables users to get the final VM ID at the OpenStack site. The second improvement regarding the OpenStack connector has been to enable adding/removing floating IPs and volumes to OpenStack running VMs. Finally, the process of network creation has been improved enabling CIDR selection from the existing created networks.

In order to offer hybrid deployments that can be deployed across multiple IaaS Cloud sites, TOSCA support was enhanced to include the private network mapping. This allows users to choose virtual private networks that are used when provisioning virtual infrastructures across multiple Cloud sites.

Finally, support has been included for the creation of networks in OpenStack, GCE and EC2 connectors (the Azure connector already supported it). This enables the user to create private networks in the public cloud providers and in the OpenStack sites that support it (in some cases site administrators do not allow this).

5.2.3   FutureGateway extensions for EOSC-hub

FutureGateway has been actively developed in the EOSC-hub project to improve its configurability. The efforts have been focused on a few key directions.

The first one is an enhanced protocol for container image creation. FutureGateway is a multi-component service. Even though these components are loosely coupled by design, the installation procedures did not fully benefit from that. To address this issue, Ansible roles have been developed with special care taken in order to make them usable by the Ansible Container subproject. The provisioning of virtual machines and container image creation has a lot in common, but there are important differences to be accounted for e.g. lack of system services in the containerized system by default. Nevertheless, Ansible roles can be prepared to work in VM and container scenarios. In the EOSC-hub project, some components of FutureGateway have been prepared in this way. They work together in a common environment as created by Docker Compose: https://github.com/FutureGatewayFramework/fgSetup/blob/master/docker/setup_futuregateway.sh. The effort for the other components is ongoing.

The second main direction is stability and ease of maintenance. FutureGateway has been updated to fix some of the issues reported previously. The documentation has also been improved with focus on future goals to achieve.

One of the FutureGateway clients is a module to Kepler – a scientific workflow system. The module contains actors i.e. entities with well-defined inputs and outputs and processing logic embedded in them. Actors are reusable elements of the scientific workflows and their inter-connections define the overall computing process with conditional execution and looping mechanisms. The Kepler module brings bindings to FutureGateway REST API and provides Java classes to represent objects in the FutureGateway domain, such as *tasks* or *infrastructures.* In the EOSC-hub project, the module has been updated to be up-to-date with upstream changes. Additionally, the TOSCA template to deploy a Docker image with Kepler and run a workflow has been published.

In the second year of this project, FutureGateway components have been updated as a maintenance measure. First, a module for Kepler scientific workflow system, which is responsible for interaction with FutureGateway services, required newer versions of its dependencies due to security updates. Second, the base Docker image has been updated from Ubuntu Xenial (16.04 LTS) to Ubuntu Bionic

(18.04 LTS). Apart from that, the tool used to generate Docker images with Future Gateway components has been changed from Ansible Container (no longer maintained upstream) to Buildah - the official tool supported by Open Container Initiative (OCI). Buildah scripts are easier to maintain and more flexible, as they allow for incorporating conditional behaviour and interaction with the building environment. Images built with it have been tested and validated in Docker Compose and Kubernetes configurations.

5.2.4   TOSCA HEAT-Translator evolution

The evolution roadmap for the TOSCA heat-translator tool is primarily focused on maintenance activities. In particular, the active support on the OpenStack-related TOSCA templates from the INDIGO repository in GitHub: https://github.com/indigo-dc/tosca-templates. This includes the effective translation of recently added templates such as the EOSC hub's DODAS thematic service. In order to guarantee a successful translation, a continuous integration (CI) pipeline has been set up in Jenkins (https://jenkins.indigo-datacloud.eu:8080/job/Pipeline-as-code/job/heat-translator/job/devel/). Currently, this pipeline triggers a translation check for each change in the heat-translator code. Future plans include extending the aforementioned checks whenever a change is made in any of the TOSCA templates meant to be deployed in OpenStack Heat.

As a result of the maintenance work, new enhancements have been added to the codebase i.e. the support for the normative type `tosca.nodes.SoftwareComponent` and an upstream contribution to the OpenStack code repository that prevented log messages from being lost when using the translator through the OpenStack client.

5.2.5   Implementation of EOSC-hub requirements in CMDB and SLAM services

CMDB (Configuration Management DataBase) is an INDIGO PaaS platform component providing other components with technical information needed to run services from TOSCA templates. CMDB consists of a database and programmatic interfaces for Orchestrator, SLAM and Cloud Provider Ranker, allowing for retrieval and management of Sites, Services, and Images provided by the underlying computational infrastructure.

As it was revealed in a study of technologies used in the EOSC-hub project environment, there are some overlaps between CMDB and AppDB that is already used in EOSC-hub – playing a role that is similar to CMDB for the community built around the EGI infrastructure. Although AppDB seems a mature, more widely adopted tool with a potential to be a replacement for CMDB, there is some data missing in AppDB, from the INDIGO PaaS point of view (such as public providers, information about networking, etc.). Some architectural concepts such as the method for data acquisition (push vs pull) are also different in these two solutions, what makes using AppDB in this context a non-ideal solution.

With interface unification in mind, a feasibility study of CMDB redesign with the use of AppDB publisher has been done. This way, a PoC version of CMDB based on AppDB source code has been equipped with an interface that is widely adopted in EGI–centric services of EOSC-hub, maintaining the possibility to introduce changes that are essential for the other components. In order to fulfill requirements coming from the INDIGO PaaS environment, information schema, database contents and programmatic interface of AppDB needed to be extended. Unfortunately integration tests of

the PoC implementation with other components (Orchestrator and Cloud Info Provider) showed that this approach of the CMDB development is not as promising as expected. On the one hand the scope of changes needed to be applied to the Glue Schema (used by AppDB) makes the schema barely usable in the context of the PaaS system. On the other, trying to fit the new data (VM flavors, cloud tenants) into the Glue Schema makes the data badly structured. Extending the Glue Schema is not a viable option for long term development, so this direction of development becomes infeasible. As the changes of the data schema are easy to be introduced and maintained in the proprietary CMDB format, sticking with it seems to result in a more reasonable solution for the PaaS infrastructure. Therefore, in the next development cycle the CMDB development will focus on providing a working use case with integration of adjacent infrastructure components (CIP and Orchestrator).

SLAM (Service Level Agreement Manager) is an INDIGO PaaS component that allows for negotiation between customers and infrastructure providers. By using SLAM customers are able to reflect their requirements for the infrastructure properties (SLA targets), providers in turn, are able to reflect their readiness to fulfill these requirements. Once Service Level Agreement is met, other components (such as Orchestrator) are able to use it for automatic service provisioning, as a set of customer requirements and provider restrictions for the underlying infrastructure.

SLAM uses CMDB as a registry of sites, services, and images. Connection between the components is established via the CMDB API. As the redesign of CMDB API imposes the need for SLAM source code adaptation, the changes were introduced using CMDB API based on the GraphQL protocol (compliant with the AppDB API). As we resigned from taking this path in the CMDB development the SLAM interface needs to be adapted as well.

6   Data and Metadata Management

The EOSC-hub common repository services and the policy-driven data management/stewardship services with particular regard to registered data, which are data associated with persistent identifiers, are described in details in the following paragraphs and shown in the picture (Figure 2).

Those services allow users to store a data set in a repository, which is geographically distributed, and associate a persistent identifier with it, making the data set location independent from the references pointing to it. The identifier is globally resolvable, and the data set is replicated in multiple copies, which are tracked in the metadata associated to the identifier. Data can be published, and community specific metadata associated with it, then those metadata can be harvested and indexed by a discovery service to make the data findable. Data can also be annotated, manually or programmatically via API. And last, but not least, data are curated through a set of policies that each data manager can define.
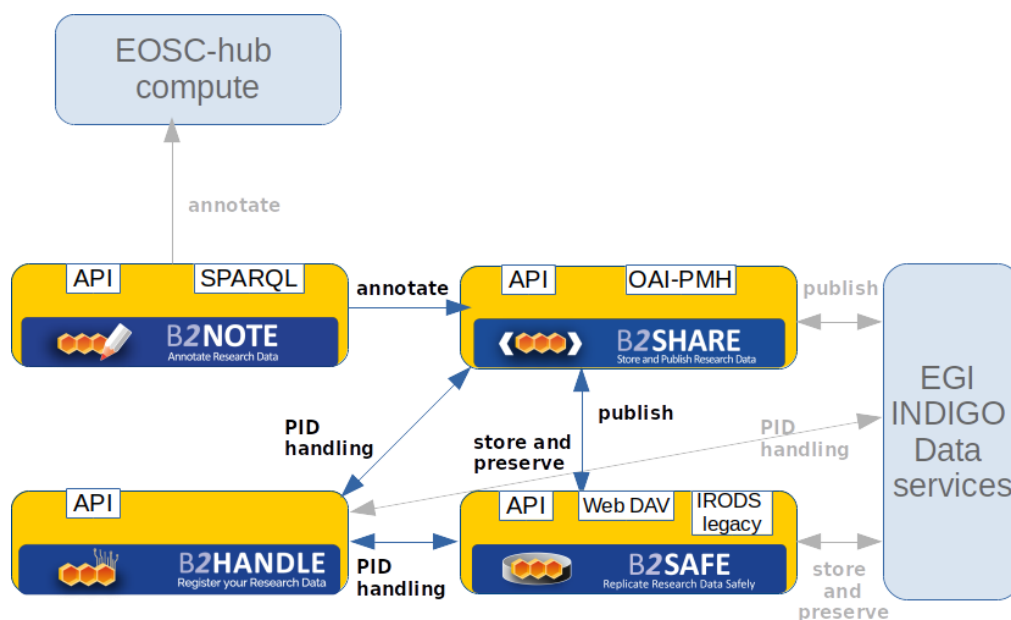

Figure 12. Data and metadata management services

6.1 Maintenance, interfaces and integration options of the services

6.1.1   B2HANDLE

B2HANDLE is a service for provisioning of Persistent Identifiers (Handles), including provision of namespaces (prefixes), hosting of Handle servers and additional server components for search and programmatic access. For detailed description of the service see D6.1. and https://eudat.eu/services/userdoc/b2handle

To accommodate requirements of EOSC-hub, B2HANDLE has improved some of its components already. Most importantly, an update to the key component for the Central PID Catalog, the Handle Reverse-Lookup Servlet (HRLS), has been released (v1.0.4) to accommodate requirements for metric

measurement as part of virtual access and extended monitoring. Also, the component was updated to be compatible with a recent release of the Handle System component (v9.0).

The b2handle library has been improved as well and a pre-release (pre-1.1.2) has been released, improving documentation, providing fixes and minor functionality updates. A full release is expected in early 2020 that will, among other things, introduce support for Python 3.6 and 3.7. A number of issues raised on GitHub are also addressed.

6.1.1.1 Service Interfaces

B2HANDLE offers multiple APIs to enable CRUD (create, read, update, delete) operations on Handles. These APIs include, aside the native Handle interface, easy programmatic access via the b2handle/pyhandle Python libraries, and support for the built-in REST API. The additional server component for searching (HRLS) offers its own API, which is accessible easily together with the main Handle API via the b2handle/pyhandle libraries as a coherent interface.

6.1.1.2 Possible Integration Partner Services

Increased integration of B2HANDLE with B2SHARE has taken place with a specific focus on supporting key metadata elements in a workflow that relies on B2SHARE to include metadata elements from B2HANDLE. B2HANDLE has also been integrated with EGI services. The DataHub uses B2HANDLE to assign and retrieve persistent identifiers for user data. Through this integration, also the federated data manager and online storage are being addressed.

6.1.2 B2SAFE

B2SAFE is a long-term preservation and policy based data service. It allows community repositories to implement data management policies on their research data that is distributed across multiple administrative domains.   For detailed description of the service see D6.1. and https://www.eudat.eu/b2safe .

6.1.2.1 Service Interfaces

B2SAFE is accessible through various interfaces.

- IRODS legacy: the interfaces offered by the iRODS component. They are CLI and API, java, python and C (https://irods.org). Moreover a WebDAV interface is available as a separated component (https://github.com/UtrechtUniversity/davrods)
- HTTP API: it is a RESTful interface which exposes functions to upload and download data (see above the paragraph B2STAGE)
- GridFTP: a bridge between iRODS and the GridFTP service is available as explained in the paragraph about B2STAGE. That allows users to upload and download data relying on the high performance transfer features of GridFTP.
- Data Policy Manager web UI: the user web interface that allows to define data policies and store them in a DB, from where they can be distributed to multiple B2SAFE instances.
- Data Policy Manager REST API (BaseX API): the HTTP API of the BaseX XML DB component, which stores the data policies XML documents.

6.1.2.2   Possible Integration Partner Services

- B2SAFE can be integrated with EOSC-hub Service DataHub.
- B2SAFE can be integrated with EOSC-hub Service B2DROP.
- B2SAFE can be integrated with EOSC-hub Service B2SHARE.
- B2SAFE is integrated with EOSC-hub Service B2ACCESS.
- B2SAFE is integrated with EOSC-hub Service B2STAGE.

6.1.3   B2SHARE

B2SHARE is a data storage and sharing service for research communities and individual researchers. It allows discovery and publication of research datasets by providing detailed descriptions in the form of standardized metadata. For a detailed description of the service see D6.1. and the EUDAT website.

6.1.3.1   Service Interfaces

B2SHARE is accessible through a web interface and a REST API that allows a user to create, modify and manage records. The service is integrated with B2DROP to allow direct uploads, B2HANDLE to mint new handles, B2NOTE for additional annotation of files in records and B2ACCESS for authentication.

For metadata harvesting, an OAI-PMH endpoint is available that supports multiple metadata prefixes for compatibility with B2FIND, OpenAIRE RCD and other metadata catalogues.

6.1.3.2   Possible Integration Partner Services

B2SHARE can be integrated with EOSC-hub Services from EUDAT, OpenAIRE, EGI and INDIGO.

Effort has been put in design documents that describe the requirements and necessary changes to B2SHARE in order to improve the interfacing to OpenAIRE Community Dashboard, EGI DataHub and Online Storage and B2NOTE.

For monitoring purposes, a Nagios plugin has been developed and released that interfaces with the EUDAT Argo monitoring service.

6.1.4   B2NOTE

B2NOTE is a data annotation service integrated with data repositories/data publication services. It allows the service users to add extra information without modifying the underlying data record. Annotations are structured using the W3C Web Annotation data model, serialized in JSON-LD and stored in a document database (MongoDB). These annotations can then be used to organize and retrieve datasets based on the user's needs. For detailed description of the service see D6.1. and in Kulhanek and Le Franc (2019)[15]. The version 3 of the service will be released in production in the upcoming weeks. This new version of the service based on javascript offers now two additional ways of integrating the service: javascript client integration and web component integration.

---

[15] Tomáš Kulhánek, & Yann Le Franc. (201 9). Service for Digital Annotation of Scientific Data. Zenodo. http://doi.org/10.5281/zenodo.3369156

6.1.4.1 Service Interfaces

B2NOTE offers two different Interfaces:
1. a User Interface available as a widget for the integration within the User Interface of partner services and
2. a RESTful API to initialize the annotations and retrieve stored annotations.

The User Interface has been designed to offer functionalities for the creation, the management and usage of annotations despite the reduced size. This interface should be extended to offer more convenient functionalities for users. For this purpose, we are designing a central UI to provide more space for users to work with their annotations.

The RESTful API of the latest version offers a large set of functionalities to access the content of annotations and is now secured, allowing for the creation of private annotations.

6.1.4.2 Possible Integration Partner Services

- B2NOTE is integrated with B2SHARE and this integration can be improved (see Integration activities).
- B2NOTE can be integrated with B2FIND.
- B2NOTE can be integrated with EGI DataHub.
- B2NOTE can be integrated with community services such as the CLARIN Virtual Language Observatory, Herbadrop.
- B2NOTE can be integrated with OpenAire data services such as Zenodo, OpenAIRE search and the Research Community Dashboard.

6.2 Integration activities

6.2.1 B2SAFE

6.2.1.1 Integration improvements between B2ACCESS and B2SAFE

During the first year the integration between B2SAFE and B2ACCESS was consolidated. It was already achieved before the beginning of the project, but at certain point it was broken because of an update of the B2ACCESS policies. Therefore we re-implemented it adopting a different approach based on the PAM (Pluggable Authentication Module): the openID token, got by the user from B2ACCESS, is passed to iRODS as the password parameter during the authentication process, it is intercepted by a custom PAM module which acts as a B2ACCESS client, contacting the validation endpoint and getting back the attributes of the user, in particular the email address. The email address is used to map the global user to the local B2SAFE user and give access to the data (https://github.com/EUDAT-B2SAFE/pam-oauth2).

6.2.1.2 Extension of Data Policy Manager

The Data Policy Manager (DPM) is a service that is part of the B2SAFE service. The original intention was to provide a service that makes it easy for users to define policies for managing their data in a service-implementation independent manner. Currently the main interest for users of the B2SAFE

service is for replication policies. The DPM provides a web interface for creating policies that are stored in an XML database and a REST API to access the XML policies. Client software has been written to transform the XML documents into iRODS rules. The client software is currently being updated to make use of the iRODS API and is being rearranged to also accommodate the HTTP API.

6.2.1.2.1   For EGI services

An investigation into the interoperation of the DPM with onedata (https://onedata.org) was briefly carried out where the data policy manager could be used to create a policy to initiate a replication to onedata from B2SAFE (or vice-versa). At the moment the functionality provided by the DPM does not meet the needs of onedata and further work is paused.

6.2.1.2.2   For EUDAT services

Work has started on integration of B2FIND and the DPM based on a use case from Norway to regularly populate B2FIND with metadata from the Norwegian Research Data archive. The Norwegian Research Data Archive provides an archiving service for Norwegian-funded research data. The datasets are primarily publicly accessible and issued with DOIs. The goal is to have a policy that supports regular harvesting of metadata from the archive. The implementation neutral policy makes it easy to transform the policy should the infrastructure change (this is something envisaged for the archive in the short-term). The policy is currently being modelled based on the workflow for populating B2FIND.

Replication policies for the CompBioMed community are described below. Replication policies for other communities are also under development. The replication policy has been tested in a test environment and the setup of the production infrastructure is currently underway.

CompBioMed data replication use case

The solution encompasses the definition of a data pipeline and of the related data policy as described below, while the requirements are detailed in the paragraph Use Cases.

Data Pipeline

The data pipeline includes the following major steps:

1. **Data creation and transfer:** The raw data is collected at ESRF (European Synchrotron Radiation Facility) in France. The data is being stored locally on tapes. Currently, a copy of the data is transferred to BSC.
2. **Data pre-processing:** In BSC researchers pre-process the data which includes manual and automated steps for image stitching, segmentation, and meshing
3. **Data Replication:** The preprocessed data needs to be replicated from BSC to SURFsara and EPCC. The replicated data will then be used to run simulations with the Alya software which is installed on the supercomputers in these sites (i.e. Cartesius in SURFsara).

The graphic below shows the data workflow, services and centers that are involved:
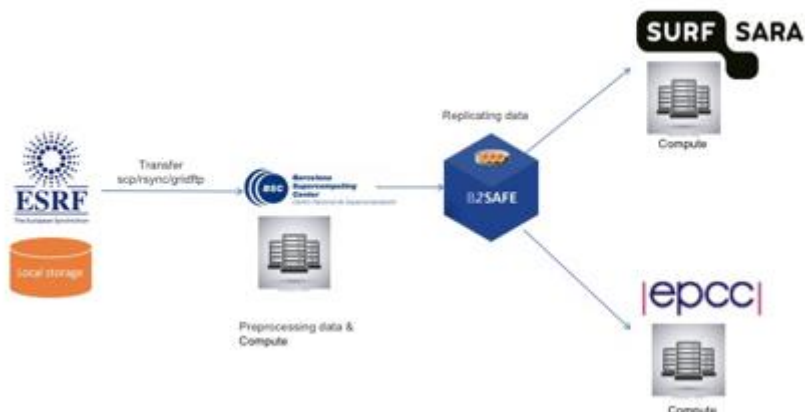
Figure 13. Data workflow, services and centers involved

Replication policy:

The replication policy for the CompBioMed use case was defined in the EUDAT DPM tool. The generic replication policy schema of B2SAFE covered the replication requirements of the community, including having two replicas of data on both disk and tape.

Activities

Activities	Description	Status
Identify the concrete use case	- Define data pipeline - user workflow - services involved	Done
Administrative tasks	- Requesting resource allocation - Requesting access - Assigning technical contacts at each data/compute center	Done
Technical setups and Configurations	- installing B2SAFE - enable resources - create accounts - Set access rights	In progress
Define and enforce the replication policies	- Define the replication using EUDAT DPM tool (https://dpmgr.eudat.eu) - Enforce the policy in the source compute center (detailed instructions for creating and enforcing a policy: https://docs.google.com/document/d/1K8-tF_eSiI0itCKUsbxYkuS6MVT1eGnXs0nnaWFRXnk/edit#)	In progress

Replicate data	- The source data center (BSC) initiate the replication to replicate data	Not started

6.2.1.3   Extension of B2SAFE with other persistent identifiers used in EOSC-hub

This task did not have any activity during the first year due to the lack of requirements. It still does not seem that the integration with other kinds of persistent identifiers is a priority at the moment, therefore it has been postponed.

6.2.1.4   Integration improvements between B2HANDLE and B2SAFE

At the moment B2SAFE uses a separate python client and python library to interact with the B2HANDLE service. To improve performance and clean up the iRODS B2SAFE code a few new iRODS uServices are being developed. The new iRODS uService can take over the implementation using the python client and python library. Once development/testing is finished it will be part of the B2SAFE rules if the user installs the uServices.

6.2.1.5   B2SAFE data discovery and access

The data discovery is enabled through the integration with B2SHARE. As PoC of such integration a number of python scripts and iRODS rules were created. This needed to be improved and tested, so that they fulfill the requirements of EOSC and are stable and production release ready, so they could be integrated into B2SAFE.

The development of the connection component and improvement possibilities were coordinated with the B2SHARE team in a number of meetings.

Unit tests were developed that are testing all methods in python scripts of the connection component for normal and error cases. All unit tests are combined in a python test suit. The development of the unit tests lead to an improvement and some refactoring of the python scripts of the connection component.

For the integration tests a set of mock data on a productive B2SAFE instance was created with files having actual PIDs, which is essential, as B2SHARE validates the list of the PIDs B2SAFE is trying to send during the draft creation. The HTTP API of the training instance of B2SHARE was called for these tests. Prior to this the API was extended by the B2SHARE team to be able to process the B2SAFE call for a draft creation with a list of PIDs for files, which a collection in B2SAFE has.

The documentation of the B2SAFE-B2SHARE connection component was extended to contain the description and execution example of the unit tests.

Code and the documentation of the B2SAFE-B2SHARE connection component and the corresponding unit tests are to be found at

https://github.com/EUDAT-B2SAFE/B2SAFE-core/wiki/B2SHARE-connection-component

6.2.2 B2SHARE

6.2.2.1 B2SHARE extensions for diverse data organizations

Several communities have been supported and effort has been put into enabling dataset publication workflows:

- InGrid: a community schema has been developed and implemented.
- IBPT: support for instancing, community integration and schema definition
- LOFAR: support for community integration and schema definition
- EPOS: enabling of community
- HPC-Europa3: enabling of community
- TBopen: enabling of community
- OPENCoastS: enabling of community
- CompBioMed: enabling of community
- STARS4ALL: enabling of community, additional development

- EOSC-Hub: enabling of EOSC-Hub's own community

6.2.2.2 Initial B2SHARE integration with EOSC-hub services

Concerning integration with other EOSC services, the following activities have taken place with regard to B2SHARE:

- Expanded integration with B2HANDLE has been discussed for the display of PID profile information. This is a community-specific request and requires extension of B2SHARE handle information fetching and some redesign of the record landing page. A design and development document has been written that covers the required changes and effort.
- Integration with EGI DataHub has been discussed with the DataHub team to allow direct data uploads from DataHub to B2SHARE. A design and development document has been written that covers the required changes and effort. A technical integration workflow has been designed that describes the required API calls to EGI DataHub from B2SHARE to enable data transfers.
  - A similar workflow will be designed to integrate EGI Online Storage and INDIGO-DataCloud
- A monitoring probe has been released that pushes monitoring data to the central EOSC ARGO monitoring service.
- Extended functionality regarding dataset and file annotation with B2NOTE has been discussed and prioritized. Development effort will add annotation for datasets, display of annotation count on record landing pages and clickable links to show current annotations.

6.2.3 B2NOTE

The integration work for B2NOTE started at M7. The initial effort has been focused on discussing and evaluating the different possible integration scenarios either directly with the concerned service team or as initial internal work. We chose to focus our attention to the improvement of the existing integration with B2SHARE to address existing user requests, to extend the integration OpenAIRE Search.

6.2.3.1   Improve two-ways integration of B2NOTE with B2SHARE

We worked directly with the B2SHARE team to define the necessary updates for the integration. Four additional features have been identified:
1. Showing the number of annotations associated with B2SHARE data elements and datasets;
2. Extending annotation to datasets;
3. Showing the annotations associated with the data elements and datasets in B2SHARE;
4. Integrating annotations in the B2SHARE search engine.

During the reporting period, we started to work on defining the roadmap, identifying the key tasks to be executed to implement these changes and the technical implications. A well-defined roadmap has been proposed and a clear documentation of the integration with UI integration mockups and the necessary associated REST API calls has been provided by the B2NOTE development team. The integration within B2SHARE code are in progress.

6.2.3.2   B2NOTE Integration with OpenAire Search

We started to evaluate internally initial requirements for the integration of B2NOTE with OpenAIRE services such as Zenodo, the OpenAIRE Search and the Research Community Dashboard. As the resources were limited on our side, the interaction with OpenAire teams have been limited during the initial reporting period. In the past 6-8 months, we started discussing with the team in charge of the OpenAIRE Search interface which provides search capabilities on the OpenAIRE Knowledge Graph. After several meetings to understand each service, we agreed on a roadmap for the integration of B2NOTE with a developer documentation providing UI integration mockups, REST API calls and the process to integrate B2NOTE as an iFrame. A successful test of this integration has been achieved in December but due to several unforeseen technical issues, the final version of the integration has been delayed. We are now waiting for the release of the new version of the service to finalize the initial integration and push it on the Beta version of the OpenAIRE search interface.

6.2.3.3   Integration with other services

At the moment, we only managed to initiate discussions with different services for potential additional integrations. In particular, we discussed the practicalities of the integration during the EOSC-Hub week in Prague with the developer team of EGI DataHub. One of the requirements which was clearly identified was the need to have a javascript client to support the integration as well as the possibility for a local deployment. These two requirements guided our choice in the service update to a full javascript version (easier to maintain) and the creation of a B2NOTE Docker instance (easier deployment). Unfortunately, we haven't been able to continue the discussions during this reporting period.

We also discussed with B2FIND about a potential integration which will be made possible with the transition to the new javascript version. As soon as the new version is released, we will get in touch with the developer team to start a Proof of Concept integration. This integration could be used in the context of the Herbadrop use-case to offer additional functionalities for the users.

6.3 Future Integration Plans

This section presents the overview of new or improved features achieved by extending or integrating existing services, and their relevance for thematic and specialized services.

- Normal software and service maintenance activities; update B2SHARE and its underlying frameworks and software dependencies as needed.
- Implement more record metadata exporters (e.g. support for exporting metadata as Datacite XML).
- Make necessary changes and additions to B2SHARE to enable harvesting B2SHARE metadata to OpenAIRE RCD.
- Further development of B2SHARE to enable and support new communities to start using B2SHARE.
- Improve integration with B2HANDLE service for displaying PID metadata in B2SHARE.
- Improve two-way integration with B2NOTE and B2SHARE.
- Integrate with suitable, non-EUDAT services that are part of EOSC-Hub service catalog. Possibly, for example, with EGI DataHub and EGI Online Storage.
- Finalize the integration between B2STAGE and B2SHARE
- Investigate the integration between B2STAGE and EGI DataHub
- Finalize the integration between B2SHARE and B2SAFE.
- Complete the data transfer tests between B2SAFE and DataHub.
- Finalize the integration of B2SAFE with B2DROP, through the WebDAV interface.
- Complete the tests of the Data Policy Manager to support the CompBioMed use case and the ICEDIG use case.
- Finalize the update of the Data Policy Manager client.
- Extend the data policies to support further services and communities.

7 Summary and Outlook

In this deliverable we have presented the second integration and maintenance plan of EOSC-hub common services catalogue. These integration activities cover services from 3 different and evolving e-Infrastructures namely EGI, EUDAT and INDIGO. The evolving nature of these services means that integration activities are always ongoing, following the service evolution and also taking advantage of new functionality. Furthermore, increased focus is now being placed on finding new community use cases and finally on contributing to the EOSC-Hub technical roadmap.

8 Appendix

1. ICEDIG milestone 39, 'Digitisation infrastructure test on EUDAT' : Milestone_MS39_ ICEDIG_Digitisation infrastructure test on EUDAT_v1.pdf at https://wiki.eosc-hub.eu/download/attachments/26416995/Milestone%20MS39_%20ICEDIG_Digitisation_infrastructure_test_on_EUDAT_v1.pdf?api=v2