D6.2 First report on the maintenance and integration of common services

Lead Partner:	DKRZ
Version:	1
Status:	Final
Dissemination Level:	PU
Document Link:	https://documents.egi.eu/document/3480

Deliverable Abstract

This deliverable provides overview of the maintenance and integration of common services in the areas T6.1, 6.2, 6.3 and T6.4 and T6.6. **The task T6.5, which started in project month 7, is not considered in this report.** 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 the plans for the next future.

COPYRIGHT NOTICE

DELIVERY SLIP

Date	Name	Partner/Activity	Date
From:	Heinrich Widmann Claudio Cacciari	DKRZ/WP6 CINECA/WP6	16/07/2019
Moderated by:	Malgorzata Krakowian	EGI Foundation/ WP1	
Reviewed by:	Shaun de Witt Roksana Różańska	UKAEA Cyfronet	6/05/2019
Approved by:	AMB		17/07/2019

DOCUMENT LOG

Issue	Date	Comment	Author
v0.1	26/11/2018	Draft table of contents	Heinrich Widmann (DKRZ)
v0.2	05/12/2018	Started discussion about structure and sections	Heinrich Widmann (DKRZ), Claudio Cacciari (CINECA)
v0.3	14/12/2018	Revised structure and table of contents	Heinrich Widmann (DKRZ), Claudio Cacciari (CINECA), Mattia D'Antario (CINCECA)
v0.8	29/01/2019	Added terminology, introduction and summary	Heinrich Widmann (DKRZ), Claudio Cacciari (CINECA)
v0.8	01/02/2019	Combined contributions from partners	Heinrich Widmann (DKRZ), Claudio Cacciari (CINECA), Mattia D'Antonio (CINECA), Johannes Reetz (MPG), Andrea Ceccanti (INFN), Claudia Martens (DKRZ), Enol Fernandez (EGI), Catalin Condurache (STFC), Jorge Gomes (LIP), Andrei Tsaregorodtsev (CNRS), 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), Sofiane Bendoukha (DKRZ)
v0.9	05/02/2019	Compile to word doc template	Claudio Cacciari (CINECA,

			Heinrich Widmann (DKRZ))
V0.10	08/02/2019	Overhand to the internal review	Heinrich Widmann (DKRZ), Claudio Cacciari (CINECA), John Kennedy (MPCDF)
V0.11	15/02/2019	Internal review	Heinrich Widmann (DKRZ), Claudio Cacciari (CINECA), John Kennedy (MPCDF)
V0.12	27/02/2019	Overhand to the external review (Shaun, Roksana)	Claudio Cacciari (CINECA), John Kennedy (MPCDF), Heinrich Widmann (DKRZ)
V1	16/07/2019	Updated after external review	Claudio Cacciari (CINECA), Heinrich Widmann (DKRZ), John Kennedy (MPCDF)

TERMINOLOGY

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

Terminology/Acronym	Definition
API	Application Programming Interface
CaaS	Computing as a Service
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
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 Catalogue1 for the first year of the project. In order to meet the needs of users, we first analysed the use cases in order to identify the requirements for the integration of the common services. In addition, the individual services were 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 work during the first year focused on providing a set of baseline services in four domains; Data Discovery and Access; Federated Compute; Processing and Orchestration; and Data and Metadata Management. Moreover, we have assessed the need for integration of services in both an intra and inter domain manner. In some cases, improved service integration was needed within an e-infrastructure while in others integration activities were required across e-infrastructures.

- Data Discovery and Access: The Data Discovery and Access task focused on providing a common data discovery and access layer which supports the FAIR data principles. This layer is formed of numerous services including B2FIND, EGI Datahub, INDIGO IAM, B2STAGE and B2DROP. These services were further developed, and several integration activities were undertaken including exposing data in EGI DataHub via B2FIND, integration of EGI DataHub and B2SAFE, B2STAGE and IAM to enable seamless data access and staging across services and integration of B2DROP with B2SAFE to retrieve and store small datasets.

- Federated Compute: The Federated Fompute activities focused on the continued development of the cornerstone services; Cloud Compute, Cloud Container Compute and High Throughput Compute with integration activities designed to improve service/task deployment and management including activities in the areas of AppDB, Cloudkeeper, Workload Manager and AAI.

- Processing and Orchestration: The Processing and Orchestration task focused on the integration of orchestration services with the federated compute services. The development activities have mainly been based around the use of TOSCA templates to describe deployments and services such as Infrastructure Manager, INDIGO PaaS and the FutureGateway to enable the deployment of complex environments or workflows. Additional activities were undertaken to improve integration with the Configuration Management Database (CMDB) and Service Level Agreement Manager (SLAM).

- Data and Metadata Management: This task focused on common repository service and policy driven data management/stewardship. The activities primarily focused on the further development and integration of services such as B2HANDLE, B2SAFE, B2SHARE and B2NOTE.

---

[1] For detailed description of the common services see the report of deliverable D6.1 First release of common services software

(https://wiki.eosc-hub.eu/display/EOSC/D6.1%09First+release+of+common+services+software)

In particular working on improving the authentication workflow with B2ACCESS and the interoperability among data services.

In addition to the technical obstacles, which were faced in each of the service domains, the following general problem areas turned out to be particularly critical:

- The lack of a common and easy-to-use AAI approach. While authentication and authorization were implemented for some specific cases, e.g. the use of Indigo IAM as AAI proxy for EGI-DataHub, a generic and easy-to-use AAI solution is missing.

- Suboptimal collaboration and communication with users and communities. Although we already put 'Use Cases' at the beginning as the driving motivation for the required integration, in the following project year the cooperation between WP7 (Thematic Services), WP8 (Competence Centers), WP10 (Technical Coordination) and WP6 (Common Services) should be enhanced.

- Integration of services from at least two of the e-Infrastructures merged in EOSC-hub - EGI, EUDAT and Indigo - proved difficult, as the technologies and approaches used were very different.

The most important next steps are therefore:

- The continued evolution/integration of the common services to suit end user needs as outlined in the work package description of work. More emphasis is to be placed on understanding the driving use-cases and cross work package communication.

- Promote and improve the integration activities, which have already been undertaken, to demonstrate them by applying them to real use cases and to generalize and adapt them to other thematic services.

- Explore possibilities to ease service integration across e-infrastructures. Looking at promoting standards, asking service providers to consider integration up-front during service design.

- Address the AAI issues in collaboration with WP5 where we will seek to achieve a generic solution that will offer seamless access to services across infrastructures.

1. Introduction

This document presents a report on the integration and maintenance of EOSC-hub services, focusing on the driving use cases, the status and the 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 first year of the project was focused on solving the most important issues related to integration of services such as enabling authentication and authorization mechanisms between EGI, EUDAT and Indigo platforms, unifying application interfaces, enabling data transfers and access among different services and improving services documentation. Thus, this deliverable provides initial integration notes about the services in the first year of the EOSC-hub project along with description of developments and improvements planned for the remaining part of the project.

The document is organized as follows. Since the problems and applications of the users result in the requirements for the integration of the EOSC services, we describe the driving use cases at the beginning in section 2. The following paragraphs are dedicated to the individual WP6 task areas, covering 'Discovery and Access' in section 3, 'Federated Computing' in section 4 , 'Processing and Orchestration' in section 5 and 'Data Management' in section 6 . Each of these area sections is in turn subdivided in sub sections for each service, explaining their 'Maintenance', 'Service Interfaces' and 'Possible Integration Partner Services', following by sections 'Integration activities', reporting on the work done within the first project year, 'Future Integration Plans' and 'Issues and Delay'. The entire report concludes with a 'Summary and Outlook' in section 7.

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). Following this, WP10 has analysed the use cases and helped the communities to talk with the WP6 developers, organizing meetings and opening tickets on the EOSC-hub JIRA system.

The use cases are derived 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. B2DROP can be used within different parts of ECAS.

The first one is the workflow framework Ophidia. 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 and the files, to be uploaded to B2DROP, within the operator configuration. 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. It was extended by ECAS to access two different kinds of storage. The first one is a shared space for all ECAS users and does not need further authentication of the 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 Orphidia 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

HPC Centers: DKRZ and CMCC

Resources: allocation of B2DROP storage depending on use case

Use case requirements:

- B2DROP account (optional, if only open sharing with all ECAS 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

2.2 Marine use case

The Marine CC[2] 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 goal was 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, EGI FedCloud for running the compute and analysis applications, B2DROP for collaborative analysis on shared data , Jupyter Notebooks for interactive analysis and B2SHARE for publishing the final result data and associated publications and reports.

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

HPC Centers: CSC, CINECA, Jülich and EGI FedCloud

Resources:

- allocation of 300 GB storage in B2SAFE

- Processed data to share must not exceed user quota of B2DROP

---

[2] https://wiki.eosc-hub.eu/display/EOSC/T8.3+Marine

Use case requirements:

- B2DROP and B2SHARE accounts
- Output data must not exceed user quota of B2DROP
- Optional: Notification of the user when processing is 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's 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: B2DROP for syncing the data between applications and collaborating on, EGI 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, EGI FedCloud, Kubernetes and B2ACCESS

HPC Centers: 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 quota of 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 in 'an innovative approach to long-term preservation and analysis of digitised herbarium specimens'[3] and is now being treated in the EU-funded project ICEDIG[4]. The project milestone 'Digitisation infrastructure test on EUDAT'[5] describes the integration of the services in detail.

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 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 Herbadrop's Elasticsearch repository via 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 service, which allows anonymous access on the B2SAFE 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 metadata field 'Source' offers a direct link to the digital object or – like in this example – redirects to a landing page of the institution maintaining the digital objects as shown in figures 2 and 3.

---

[3]   https://www.eudat.eu/herbadrop-an-innovative-approach-to-long-term-preservation-and-analysis-of-digitised-herbarium

[4] ICEDIG stands for "Innovation and consolidation for large scale digitisation of natural heritage" https://www.icedig.eu/

[5] https://wiki.eosc-hub.eu/download/attachments/26416995/Milestone%20MS39_%20ICEDIG_Digitisation_infrastructure_test_on_EUDAT_v1.pdf

Fig. 1 – B2FIND web page showing the result of a search based on tags

Fig. 2 – B2FIND dataset display with link to the resource ('Source')

Fig. 3 – Landing page of the Herbadrop resource the B2FIND identifier redirects to

Service: B2SAFE, B2HANDLE, B2FIND

HPC Centres: 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 mount point enabled by Oneclient command line tool.

Service: EGI-datahub, "Virtual Folder"

HPC Centers: EGI, CYPHRONET

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

HPC Centers: 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 [6](CMS), one of the two biggest and general purpose experiments at the CERN Large Hadron Collider[7] (LHC), and into the Alpha Magnetic Spectrometer [8](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

---

[6] https://home.cern/science/experiments/cms

[7] https://home.cern/science/accelerators/large-hadron-collider

[8] https://ams.nasa.gov/

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 Check-in). ESACO service is also part of the DODAS integrated service and this is responsible to guarantee 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, which guarantee selected access to the resources as well as data protection and privacy.

Service: Indigo-IAM, PaaS Orchestrator, WaTTS

HPC Centers: 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" which 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 Future Gateway 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 Future Gateway 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: Future Gateway, PaaS Orchestrator, Infrastructure Manager (IM)

HPC Centers: INFN

Resources:

- Deployment of a dedicated IAM and Orchestrator instances:
  - https://dariah-iam.cloud.cnaf.infn.it/
  - https://dariah-paas.cloud.ba.infn.it/orchestrator

Use case requirements:

- automated deployment of Invenio-based repository on Cloud environment

3. Discovery and Access

The overall objective of the task 'Discovery and Access' is the establishment of the *Common Discovery and Access Interoperability Layer* through which end-users can find, localize and use data resources within EOSC-hub for their own scientific purposes.
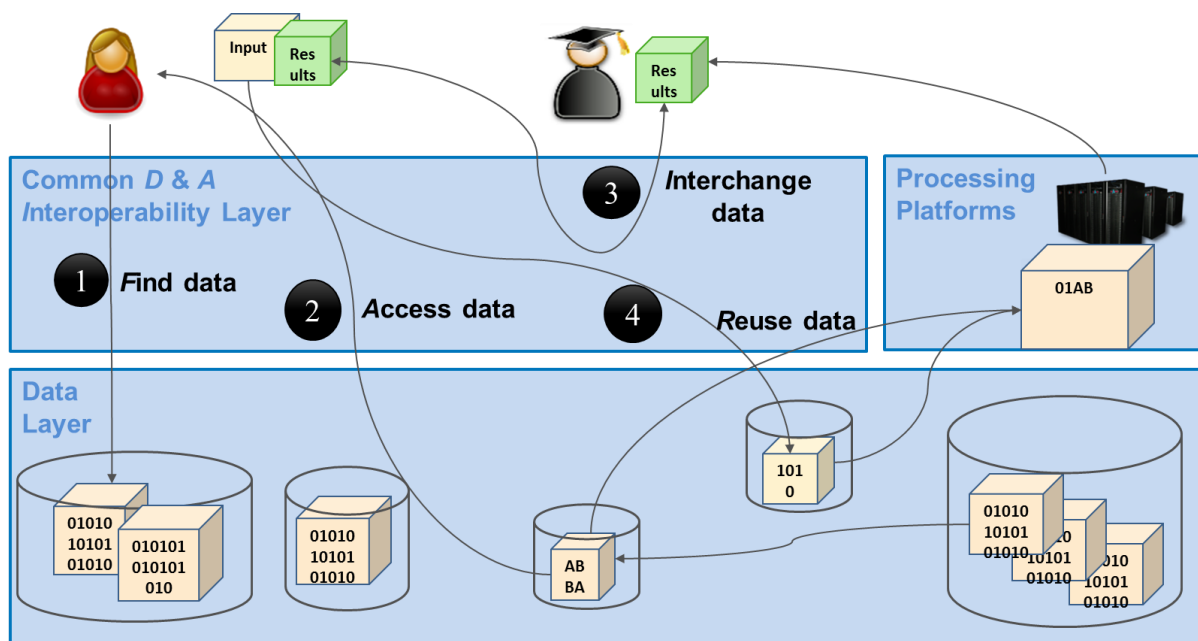
Fig. 4 - The Common Discovery and Access Interoperability Layer enabling FAIR data management

As shown in figure 4 end-users should following the FAIR principles be enabled to:

1. [F] search for distributed data in EOSC-hub and beyond
2. [A] seamlessly access distributed data resources wherever they are located
3. [I] interoperate by sharing and publishing research output
4. [R] reuse, exchange and stagedata (depending on the use case)

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 from storage services as EGIDataHub and B2SAFE and data archives within and beyond EOSC-hub. Regarding data access, we elaborated and assessed the possibilities and issues of seamless access to data collections in the storage services: EGI datahub and B2SAFE. Hereby the B2STAGE service helps to manage the data transfer and Indigo IAM is used as AAI proxy for EGIDataHub. Finally, B2DROP serves to synchronize and exchange user data with colleagues and other services. The realisation and implementation of the Common Discovery and Access Interoperability Layer follows a roadmap comprising work-package activities (WPAs), each describing the integration between a pair of common services from task T6.1.:

- WPA 6.1.3 & 4: Integration of EGI DataHub with B2FIND (Indexing and Discovery of EGI data resources)
- WPA 6.1.5: Indigo IAM integration for EGI DataHub (Authentication for access to EGI data)

- WPA 6.1.6: EGI DataHub integration with B2STAGE (Staging data from EGI to processing platforms)
- WPA 6.1.7: B2STAGE integration with B2SHARE (Retrieve processed data and store in B2SHARE)
- WPA 6.1.8 & 9: B2DROP integration with B2STAGE, EGI DataHub; and use B2DROP to prepare input data for B2STAGE and retrieve/store, small sized, data.

To achieve this, we defined an integration plan and identify work plan activities WPA 6.1.N as shown in the figure 5.
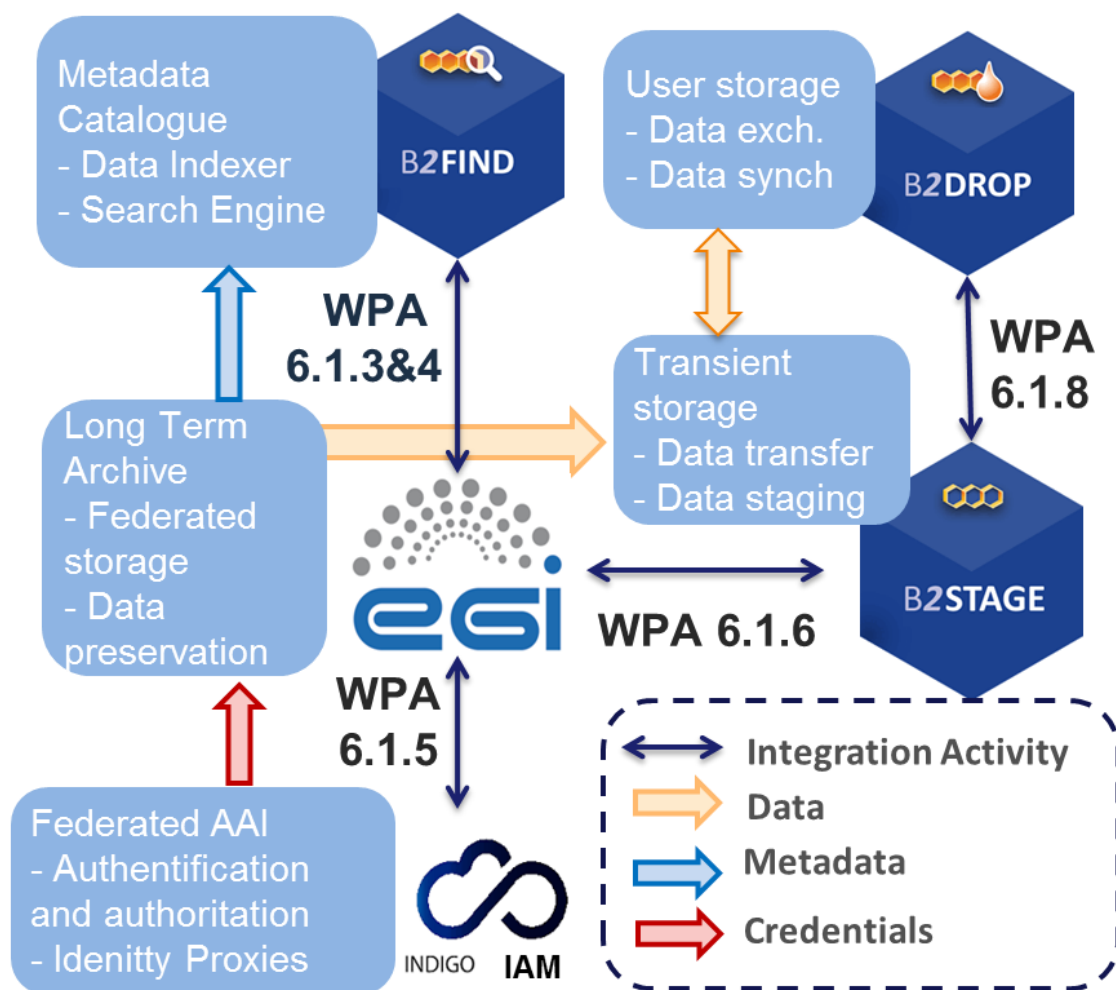
Fig. 5 - The integration plan of T6.1 with associated common services and work package activities (WPAs)

The figure includes only the five services from T6.1 with their pairwise integration activities and associated flow of data or information. In addition to these five services, services from other areas, e.g. B2SHARE from T6.4 which is used to publish, is also a part of the Discovery and Access layer.

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

The progress and status of the WPAs are treated in detail in the section 3.2 '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. While details are explained in the following sub sections, we provide a summary in table 1.

Table 1 : Overview over Services and associated Interfaces, Partner Services and Integration Status

Service	Interfaces	Partner Service	Integration Status
B2FIND	WebUI	B2SHARE	✓
	OAI-PMH	EGI DataHub	✓
		B2SAFE	O
EGI DataHub	POSIX	B2FIND	✓
	CDMI	B2HANDLE	✓
	WebUI	B2ACCESS	✓
		B2STAGE	O
INDIGO IAM	SAML	INDIGO PaaS	✓
	OpenID Connect	EGI Checkin	✓
	X.509	OneData	✓
	REST		
B2STAGE	GridFTP	B2SAFE	✓
	REST	B2ACCESS	✓
		B2SHARE	O
B2DROP	WebUI	B2ACCESS	✓
	WebDAV	B2SHARE	✓
		CLARIN Switchboard	O

3.1.1. B2FIND

B2FIND[9] 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 pillars 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 whereby 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, Dublin core, iso19139, json-config). As a prerequisite for an enhanced metadata ingestion workflow several software upgrades and extensions have been implemented:

- upgrade from CKAN 2.2 to 2.7
- upgrade from SolR 4.4 to 6.2.1

- upgrade from CentOS 6 to 7 (integrating CKAN required libraries and modules from CentOS repository)

New VMs for development and testing have been installed in order to enable a fruitful communication with data specialists regarding improved metadata quality and for testing technical developments. For meeting Community needs and to be compliant with other standards (Datacite) the B2FIND metadata schema was enhanced, including now e.g. <Contributor> and <alternateIdentifier>. These changes have been deployed in B2FIND software stack. The upgrade to B2FIND 2.4 was done on a new production machine, due to the extended workflow a reingestion of all metadata records (again in close communication with Community representatives) took place.

- The major upgrade to the next service version B2FIND 2.4 includes:
  - enhanced Metadata Schema[10],
  - new graphical user interface with extended search facets for <Contributor> and <ResourceType>, displaying labels for persistent identifiers DOI and PID (if available) and a nicer look and feel display of the datasets,

---

[9] http://b2find.eudat.eu
[10] http://b2find.eudat.eu/guidelines/mapping.html

○ further implementation of Community needs, e.g. harvesting via OGC CWS[11], SparQL[12] and CKAN-API[13] in order to crawl directly from geonetwork, RDF or other CKAN repositories, respectively.

Progress has been made regarding a common Classification for Research Areas as a Collaboration Agreement was fixed for developing this closed vocabulary cooperatively with other partners involved in interdisciplinary research.

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).

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 on a daily basis 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[14] is a service for provisioning large reference open data sets, based on Onedata distributed virtual file system 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.

---

[11] https://geonetwork-opensource.org/manuals/2.10.4/eng/developer/xml_services/csw_services.html
[12] https://www.w3.org/TR/rdf-sparql-query/
[13] https://docs.ckan.org/en/2.8/api/
[14] https://datahub.egi.eu

3.1.2.1 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 providers 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

3.1.2.2 Possible 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 being integrated with B2STAGE via WebDAV protocol to enable data transfers between the EGI and EUDAT users.

3.1.3. INDIGO IAM

The Identity and Access Management Service is an integrated AAI solution which provides a layer where identities, enrolment, group membership and other attributes and authorization policies on distributed resources can be managed in a homogeneous way. For a detailed description of the service see D6.1, and the service documentation[15].

3.1.3.1 Service interfaces

IAM providers can integrate with services via standard OpenID Connect/OAuth interfaces and heterogeneous authentication mechanisms (SAML, OpenID Connect, X.509 certificates, plain username/passwords).

IAM provides a REST-ful provisioning and management API based on the SCIM[16] standard as well as the ability to integrate natively with Grid services via a VOMS[17] compatibility layer.

---

[15] https://indigo-iam.github.io/docs/v/current/
[16] http://www.simplecloud.info/
[17] https://italiangrid.github.io/voms

3.1.3.2 Existing and Possible Integration Partner Services

IAM has been integrated via SAML and OpenID Connect interfaces with the INDIGO PaaS services, ONEDATA (which is the technology underlying the EGI DataHub service), StoRM WebDAV and other services.

IAM has also been successfully integrated with the EGI-Checkin service in order to enable federated authentication leveraging both the OpenID Connect and SAML interfaces. This integration has been demonstrated using the IAM instance dedicated to the DODAS thematic service.

3.1.4. B2STAGE

B2STAGE is a suite of services aimed to transfer data into and out of EUDAT data nodes and exposes two protocols for staging data: GridFTP 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

3.1.4.1 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.

The HTTP APIs service is a set of RESTful endpoints that allow accessing to both B2SAFE data and metadata and it is aimed for small to medium datasets. This service offers programmatic access to data and thus allows for smooth integration of such data into other applications and data services.

3.1.4.2 Possible Integration Partner Services

B2STAGE is integrated with EOSC-hub Service B2SAFE by supporting all 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 can also be integrated with EOSC-hub Service B2SHARE by using B2ACCESS as common authentication layer. With this integration, users will be able to retrieve data from B2STAGE and share into B2SHARE.

3.1.5 B2DROP

B2DROP is a Sync & Share service offering researchers an easy way for collaborative working on documents and synchronisation of data across multiple devices. Besides 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

3.1.5.1 Service Interfaces

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

3.1.5.2 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[18] 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 must 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.

In figure 6 some sample harvested, and mapped records are shown as indexed and displayed in the B2FIND portal.

---

[18] Associated OAI-PMH harvest request
http://datahub.egi.eu/oai_pmh?verb=ListRecords&metadataPrefix=oai_dc
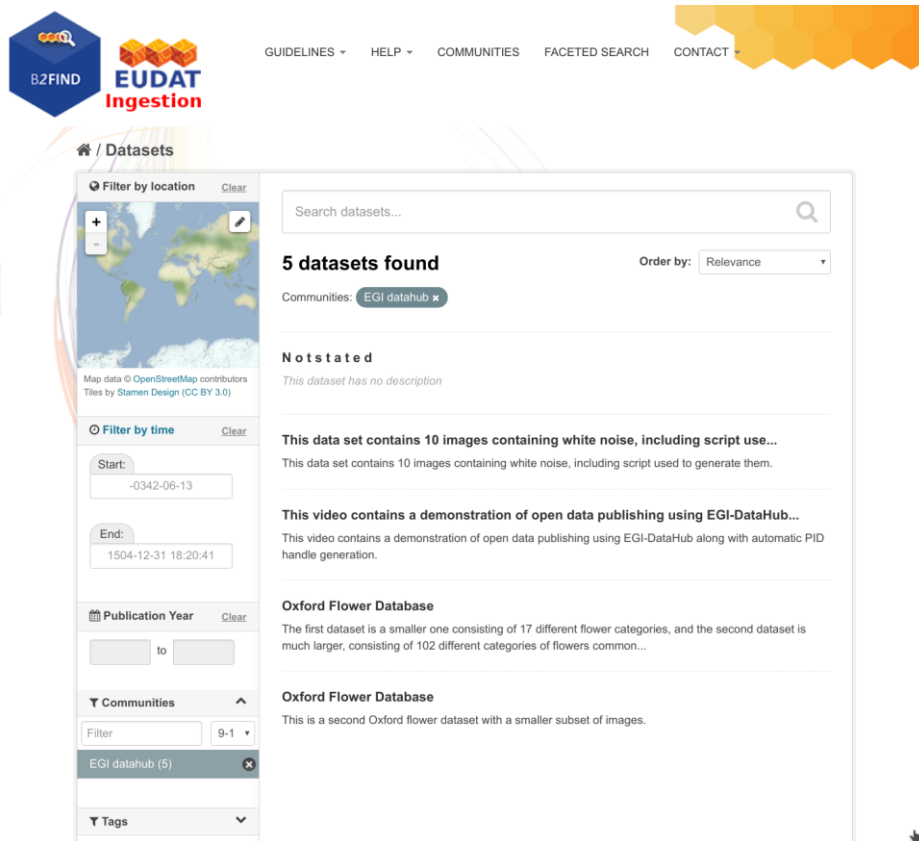
Fig. 6 - Test open data sets harvested from EGI DataHub by B2FIND
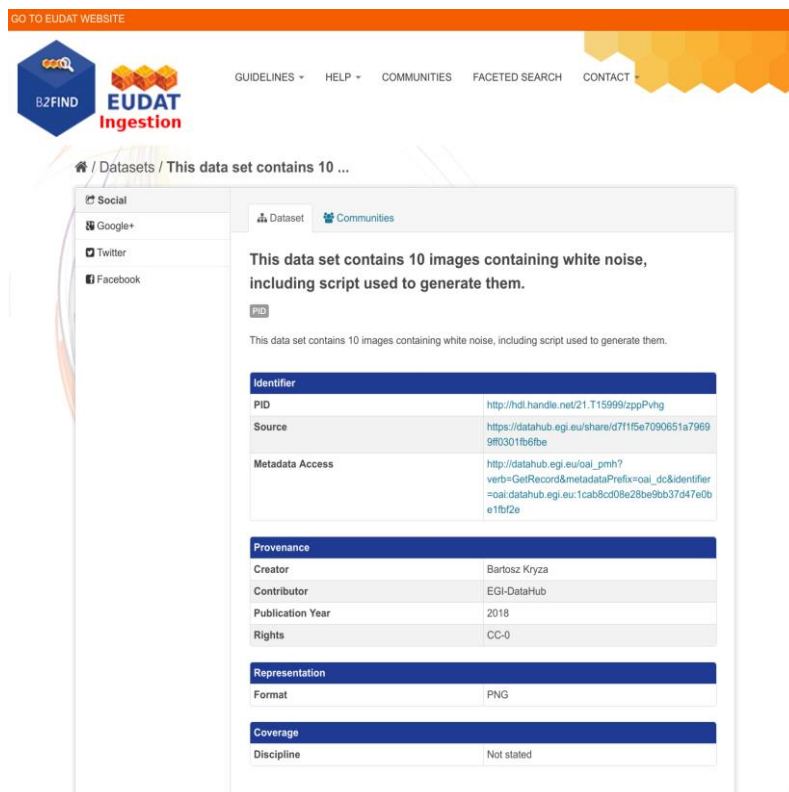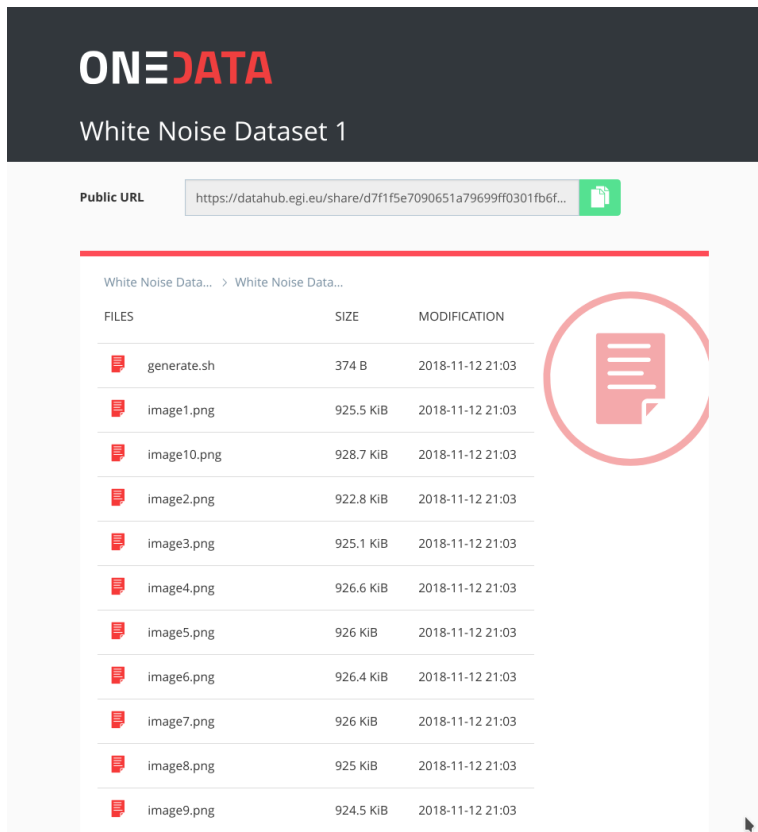
Fig. 7 - Metadata of a selected dataset harvested by B2FIND

As shown in figure 7, by opening the dataset display under *Identifier* to references to the underlying resource are provided in the fields *PID* and *Source*, which redirects to the file list as shown in figure 8.

Fig. 8 - Referenced data set can be accessed from web browser directly by following the handle

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

3.2.2 Staging data stored in EGI DataHub by B2STAGE for processing

During the current reporting period, the integration between B2STAGE and EGI DataHub has been investigated. This task was initially not feasible due to the lack of integration between B2STAGE and B2ACCESS. As a result, the activity was 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.

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

Now the integration between B2STAGE and B2ACCESS has been completed, as described in detail in a subsequent paragraph the task to integrate EGI DataHub and B2STAGE can proceed, as originally planned.

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

The integration activities between B2STAGE and B2ACCESS were undertaken to improve user access to B2STAGE. Via single sign on, and to enable integration with other services, for instance the EGI DataHub service as mentioned above.

The B2STAGE HTTP-APIs fully implemented the OAuth2 workflow required to support the B2ACCESS authentication (described in the Service Integration Documentation[19] from B2ACCESS) by exposing two different endpoints. The first (/auth/askauth) is intended to let B2STAGE manage the whole OAuth2 workflow and it requires user operateion through a web browser. The second one (/auth/b2safeproxy) lets instead to skip the authorization part on B2ACCESS 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 until now 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 Json Web Token (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.

HTTP APIs are connected to B2SAFE by means of the python-irodsclient[20] (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

---

[19] https://eudat.eu/services/userdoc/b2access-service-integration
[20] https://github.com/irods/python-irodsclient

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 Sharing processed data in B2SAFE via B2STAGE and B2SHARE

The integration between B2STAGE and B2SHARE has been investigated during this reporting period, coming to the conclusion that a common authentication layer is required to be able to achieve a simple to use implementation. B2ACCESS represents the obvious choice but, since B2STAGE was not integrated with B2ACCESS´at the time, the integration effort towards B2SHARE has been delayed and postponed to the next period. Having now completed the preparatory steps, as described in the previous paragraph, we are ready to proceed with integration tests and implement a driver to be able to share data from B2STAGE into B2SHARE.

3.2.6 Retrieve and store small data sets with B2DROP

Within the last reporting period the integration of B2SAFE with B2DROP was investigated. We investigated the integration of B2SAFE with B2DROP in two directions. First, the integration of B2DROP as a backend to B2SAFE (access B2DROP through B2SAFE) and secondly the integration of B2SAFE as a backend to B2DROP (access B2SAFE through B2DROP).

The integration of B2DROP as a backend to B2SAFE is not considered feasible for various reasons, for example file size limitations and the inability for users to mount B2DROP spaces on B2SAFE servers. The file size of data in B2SAFE might be bigger than the limitations of B2DROP. If users are not aware of the limitations and upload files which are too large, chunks of the file are consuming the storage, but the file is not usable in B2DROP or on synchronised devices. To integrate B2DROP as a backend to B2SAFE, the B2DROP space needs to be mounted on the B2SAFE server. Users do not have the ability to do this, either via command line or GUI.

To integrate B2SAFE as a backend to B2DROP, B2DROP needs to support external storage, which is possible. The user needs to enter the server URL and their credentials. The B2SAFE storage is mounted within a folder in the B2DROP space of the user. The files are not transferred to B2DROP but still stored in B2SAFE. The limitations of provided B2DROP storage and file size are still valid, but files in the external storage do not count against these limitations. Files will only be accounted if they are moved into the B2DROP storage. If the B2SAFE storage is mounted within the B2DROP account, users can easily move files between the storages. Accessing the B2SAFE storage the first time in a session will take some time to scan the remote B2SAFE storage and list the stored data.

Beside the technical possibility of the integration of B2SAFE as a backend to B2DROP, the operational workflow needs to be defined. This includes how B2SAFE deals with changes in the data caused by B2DROP and handle the gap of missing metadata. On the one hand, changes in data mean that files are added to or removed from the storage and on the other hand that a file has been modified. Because B2DROP is used for volatile data, which is still in processing, B2DROP does not store metadata, which describes the data, but this metadata should be present in B2SAFE.

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 persistent identifiers to the published data sets which can be then referenced.

3.2.8 EUDAT dataset discoverability

OpenAire and EUDAT-B2FIND enhanced the compliance of their guidelines for data providers[21]. 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, within the cooperation between EOSC and OpenAIRE Advance[22] , that B2FIND will provide access to an enriched metadata indexed to OpenAIRE. This will lead both to a further spreading and to an improved curation of metadata by the services of OpenAIRE. In order to implement this, B2FIND plans to offer its processed metadata in a format compatible with OpenAIRE via OAI-PMH.

In this context, we would also like to point out the cooperation with other initiatives such as EOSCpilot 6.2 'Data Interoperability'[23] and the RDA's Data Discovery Paradigms IG[24], which work as well on building interoperable standards and schemas for metadata management.

3.3 Future Integration Plans

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

3.4 Issues and Delay

Although there was ultimately no significant delay in achieving the objectives set, especially in the first months of the project, it took time for the cooperation between the teams of the three major e-Infrastructures to be coordinated and gain momentum.

---

[21] compare guidelines of OpenAire (https://guidelines.openaire.eu/en/latest/data/index.html) and EUDAT-B2FIND (http://b2find.eudat.eu/guidelines/introduction.html)
[22] https://docs.google.com/document/d/1zXcDrrS2Ud8XL2lDFJcj2b1CQjMzmHKp7USBBJkKvVc
[23] https://eoscpilot.eu/content/d66-2nd-report-data-interoperability
[24] https://www.rd-alliance.org/groups/data-discovery-paradigms-ig

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 services:

Table 2 - Federated compute services: available options

	Cloud Compute	Cloud Container Compute	High Throughput Compute
What is it?	Multi-cloud IaaS	Orchestration on top of EGI Cloud Compute (e.g. Kubernetes)	The grid, a scalable batch system
What you run?	VMs	Containers (e.g. Docker)	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

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[25] 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

The following interface are provided as service interfaces to the EGI Federated compute resources:

- GUI access: AppDB VMOps https://dashboard.appdb.egi.eu/vmops
- API/CLI access:
    - Discovery: AppDB is API (REST and GraphQL) accessible by
       https://wiki.egi.eu/wiki/Federated_Cloud_Discovery#AppDB
    - IaaS Federated Access Tools, see
      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

4.1.1.2 Possible Integration Partner Services

This section highlight possible partner services with could be considered for integration with the services in question, in this case the EGI Compute Cloud, to provide added value. Similar sections are provided per service description throughout the following text.

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 with easy provision of 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

Native Kubernetes API with OpenID Connect authentication is used, see at

---

[25] https://www.egi.eu/services/cloud-compute/

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 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 running 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 three 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 for both 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 in the command line and offers a Docker like command line interface.

4.1.6.2. Possible Integration Partner Services

uDocker is integrated with the EOSC-hub thematic service OPENCoastS

uDocker can be used with the EOSC-hub service EGI High-Throughput Compute

uDocker can be used with the EOSC-hub service EGI Workload Management

uDocker can be 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. Service Interfaces

CernVM-FS is implemented as a POSIX read-only file system in user space (a FUSE[26] module) which may be mounted in a virtual machine or container. Once mounted, end users or services can access CernVM-FS as if it were a local filesystem.

4.1.7.2. Possible Integration Partner Services

CVMFS can be integrated with EOSC-hub service Cloud Container

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 flavours available on the providers of the EGI Cloud Compute infrastructure was completed during the first year of EOSC-hub. The integration consists of the support of OpenID Connect by 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.

---

[26] https://github.com/libfuse/libfuse
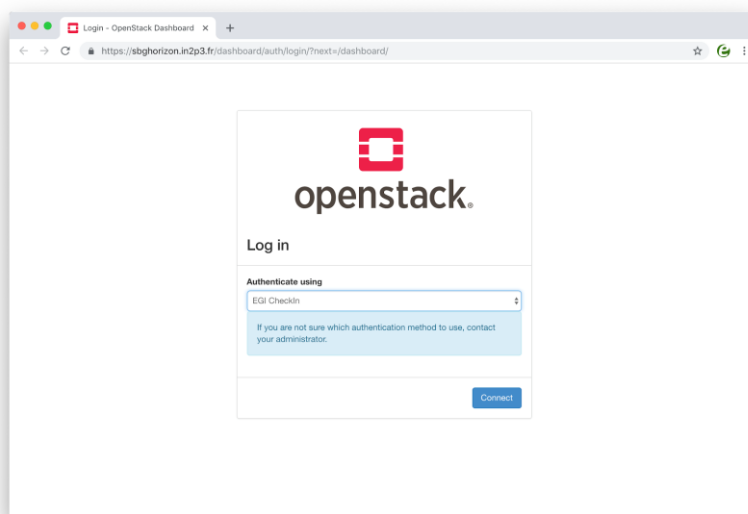
Fig. 9 - OpenStack Login dashboard with EGI Check-in

Detailed documentation on how to integrate and configure providers is available through the EGI Cloud integration documentation[27] 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[28] service

- Use of OpenID Connect with OpenStack clients is described in the service documentation

- OCCI clients were updated to support OpenID Connect and documentation was 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 to 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[29] 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

---

[27] https://egi-federated-cloud-integration.readthedocs.io/en/latest/
[28] https://argoeu.github.io/
[29] https://ggus.eu/

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 10).
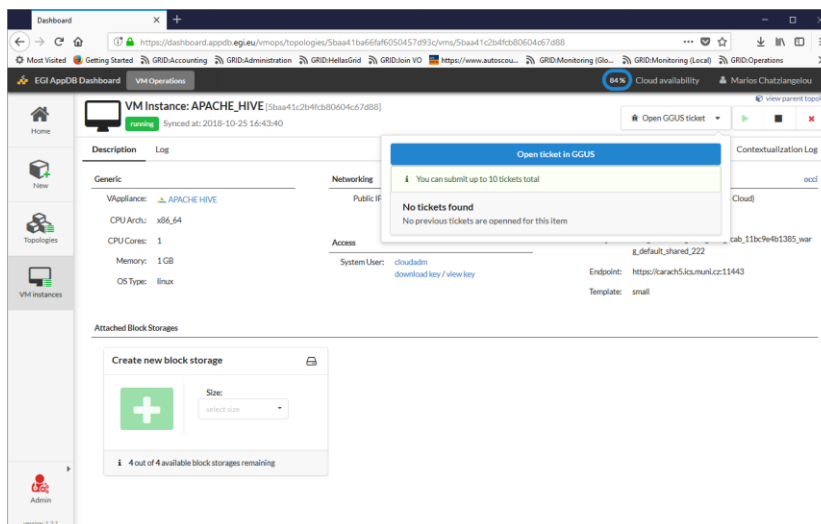
Fig. 10 - File a GGUS ticket directly to the site via the VMOps

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 VM 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.

Table 3: Checks performed by SECANT

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[30] 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 puppet forge 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 near 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 takes 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.

---

[30] http://www.ws-i.org/

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, see figure 11. 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 site-centric to VO-centric operation of 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 synchronize 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.
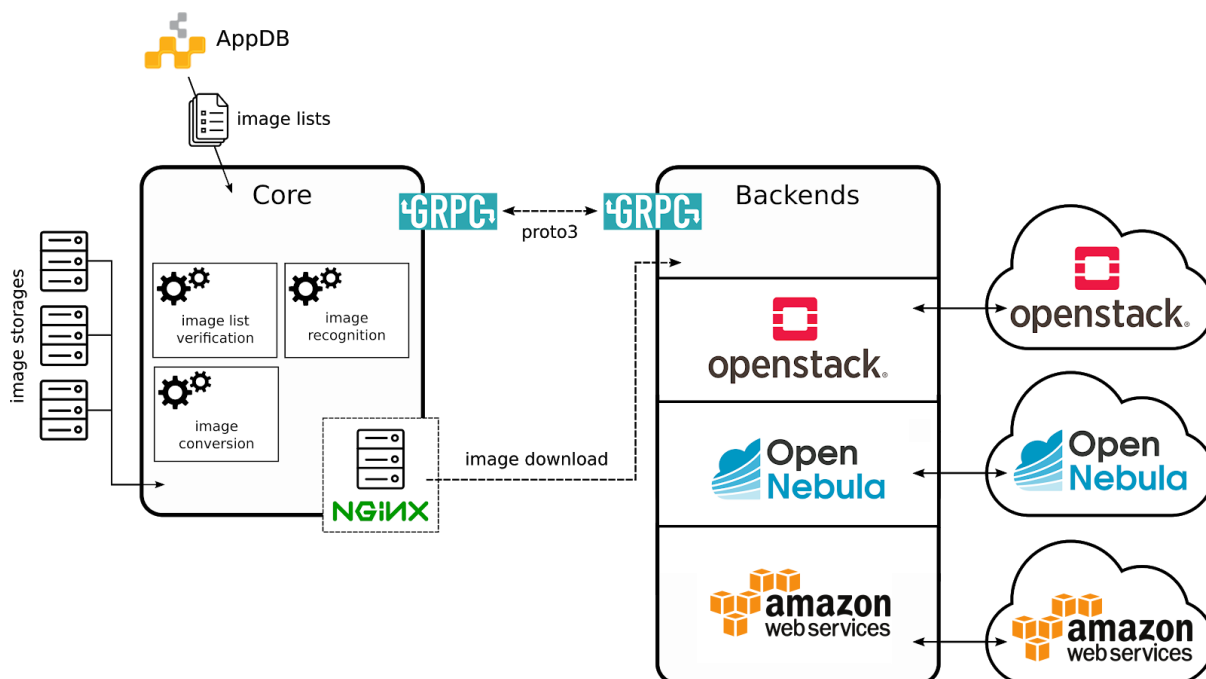
Fig. 11 - Cloudkeeper 2.0 ecosystem

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 Open Stack have been identified and implementation is ready to commence.

4.2.5. Elastic Kubernetes cluster support

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.

This task focused on adding 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

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).

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

- Enhancements to the interface with Docker container repositories such as Docker hub and others enabling more robust download of images and addressing interoperability and authentication limitations.
- Better and more resilient auto download and auto installation of uDocker and its engines at execution time, several enhancements to the command line interface especially important in the batch and background execution typical of the EGI services.
- Better and more universal support for NVidia GPGPUs supporting a wider range of heterogeneous host configurations and distributions.
- Experimental support for small containers based on Alpine in the Fn modes to enable more lightweight containers easier and faster to transfer.
- Finally, many improvements have been introduced in the execution engines among others improving their robustness and providing a better access to host directories from within the containers.

In collaboration with the thematic service, OPENCoastS the uDocker integration was validated with the EGI High-Throughput Compute, EGI Workload Management, and EGI Cloud Compute services.

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. The accounting of VMs with a long lifetime was incorrectly reported from the cloud providers and incorrectly managed at the APEL repository. Several releases of the accounting probes have progressively provided better reporting for these Virtual Machines and now all known issues are corrected.
- Provide accounting for public IP addresses. Public IP addresses are a scarce resource for some providers and subject to charges so WP6.2 has put effort on providing a solution for acknowledging the usage of this kind of resources in the infrastructure by capturing accounting information about them. The format of the record for storing information about the IPs was agreed between APEL team and experts of the cloud middleware platforms. Implementation of the probes is proceeding.
- Provide accounting for block storage devices. During this period, the WP6.2 team has agreed to use the StAR format (https://www.ogf.org/documents/GFD.201.pdf) for providing usage information about block storage devices allocated at the cloud providers. The implementation of the record is proceeding now in the probes.

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 supports 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 has been 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 to cloud-info-provider. Recently, the OpenStack site with GPGPU has also been 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. 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 the end users, the NVIDIA Docker Virtual Appliance Image was created and is provided through EGI Applications Database. NVIDIA Docker technology allows building and running Docker containers leveraging NVIDIA GPUs. The VA Image allows users to run various pre-built GPU-ready containers (e.g. deep learning library Tensor Flow) inside virtual machines with attached NVIDIA GPU accelerators.

4.2.9. Improved support for native API's of cloud stacks

The Cloud Compute service no longer mandates a single API for interacting with the IaaS cloud provider. The OCCI standard API used up to now was meant to homogenise the interface of every provider, but on the one hand, users still had to care about provider-level details, hence the portability of applications was compromised; and on the other hand, the support of the standard outside the EGI ecosystem was poor and no mainstream cloud-tooling was available, hence the effort for porting applications to the service was high even for those users that were already consuming cloud services on other platforms. Users now are encouraged to directly use the native APIs of the cloud system so they can take full advantage of the features available at the federated providers and also leverage any existing native tooling easily. The support for OCCI API will be maintained for one year to give users the possibility to migrate to the native APIs.

The support for these native APIs requires:

- Being able to retrieve information about the endpoints, capabilities and access modes of the providers in a programmatic way.
- Support the monitoring of the providers using their native APIs instead of OCCI.
- Adapt existing integration components of the federation to use only native APIs and avoiding any modification to the regular operations of a provider.

The WP6.2 team has contributed to this task with:

- Development of the GLUE 2.1 Schema standard for fully accommodating the native API information so clients can discover for each provider the best way to interact with them.

- Complete implementation of the GLUE 2.1 Schema in the cloud-info-provider tool so the information can be automatically generated by querying the native APIs of OpenStack and OpenNebula.
- Transition to a message-based system provided by the ARGO Messaging System (AMS) instead of BDII to deliver the provider's information to the clients. This transition is currently undergoing and will avoid delays in the propagation of the information and better control on how the information is delivered to the clients.
- Adapt the AppDB Information System to consume the new GLUE 2.1 Schema via messages sent via AMS.
- Improvement and development of monitoring probes using native APIs so providers joining the federation without OCCI support can still be monitored for their Availability and Reliability.
- Test supported orchestrators with native APIs (mainly IM and Terraform) and document their usage. Support users to the transition from OCCI to native API as needed.
- Document the exact APIs used and the expected access rights for the integration components for new providers.

The architecture of the EGI Cloud evolved accordingly to accept any cloud platform to be federated into the service. The figure below shows the main components of the architecture
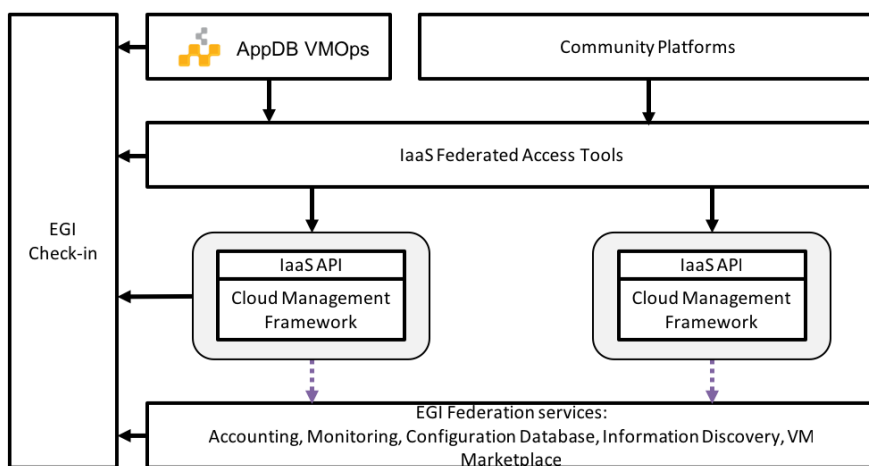
Fig. 12 - Architecture of the EGI Cloud

A proposal to further expand this architecture[31] was also presented and after collecting feedback, the work will focus this year on centralising the operations of the integration tools to make it easier to bring new 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 allows access to a diverse set of compute resources, including HPC resources, and opens the door for opportunistic resource usage. To enable access to these diverse resources it is clear that higher level frameworks are needed and that close integration

---

[31] https://docs.google.com/document/d/18spjoA1_Pe6NOVbCxPUc9Dng1OjY8sk0sxhADGMA-0M/edit

with AAI services, which may differ across resources, is required. 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:

- While the transition to a native-API cloud service of EGI is still in progress, DIRAC improved its internal implementation of the DIRAC abstract interface to cloud mangers using the REST OCCI interface. This implementation has a much better performance compared to the previous based on the rOCCI command line tool to access OCCI cloud managers.

- This 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 to allocate cloud resources with a proper authentication whenever user payloads are available in the system.

- At the same time, the Workload Manager has started to implement native cloud connectors that use directly the OpenStack and OpenNebula REST interfaces without the OCCI middle layer. This makes the system more efficient and prepares for the eventual stopping of the OCCI support. Focus was put on the OpenStack support, which is used in the majority of the EGI cloud sites, with authentication with the X509 VOMS certificates.

Another important set 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.
- The work is also ongoing to provide the EGI Check-In authentication for the 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 be potentially integrated with any other service as the VMs run on the cloud can execute any arbitrary software. Access to any service requires that the right 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 lifetime.

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 to persist the storage even if the VMs fail and to augment 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:

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 13 shows the Notebooks environment with a terminal accessing the DataHub files:
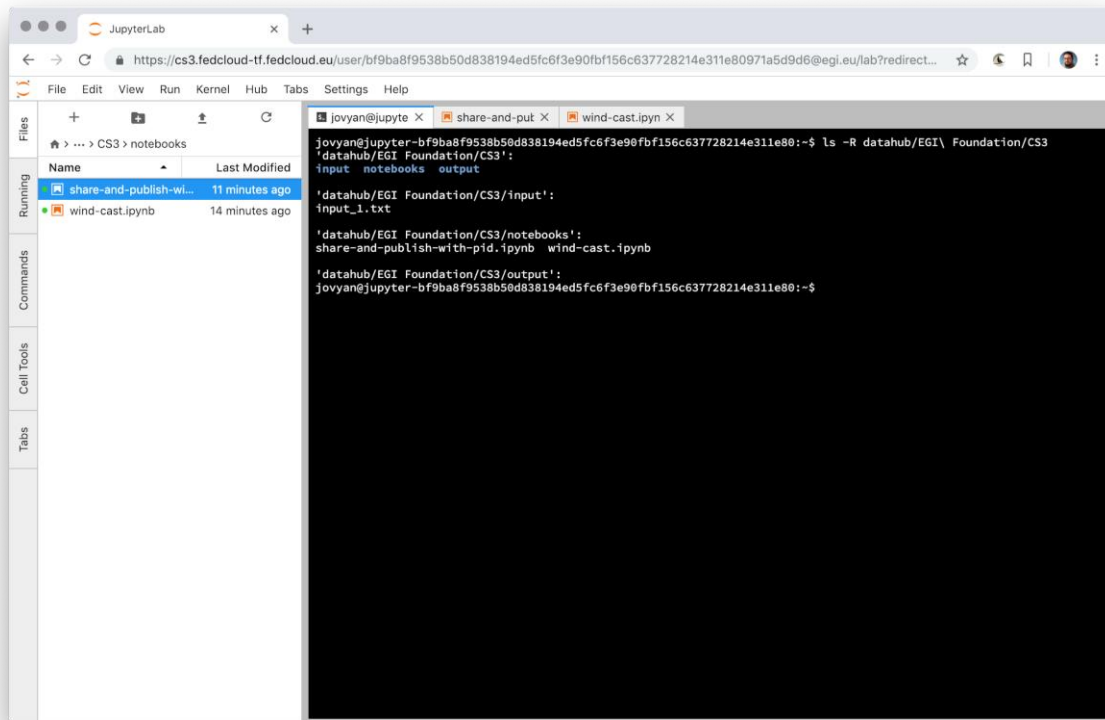
Fig. 13 - notebooks environment with a terminal accessing the DataHub files

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 of figure 14:
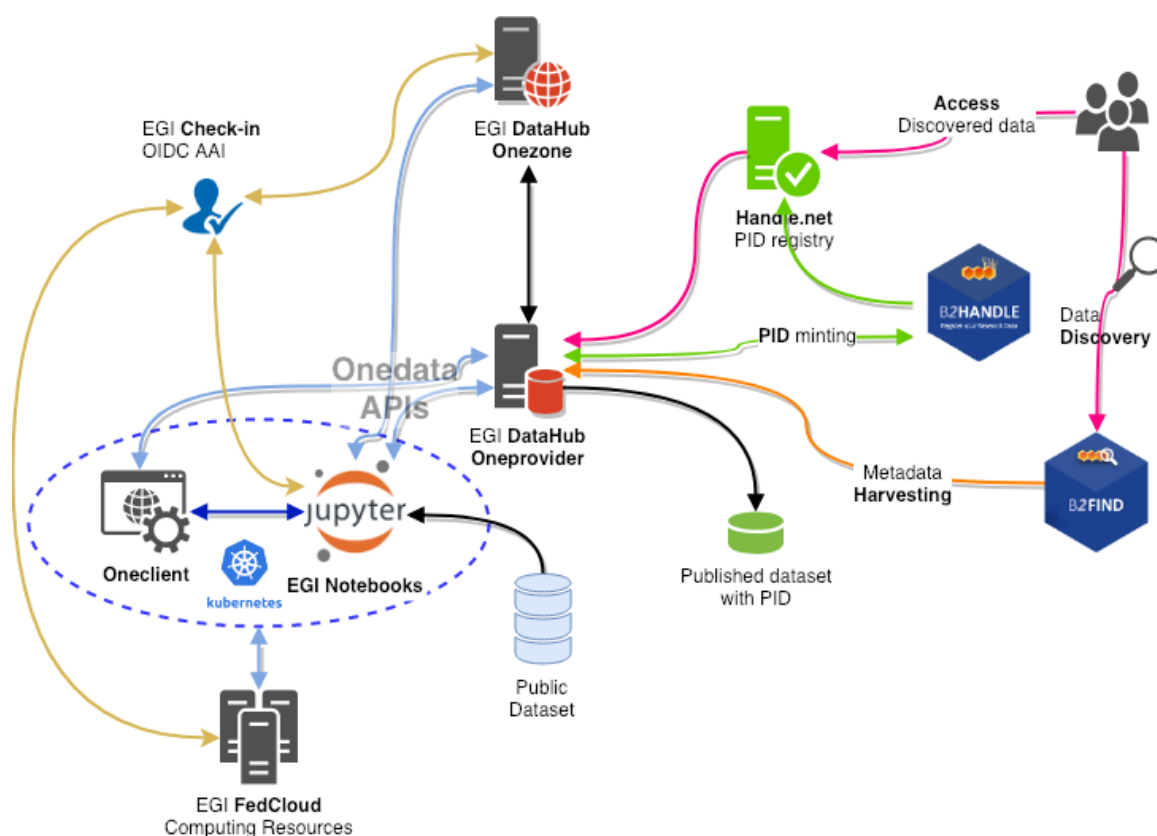
Fig. 14 – Workflow with DataHub, B2HANDLE and B2FIND

Access to B2DROP:

B2DROP offers a WebDAV interface that can be mounted from the EGI Cloud VM. However, as the authentication is handled by the SAML protocol, the integration cannot be performed automatically as with the DataHub and requires the user to provide an app password, which can be used in the WebDAV protocol to access her space 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.

4.2.12 uDocker in Sensitive data service

The EOSC-hub project is providing services for sensitive data [32]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

4.2.13. 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[33], which is a flexible data anonymization tool that transforms relational and transactional databases to dataset where formal privacy guaranties hold. It allows to remove identifying information from sensitive data.

4.3. Future Integration Plans

- Maintenance releases of the services
- Finalise integration of Check-in across the EGI Cloud providers
- Finalise implementation of GLUE 2.1 Schema and usage of the AMS for sending information, integration of clients (AppDB)
- Improve integration of EGI Workload Manager with EGI Cloud and EGI Check-in
- Enhance EGI Cloud Container Compute service with the elastic Kubernetes developments
- Improve accounting probes, focusing on storage.

- New endorser dashboard for AppDB

4.4. Issues and Delay

There was no significant delay and no critical issues in this task area within the first project year.

---

[32] https://eosc-hub.eu/catalogue/Services%20for%20sensitive%20data
[33] https://amnesia.openaire.eu/

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[34] and integrate the INDIGO-DataCloud PaaS components as orchestrator for the EOSC-hub services.

Figure 15 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.

---

[34] 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.
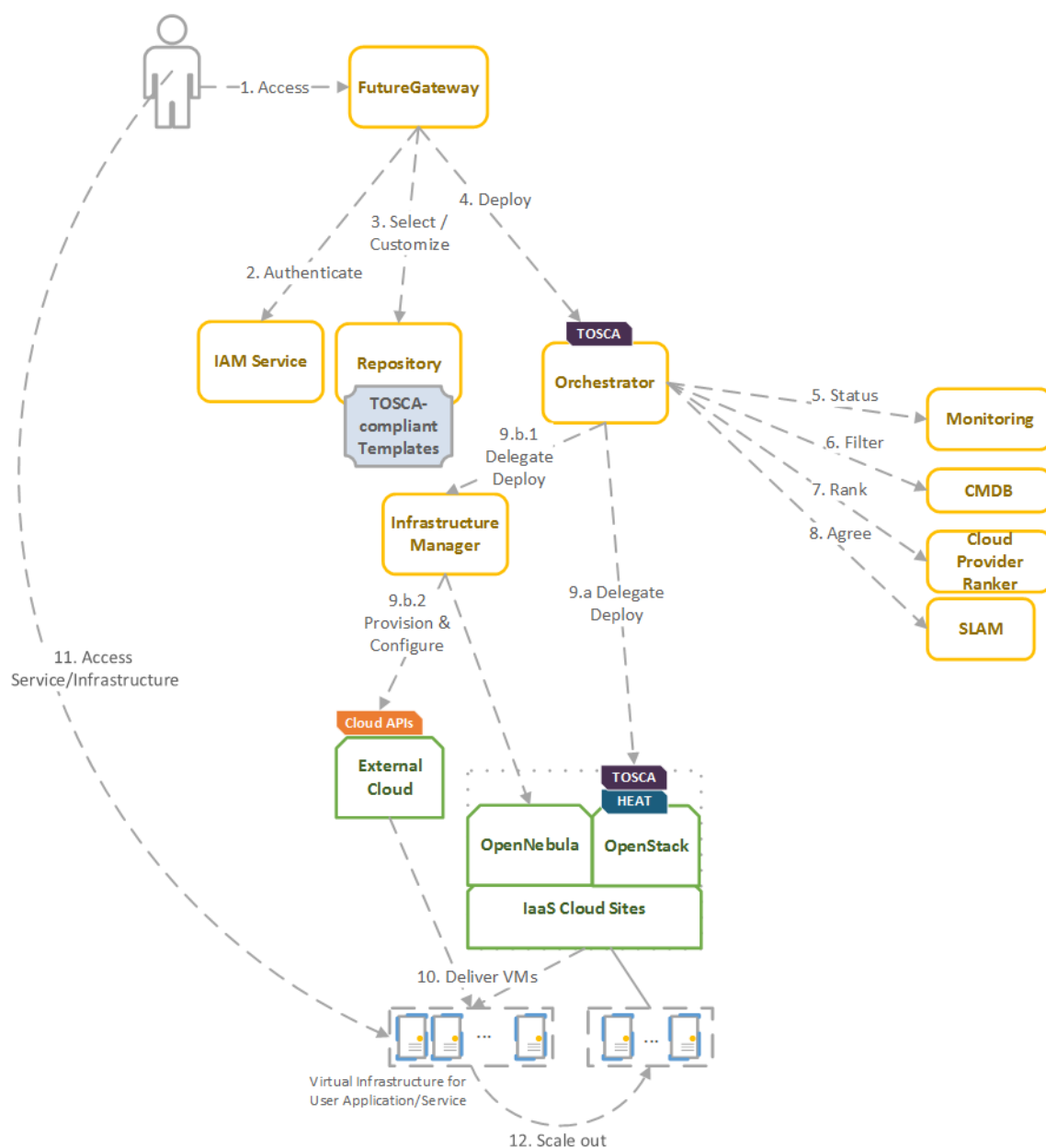
Fig. 15 - Architecture and workflow among the Processing and Orchestration services in EOSC-hub

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, it is in charge of interacting 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 *Cloud Provider Ranker* 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

Command Line Interface (CLI).

5.1.1.2 Possible Integration Partner Services

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[35], a XML-RPC API[36], a REST API [37]and a command-line application[38].

5.1.2.2 Possible Integration Partner Services

IM is integrated with EOSC-hub Service: PaaS Orchestrator, EC3, EGI VMOps Dashboard and EGI FedCloud.

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 Future Gateway

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

---

[35] https://github.com/grycap/im-web
[36] https://imdocs.readthedocs.io/en/latest/xmlrpc.html
[37] https://imdocs.readthedocs.io/en/latest/REST.html
[38] https://github.com/grycap/im-client

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 needs.

FutureGateway foresees the following software components:

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

Potentially the FutureGateway framework can be adopted by any community requesting capabilities offered by a Science Gateway. The APIs can be also used for Mobile applications and Workflow engines.

fgAPIServer can be integrated with EGI AAI and INDIGO IAM.

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

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 performance and enhanced reliability and for supporting the new requirements coming from the thematic services.

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[39] 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:

---

[39] https://www.flowable.org/

- 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: https://github.com/indigo-dc/tosca-templates .

5.2.2   Infrastructure Manager evolution

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 support for EC2 spot instances has been added. EC2 spot instances were already supported in the AWS IM connector but TOSCA support has been added. This required adding the "pre-emptible 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 definitely eases the Kubernetes cluster as a Service functionality required in the ELIXIR CC.

The deployment includes an 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 a HAProxy to balance the load among them. In addition, the IM web portal has been deployed in this cluster using a nginx service to separate the application deployed. 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/

5.2.3 FutureGateway extensions for EOSC-hub

FutureGateway has been actively developed in 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 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 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. 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 re-usable 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 EOSC-hub project, the module has been updated to be up to date with upstream changes. Additionally, the TOSCA template to deploy Docker image with Kepler and run a workflow has been published.

5.2.4 TOSCA HEAT-Translator evolution

The evolution roadmap for TOSCA heat-translator tool is primarily focused on maintenance activities. In particular, the active support on the OpenStack-related TOSCA templates from 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[40]. Currently, this pipeline triggers a translation check for each change in the heat-translator code. Plans include extending the aforementioned checks whenever a change is done 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[41] tosca.nodes.SoftwareComponent and an upstream contribution to the OpenStack code repository that prevented log messages from being lost [42]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 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 EOSC-hub project environment, there are some overlaps between CMDB and AppDB that is already used in EOSC-hub – playing a role that is like CMDB for the community built around EGI infrastructure. Although, AppDB seems a mature, more widely adopted tool with a potential to be a replacement for CMDB, we investigated that 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 non-ideal solution.

With interface unification in mind, CMDB was redesigned using the source code of AppDB publisher. In this way, CMDB was 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. Moreover, AppDB is built as a distributed system allowing for the synchronization of changes between database nodes. This architecture is still maintained in the provided solution. The main instance of EGI AppDB database node, will be the source of information for CMDB EOSC-hub. In this way, CMDB data will be always synchronized with the up-to-date master database. In order to fulfil requirements coming from INDIGO PaaS environment, information schema, database contents and programmatic interface of AppDB needed to be extended. This development also needs to be reflected in other components due to the API changes between old version of CMDB and the new one based on AppDB.

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 fulfil these requirements. Once Service Level Agreement is met, other

---

[40] https://jenkins.indigo-datacloud.eu:8080/job/Pipeline-as-code/job/heat-translator/job/devel/

[41] https://github.com/indigo-dc/heat-translator/commit/6f637df9762c11ee50261734510a7a22cb72caa8

[42] https://github.com/openstack/heat-translator/commit/ef32ac699195ce179865b6070164b67cdf0c1485

components (such as Orchestrator) can 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 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 GraphQL protocol (compliant to AppDB API).

The current CMDB data schema as well as the new content provided by its API can be used by SLAM to reflect new SLA targets (such as network properties). In the next development cycle an analysis of the potential new targets will be conducted, in order to select properties relevant for the infrastructure SLA negation.

5.3   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;
- Integrate orchestration layer with EGI Cloud Compute, enabling users to use orchestration tools with resources allocated by EGI Federated Cloud infrastructure.
- Enhance custom TOSCA types (and if required TOSCA Heat-Translator) to add new types requested by user communities.
- Enhance data management in Orchestrator.

5.4   Issues and Delay

There was no significant delay and no critical issues in this task area within the first project year.

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 to a persistent identifier, are described in detail in the following paragraphs and shown in the picture (Figure 16).

Those services allow to store a data set in a repository, which is geographically distributed, and associated a persistent identifier to 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 to 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. Last, but not least, data are curated through a set of policies that each data manager can define.
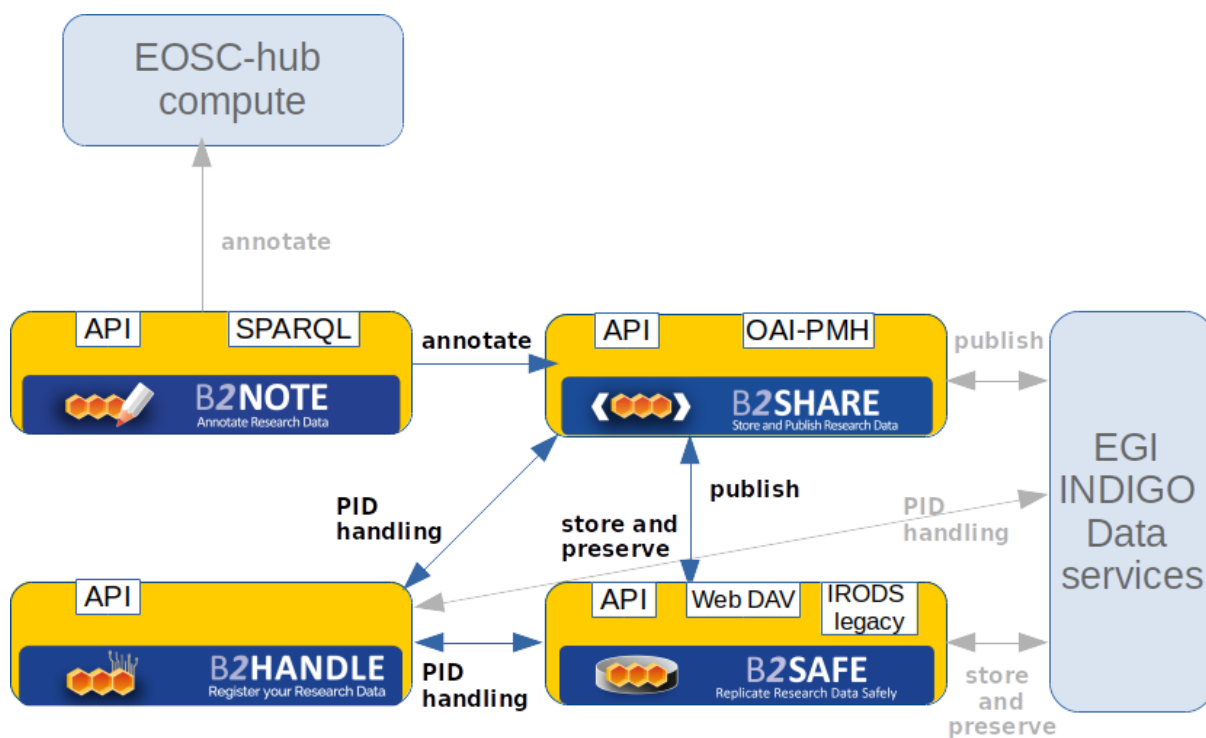
Fig. 16 - Data and metadata management services

The Thematic Services, according to D7.2, have expressed interest to integrate data services in the second year, therefore there are no integration activities with them in progress yet. One competence center, T8.2 Fusion, has requested data resources to set up a pilot: it should allow the replication of data on geographically distributed sites (STFC, CINECA, POZNAN) and the enforcement of customized policies via the B2SAFE service. The data and metadata management team are supporting two use cases coming from the previous project EUDAT2020. One is ICEDIG[43], which is the follow-up of the project Herbadrop and whose main service provider is CINES. ICEDIG needs to

---

[43] https://www.icedig.eu

make their metadata visible through B2FIND and identifiable through the B2HANDLE persistent identifiers, associated to the data by means of B2SAFE. The other is CompBioMed, which has been analyzed to define a policy with the Data Policy Manager. Both use cases are described in more detail in section 2.

6.1   Maintenance, interfaces and integration options of the services

The following subsection describe the maintenance, interfaces and integration potential of each specific service. It should, however, be noted that in addition to maintenance activities monitoring plugins have been developed for each B2 service and service monitoring is provided by the EUDAT Argo monitoring service.

6.1.1   B2HANDLE

B2HANDLE is a service to assign persistent identifiers (PIDs) to digital objects, making them referenceable across middleware services. B2HANDLE relies on the Handle System to achieve this. For a detailed description of the service, see D6.1 and https://marketplace.eosc-portal.eu/services/b2handle.

6.1.1.1 Maintenance

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).

6.1.1.2 Service Interfaces

B2HANDLE offers multiple service interfaces, including the native (Java) Handle API and the native Handle REST API for PID management (create, read, update, delete), the B2HANDLE search service at each service provider and at the B2HANDLE Central PID Catalogue. To facilitate easy programmatic access to these interfaces, including search, B2HANDLE provides the b2handle/pyhandle Python libraries. The pyhandle library also supports interaction with the B2HANDLE raw data database (for administrative purposes only) and supports the generic Handle Batch format to accommodate asynchronous bulk operations.Possible Integration

6.1.1.3 Possible integration Partner Services

B2HANDLE is already integrated with B2SAFE and B2SHARE. Additionally, the B2HANDLE team is now working on improved integration with B2SHARE, supporting PID metadata profiles, and initial integration with DataHub, Online Storage and the Federated Data Manager. B2HANDLE supports integration with EOSC-conformant AAI services where possible and useful.

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[44]. Moreover, a WebDAV interface is available as a separated component[45]
- 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 uploading and downloading 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[46].

6.1.3.1 Service Interfaces

B2SHARE is accessible through a web interface and a REST API that allow a user to create, modify and manage records. The service is integrated with B2DROP to allow direct uploads, B2HANDLE to

---

[44] https://irods.org
[45] https://github.com/UtrechtUniversity/davrods
[46] https://eudat.eu/services/userdoc/b2share

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.

Design documents have been written to 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.

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 be then used to organize and retrieve datasets based on the user's needs. For detailed description of the service, see D6.1.

6.1.4.1 Service Interfaces

B2NOTE offers two different Interfaces:

- a User Interface available as a widget for the integration within the User Interface of partner services and
- 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.

The RESTful API is currently limited to a small subset of functionalities but should be extended depending on the various integration requirements.

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 community services such as the CLARIN Virtual Language Observatory.

B2NOTE can be integrated with OpenAire data services such as Zenodo and the Research Community Dashboard.

6.2 Integration activities

6.2.1 Adaptation of B2HANDLE service to FitSM

B2HANDLE has started adaptation of its operational processes to be in line with FitSM procedures and best practices.

- Changes to the operational setup of B2HANDLE are managed in coordination with EOSC-hub and EUDAT **Change Management**.
- Releases of new B2HANDLE software components will be coordinated with EOSC-hub **Release and Deployment Management**.
- The specification of configuration items of B2HANDLE is maintained as part of EUDAT **Configuration Management** using the EUDAT DPMT.
- A first description of B2HANDLE has been included in the EOSC-hub service portfolio/marketplace (**Service Portfolio Management**).
- The B2HANDLE operations team has also started to remodel internal workflows and operational status pages to support **Customer Relationship Management**.
- B2HANDLE service monitoring is currently being updated by the monitoring team to accommodate EOSC-hub requirements.
- B2HANDLE has participated in a EUDAT-led workshop to observe and plan for future improvements, which can be fed into **Continual Service Improvement**.

6.2.2 Possibilities for integration of B2HANDLE with other EOSC-hub services

Concerning integration with additional EOSC components, B2HANDLE has taken the following steps:

- Integration with B2SHARE involves use of PID profiles and reverse-lookups, going beyond existing integration (use of identifiers only). Use of PID profiles by B2SHARE was discussed and it was agreed that an initial profile would be developed by B2HANDLE to be reviewed together with B2SHARE against a few typical use cases. The reverse-lookup service and extended filtering capabilities will be discussed as a follow-on action once profiles are established.
- Integration with the EGI DataHub was discussed with the DataHub team, particularly concerning the integration at the technical level. The available interfaces of B2HANDLE were explained and it was agreed that the best way forward is for the DataHub team to start integration based on a test identifier namespace (test prefix), which can be provided by B2HANDLE, followed by integration action by DataHub against the native CRUD interface of B2HANDLE. Activities on this are ongoing.
- Integration with EGI Online Storage was discussed with EGI. Further investigation on concrete steps to take is ongoing, pending further clarification of the scope of the EGI online Storage service.
- Integration with the EGI Federated Data Manager was discussed with EGI. The discussion indicated that the Federated Data Manager is based on OneData, which is the same solution powering the DataHub. Integration with Federated Data Manager is thus likely achieved as part of DataHub integration activities.

6.2.3 Integration improvements between B2ACCESS and B2SAFE

During the first year the integration between B2SAFE and B2ACCESS was completed. The solution which existed prior to the EOSC-hub project was rendered unusable due to a change in the B2ACCESS policies. Therefore, we re-implemented it adopting a different approach based on the PAM (Pluggable Authentication Module): the OpenID token, obtained by the user from B2ACCESS, is passed to iRODS as the password parameter during the authentication process. This 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[47].

6.2.4 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 the 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.

6.2.4.1 For EGI services

An investigation into the interoperation of the DPM with Onedata[48] 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.

6.2.4.2 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.

---

[47] https://github.com/EUDAT-B2SAFE/pam-oauth2
[48] https://onedata.org

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:

- **Step 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.
- **Step 2: Data pre-processing:** In BSC researchers pre-process the data which includes manual and automated steps for image stitching, segmentation and meshing
- **Step 3: Data Replication:** The pre-processed 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).

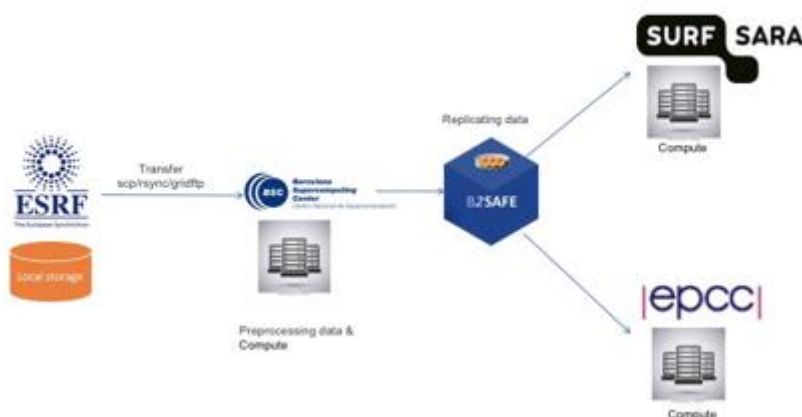Graphic below shows the data workflow, services and centers that are involved:

Fig. 17 - data workflow, services and centers involved in the data pipeline

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.

Table 4 – Activities to implement the solution for the CompBioMed use case

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[49] - Enforce the policy in the source compute center (detailed instructions for creating and enforcing a policy[50])	In progress
Replicate data	- The source data center (BSC) initiate the replication to replicate data	Not started

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

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

6.2.6 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 microservices are being developed. The new iRODS microservice 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 microservices.

6.2.7 B2SAFE data discovery and access

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

---

[49] https://dpmgr.eudat.eu
[50] https://docs.google.com/document/d/1K8-tF_eSiI0itCKUsbxYkuS6MVT1eGnXs0nnaWFRXnk/edit#

The development of the connection component and improvement possibilities were coordinated with 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 case. All unit tests are combined in a python test suit. The development of the unit tests led 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 production B2SAFE instance was created with files having actual PIDs, which is essential, as B2SHARE is validating 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. The HTTP API itself was extended by the B2SHARE team to be able to process the B2SAFE call for a draft creation with a list of PIDs.

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

6.2.8 B2SHARE extensions for diverse data organizations

Several communities have been supported and effort has been put in 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

6.2.9 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[51] 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[52] that covers the required changes and effort. A technical integration workflow has

---

[51] https://docs.google.com/document/d/11Gkl_7FCZF1U40sPeaEIHu4U-TnbOw3akGVKPEyWvD4
[52] https://docs.google.com/document/d/1uO4OJDe9SZyG2gBY_BjBAKIni5uARvlQePJVcEeRyFk

been designed[53] 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.

6.2.10 Improve two-ways integration of B2NOTE with B2SHARE

The integration work for B2NOTE started at M7. The initial effort has been focused on discussing and evaluating the different possible integration scenario 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.

We worked directly with 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 should be proposed in the first months of the next reporting period.

6.2.11 B2NOTE Integration with other EOSC-hub services

We started to evaluate the integration of B2NOTE with B2FIND through informal discussions with the B2FIND developer team. We identified two levels of integration: the User Interface integration to annotate B2FIND indexed datasets and the integration of annotation in the B2FIND search engine. Based on this common understanding, we will define an integration roadmap in the upcoming months.

6.2.12 B2NOTE Integration with OpenAire Research community dashboard

We started to evaluate internally initial requirements for the integration of B2NOTE with Zenodo and the Research Community Dashboard. As the resources were limited, the interaction with OpenAIRE teams have been limited during this reporting period.

---

[53] https://docs.google.com/document/d/1uO4OJDe9SZyG2gBY_BjBAKIni5uARvlQePJVcEeRyFk

6.3 Future Integration Plans

B2SHARE:

- Normal software and service maintenance activities; update B2SHARE and it's 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
- Finalize the integration between B2SHARE and B2SAFE.

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.

B2HANDLE

- Implementing the integration with B2SHARE.
- Finalizing the integration with DataHub.

B2NOTE

- Implementing the integration with B2SHARE.
- Implementing the integration with B2FIND.
- Implementing the integration with OpenAIRE dashboard.

6.4 Issues and delays

The activities related to B2NOTE have been delayed due to an initial lack of manpower of the involved partners. The delay will affect the activities of the second year too, but a new member of the team should be available starting from March 2019 (M15), therefore the delay is expected to be progressively reduced.

7 Summary and Outlook

In this report, we have presented the first integration and maintenance report of the EOSC-hub common services catalog. The work plan of the Common Services was defined in M6.1 (Rolling Maintenance Plan[54]) and M6.2 (Rolling Integration Plan[55]), which gives a comprehensive overview over the status and progress of maintenance and integration activities of WP6 and keeps track of the changes in the schedule. But these plans are not static, and their changes and updates are reflected in this report describing the work really done within the first project year.

In the first project year, we achieved the establishment of several service integrations and the application to several use cases. This successfully realized the EOSC Hub approach of enabling and supporting the use cases based on the 'Thematic Services' of WP7 through integrated 'Common Services', which are maintained and further developed in WP6. On the other hand, a few integration activities have been delayed due technical difficulties and human resource limitations. In some cases the integration cannot be achieved due to fundamental technical issues which would require significant reengineering of one or more components, which would put the core services at risk.

For the future, we can build on the work done and identify some next steps. The interoperability and compatibility of the services coming from three different e-Infrastructures, namely EGI, EUDAT and INDIGO, are not complete and will be the focus of the next year of the project. In addition, we will extend integration or common services to provide solutions for further thematic services. Two key challenges for the next project year in this context are the transparent integration of AAI services and high-performance data transfer between storage services.

The following integration topics are those included in the plans for the next year:

- The continued evolution/integration of the common services to suit end user needs as outlined in the work package description of work. More emphasis is to be placed on understanding the driving use-cases and cross work package communication.

- Promote and improve the integration activities which have already been undertaken (see fig. 18), to demonstrate them by applying them to real use cases and to generalize and adapt them to other thematic services.

- Address the AAI issues in collaboration with WP5 where we will seek to achieve a generic solution that will offer seamless access to services across infrastructures.

- Foster the cooperation with other initiatives such as OpenAire and GEANT to improve interoperable, open and federated data management.

More detailed integration and interoperability efforts will be reported in the consecutive deliverables from WP6, in particular in D6.4 Second report on the maintenance and integration of common services.

---

[54] https://docs.google.com/spreadsheets/d/1p5-wwDftbzgnbR8O2VTu_G2DuG4kzTQLMyX4LvJLFgA/
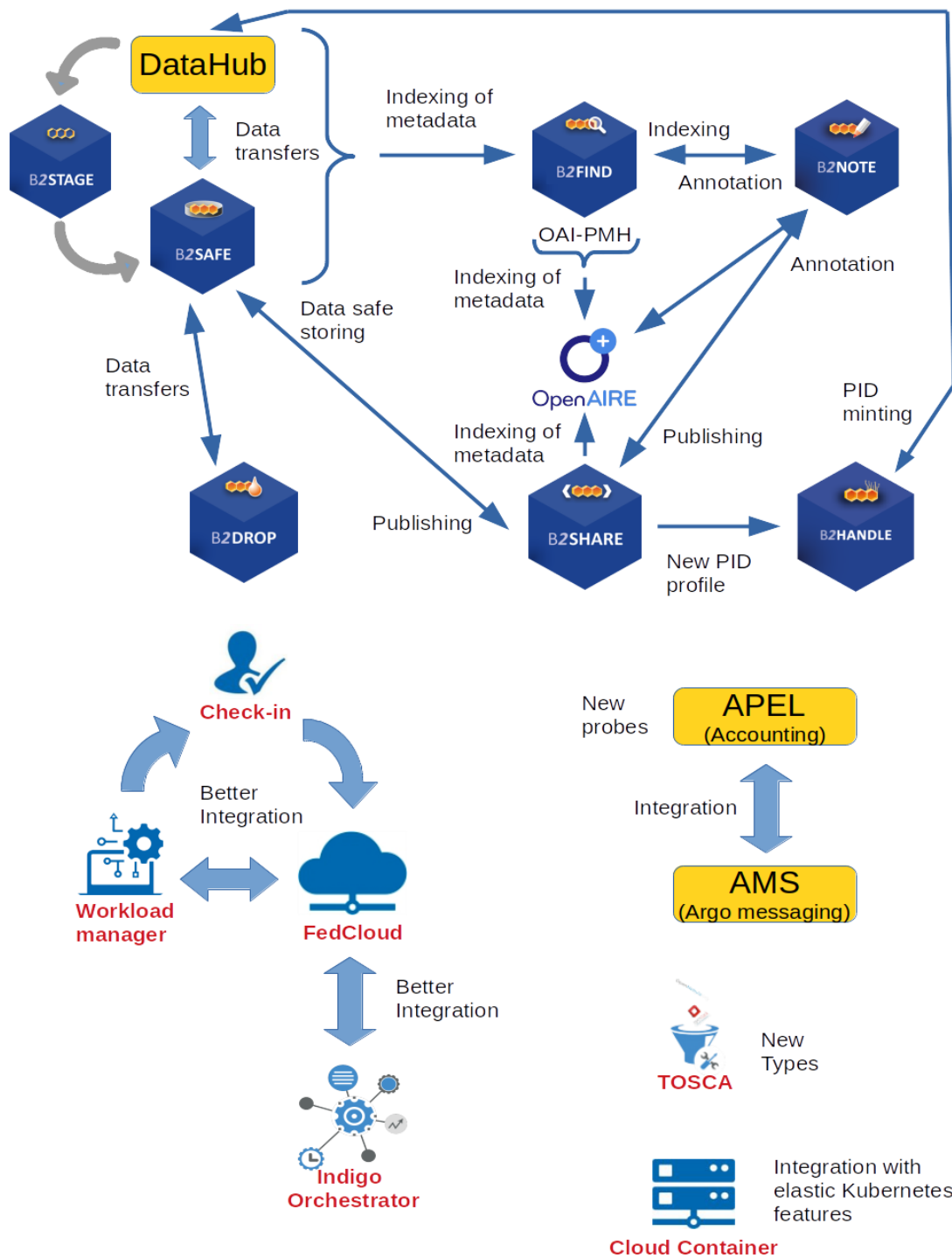[55] https://docs.google.com/spreadsheets/d/1D3kjhHtEfVLA2L1jsBS1O1NVPE2yiwmKD6KjBCNSlaM/

Fig. 18 – Summary of the planned service integrations