D2.7 Technical specifications for compute common services

Lead partner:	EGI Foundation
Version:	1
Status:	Under E Review
Dissemination Level:	PUBLIC
Keywords:	Technical Architecture, Compute Services
Document Link:	https://documents.egi.eu/document/3802

Deliverable Abstract

EGI-ACE builds on the computing e-Infrastructure of the EGI Federation to deliver the EOSC Compute Platform: an open, data-centric, distributed, hybrid and secure infrastructure consisting of computing and storage providers and platform services to support research and open science via data spaces.

The technical architecture of EGI-ACE provides a platform to build and deploy Thematic Services that can exploit baseline compute and storage resources from Cloud, HTC and HPC providers via a set of Federated Compute and Data services that facilitate running the workloads near the data to be analysed. Alongside with user-oriented services, the Core EGI Services cover the operational aspects of the federation (authentication and authorization, accounting, monitoring, helpdesk) and bridge the EGI-ACE services with EOSC Core functionalities.

This document presents the technical specifications of the EOSC Compute Platform services and components.

COPYRIGHT NOTICE

This work by parties of the EGI-ACE consortium is licensed under a Creative Commons Attribution 4.0 International License. (http://creativecommons.org/licenses/by/4.0/).

EGI-ACE receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101017567.

DELIVERY SLIP

	Name	Partner/Activity
From:	Enol Fernández	EGI Foundation / WP2
Moderated by:	Sjomara Specht	EGI Foundation
Reviewed by:	Ignacio Lamata Martinez	EGI Foundation
Approved by:	SDS	

DOCUMENT LOG

Issue	Date	Comment	Author
v.0.1	09/6/2022	Initial ToC proposal	E. Fernández (EGI Foundation) G. Sipos (EGI Foundation)
v.0.2	01/08/2022	Reviewed specifications for services	M. Antonacci (INFN), V. Ardizzone ((EGI Foundation), P. Bhoumik (ETH Zurich), J. Caballero (STFC), M. Caballer (UPV), A. Calatrava (UPV), I. Collier (STFC), R. Cooper (STFC), E. Fernández (EGI Foundation) W. Karageorgos (IASA), B. Kryza (CYFRONET), N. Liampotis (GRNET), S. Licehammer (CESNET), A. López (CSIC), A. Manzi (EGI.eu), G. Marchetti (CNRS), G. Moltó (UPV), T. Noble (STFC), V. Tran (IISAS), P. Pospíšil (CESNET), D. Spiga (INFN), Z. Šustr (CESNET), R. Wartenburger (ETH Zurich)
v.0.3	02/09/2022	Handle reviewers comments	A.Manzi (EGI Foundation), E.Fernández (EGI Foundation)
v.0.4	27/09/2022	Send to SDS	

v.1	12/10/2022	Final	

TERMINOLOGY

https://confluence.egi.eu/display/EGIG

Terminology/Acronym	Definition
AAI	Authentication and Authorisation Infrastructure
AI/ML	Artificial Intelligence / Machine Learning
API	Application Programming Interface
AWS	Amazon Web Services
CDMI	Cloud Data Management Interface
CI/CD	Continuous Integration and Continuous Delivery
CLI	Command-Line Interface
DOI	Digital Object Identifier
EOSC	European Open Science Cloud
GCP	Google Cloud Platform
GUI	Graphical User Interface
HPC	High Performance Computing
HTC	High-throughput Computing
IaaS	Infrastructure as a Service
IaC	Infrastructure as Code
IdP	Identity Provider
IM	Infrastructure Manager
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
OIDC	OpenID Connect
OLA	Operational Level Agreement
PaaS	Platform as a Service
PID	Persistent Identifier
RDF	Resource Description Framework
REST	Representational State Transfer
SLA	Service Level Agreement
SP	Service Provider
SSH	Secure Shell Protocol
TLS	Transport Layer Security
VM	Virtual Machine
VMI	Virtual Machine Image
VO	Virtual Organisation
VPN	Virtual Private Network
WMS	Workload Manager Service
XML	Extensible Markup Language

Contents

Executive summary

EGI-ACE builds on the computing e-Infrastructure of the EGI Federation to deliver the EOSC[1] Compute Platform: an open, data-centric, distributed, hybrid and secure infrastructure consisting of computing and storage providers and platform services to support research and open science via data spaces.

The EGI-ACE technical architecture provides a platform to deploy Thematic Services that can exploit baseline compute and storage resources from Cloud, HTC and HPC providers via Federated Compute and Data services that facilitate the execution of workloads close to the data to be analysed. Together with user-oriented services, the Core EGI Services cover the operational aspects of the Federation (authentication and authorisation, accounting, monitoring, helpdesk) and bridge the gap between EGI-ACE services and the Core EOSC functionalities.

This document describes the different layers of the EGI-ACE technical architecture and details each of the services that compose these layers. It is an updated version of Deliverable 2.3 [R1], including updated integration modes for providers, details on the integration of HPC providers and an updated description of the different components as they have evolved during the project lifetime.

[1] European Open Science Cloud. https://eosc-portal.eu/about/eosc

1 Introduction

EGI-ACE delivers the EOSC Compute Platform as a fully integrated compute environment that federates distributed hybrid compute and storage facilities to support processing and analytics via a set of services for distributed data and compute access.

The EOSC Compute Platform architecture is organised in functional blocks, as shown in Figure 1.

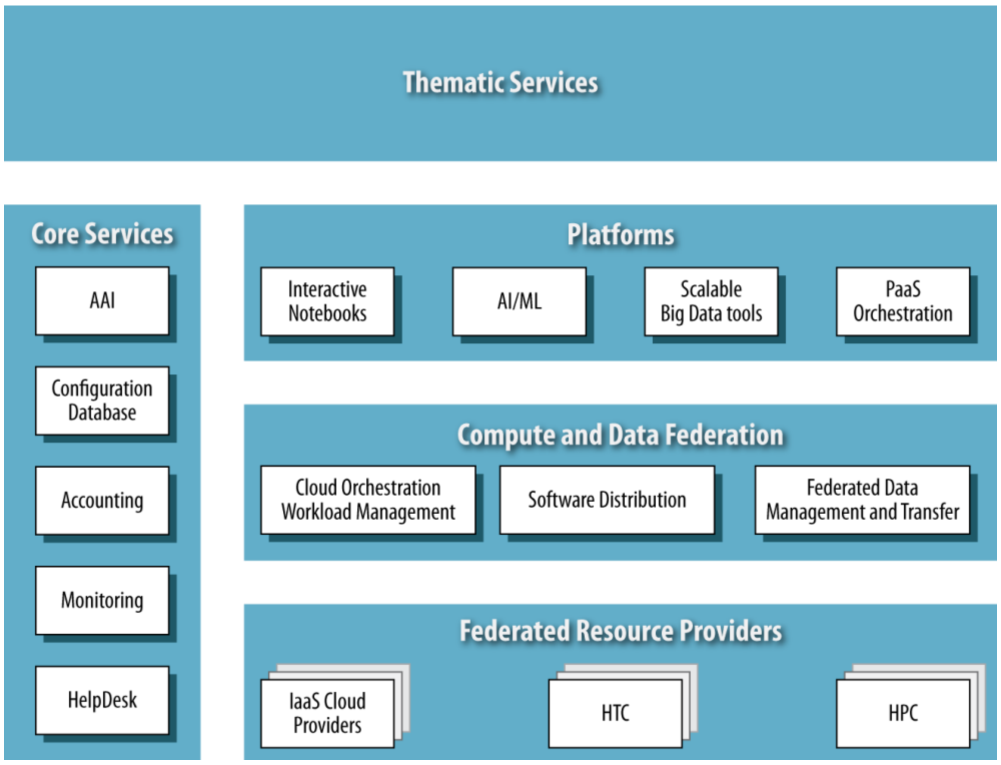


Figure 1. EOSC Compute Platform functional block diagram.

At the bottom of the architecture, the ***Federated Resource Providers*** deliver a hybrid infrastructure for hosting research applications and data. Different types of providers are considered in the EGI-ACE technical architecture:

- **IaaS Cloud Providers** provide access to Virtual Machine-based computing with associated Object and Block storage. These deliver a very flexible and customisable platform where users have complete control over the software and the supporting compute capacity. This flexibility of the computing platform enables the support of a variety of workloads: user gateways or portals, interactive computing platforms and almost any kind of data- and/or compute-intensive workloads.
- **HTC** and **HPC** provide access to large, shared computing systems for running computational jobs at scale. These allow researchers to analyse large datasets and execute thousands of parallel computing tasks without managing the underlying infrastructure. HTC is typically oriented towards the execution of many independent tasks over long times, whereas HPC supports highly optimised applications with

many dependent tasks requiring large amounts of parallel computation together with a low-latency and high-bandwidth interconnection network.

The ***Core Services*** pillar provides the necessary functionality to assist services of all other areas with the integration into the Federation. These services support the operation of the EOSC Compute platform and integrate and interoperate with the EOSC Core as it is developed in EOSC Future[2]. Examples of Core Services are:

- **Configuration Database**, which contains detailed information about the whole infrastructure of the Federation.
- **Accounting,** which tracks the use of resources over time.
- **Monitoring**, which oversees the status of resources and provides an insight on their availability.
- **AAI**, which is implemented via **EGI Check-in**[3], EGI's Authentication and Authorisation service, is a key component of the architecture that enables a single identity manager to be used across all the layers and services of the EOSC Compute platform. This effectively ensures a unified mechanism for controlling access permissions to resources.
- **Helpdesk**, which provides human support to end-users.

These services are described in more detail in Section 2. Other services are also included in this area, ranging from other non-technical services, coordination activities such asOperations Management, and Security and Incident Response.

The ***Compute Federation*** services orchestrate the execution of user workloads in the Federated Resource Providers. These support the exploitation of data locality by moving computation close to the data and facilitate application portability with the support for a diverse range of computing platforms (Cloud IaaS, HTC, HPC) and the interaction with software distribution tools (such as VM images, container images or plain binaries). There are three services considered in the architecture:

- **Hybrid cloud orchestration**, for the deployment of custom virtual infrastructure over multiple IaaS cloud backends;
- **Workload Manager**, for the scheduling and execution of jobs in the federated resource providers (both cloud and HTC/HPC); and a
- **Software distribution**, for making software available at the Federated Resource Providers (*e.g.*, as VM images).

The ***Data Federation*** services support the exposure of discoverable datasets and the staging of data in and out of the EOSC Compute Platform Cloud. The **Federated Data Management** services control the raw storage capacity offered by the Federated Resource Providers to deliver data products that, by using the **Data Transfer** service, can be

[2] https://eoscfuture.eu/
[3] https://www.egi.eu/service/check-in/

transferred between the different EGI-ACE providers and potentially to external data repositories.

The ***Platforms*** service layer provides generic added-value services to exploit the compute and storage resources of EGI-ACE, which can be easily reused by different communities to build thematic end-user services. These platforms rely on the existing Compute Federation and Data Federation services to access the Federated Resource Providers and deliver **Interactive Notebooks** as a tool for interactive analysis of data, **PaaS Orchestration** to facilitate the deployment of complex applications, **AI/ML** to support the models and algorithms required by some scientific fields, and **Scalable Big Data Tools** that can be reused in several research disciplines.

Finally, ***Thematic Services*** is built on top by combining services from all these areas to bring simulation, machine learning and data analytics capabilities that are tailored to the needs of a specific research domain. These data spaces and tools focus on data exploitation. Each thematic service will have tools adapted to the disciplines they target, under disparate arbitrary architectures, so they are not covered in this deliverable.

These function blocks depicted in Figure 1 will be discussed in the different sections of this document.

1.1. Integration Modes for Resource Providers

The EOSC Compute Platform offers a flexible approach to bringing new providers into the Federation to deliver compute and storage resources:

- **Full integration.** These providers are fully integrated into the EGI Federation through the Core Services. It enables the seamless use of multiple clouds for international communities and empowers EGI to establish and monitor Service Level Agreements (SLAs) between users and providers. This integration is a well-documented procedure [R2] that ensures that providers meet all policy and operational requirements, and that the technical integration is in place.
- **AAI Integration**. These providers use EGI Check-in for the authentication and authorisation of federated users, but do not necessarily integrate with the rest of the Core Services. While these services are not part of the EGI Federation, the hosting organisation must comply with EGI security requirements to ensure that it will operate the service in good faith, without deliberately exposing the user to security risks, without claiming intellectual property on the data owned by the user, and protecting sensitive data generated by the user's interaction with the service. EGI Check-in integration enables use from the compute and data federation layers, but it requires individual and manual configuration, as these providers cannot be easily discovered. The lack of certification, monitoring and accounting also prevents the inclusion of services to support SLAs/OLAs, since the availability and reliability of the providers cannot be monitored by EGI.
- **Application Sharing**. Providers joining the application sharing integration mode make virtualised applications and software packages from EGI available to local users. These providers integrate with the Software distribution components of the EOSC Compute Platform.

- **Data Sharing**. EGI DataHub (see Section 5.1) allows providers to replicate scientific datasets and expose existing data to the local cloud provider. This integration mode consists of the deployment of the DataHub components that expose the storage available locally to the Federation, in logical namespaces where datasets are available. Similarly, to other providers that are not fully integrated, these providers cannot be included in the SLAs/OLAs.

Table 1 presents a summary of the different integration options and its features.

Table 1 – Integration modes summary

	Full Integration	AAI integration	Application Sharing	Data Sharing
EGI Check-in	Yes	Yes	Optional	Optional
Monitoring/ Accounting/ Helpdesk	Yes	No	No	No
AppDB/ CernVM-FS	Yes	Optional	Yes	Optional
DataHub	Yes	Optional	Optional	Yes
SLA/OLAs	Yes	No	No	No

A graphical relation of the different integration options and the available services is depicted in Figure 2: *Fully integrated resource providers* shown in the centre of the figure are integrated with EGI Check-in and the rest of EGI Core Services and can be seamlessly used from the other layers of the EOSC Compute Platform. At the top, the *Application Sharing* and *Data Sharing providers* are depicted, which are integrated with the respective components of the Compute and Data Federation layer. On the right, the *AAI integration providers* are connected to EGI Check-in to allow international users to be authenticated by their system.

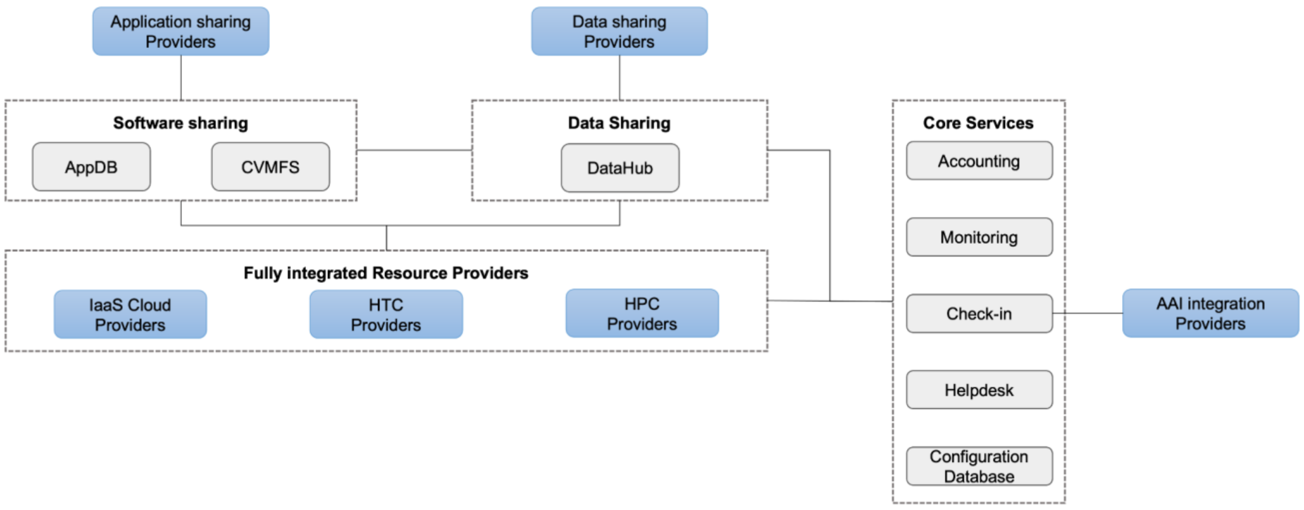


Figure 2. Integration options of the Resource Providers

1.2. Document organisation

The rest of the document is organised as follows:

The main Core Services are described in **Section 2**, which consist of the Authentication and Authorisation service (EGI Check-in), the Configuration Database (GOCDB), the Accounting, the Monitoring, and the HelpDesk (GGUS).

Section 3 provides an overview of the computing and storage services available to the different research communities, defining the cloud environment and the High Throughput Compute (HTC) and High Performance Computing (HPC) services.

The federated compute solution is discussed in **Section 4**, which describes the available services and tools that support end-users in federated operation. These include tools to support user job submission to HTC, HPC and cloud providers, virtual infrastructure deployment and software installation, software distribution and management, and support for dynamic server addressability.

Similarly, the federated data solution is described in **Section 5**. This includes solutions for data management in a federated environment, addressing tasks such as distributed data sources and transfer of data from one provider to another within the federated infrastructure.

Section 6 describes the platforms, which are additional generic tools to exploit the compute and storage resources of EGI-ACE.

Finally, **Section 7** presents some conclusions.

The relationship between the block diagram in Figure 1 and where each part is covered in this document is illustrated in Figure 3.

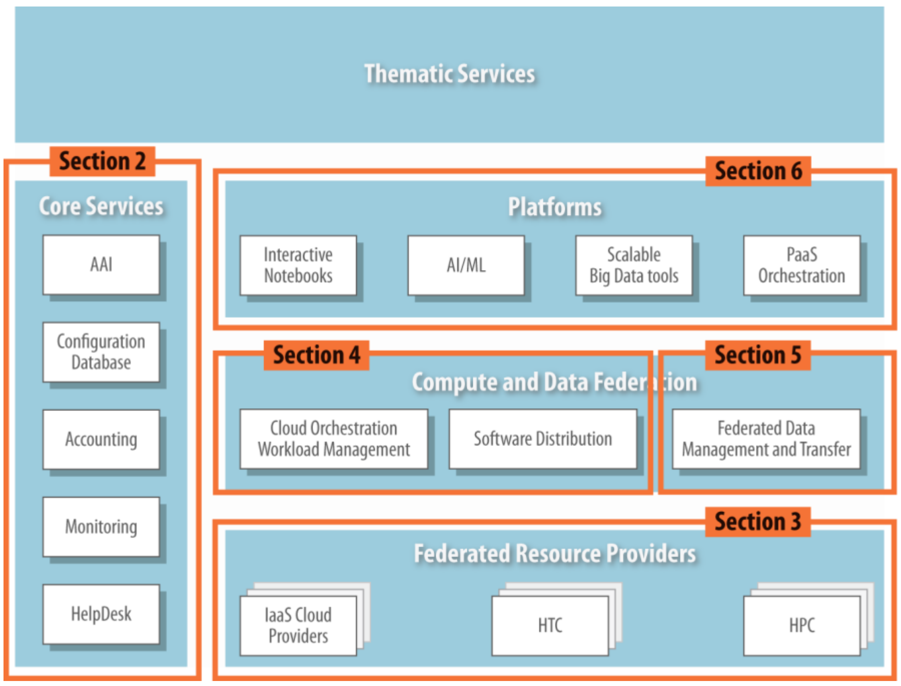


Figure 3. Document organisation

2. Core Services

This section describes the components of the EGI-ACE technical architecture that enable the operation of the Federation. The Core Services are a vertical pillar that integrates with all layers and provides uniform access and management of services. These services are integrated with their EOSC Core equivalents as they become available, allowing EGI-ACE to be integrated with EOSC.

2.1. EGI Check-in: Authentication and Authorisation

EGI Check-in[4] is the authentication, authorisation, and user management service for the EGI infrastructure. It enables users to access both EGI and third-party services (Web and non-Web based), using existing credentials managed by the different Identity Providers (IdPs) of their home organisations. This means that users can authenticate themselves in EGI Check-in by using the usual "login pages" of their respective institutions.

To enable access through academic Identity Providers, EGI Check-in has been registered in eduGAIN[5] as a Service Provider (SP). Through eduGAIN, EGI and third-party services connected to EGI Check-in become available to more than 4,500 Universities and Institutes from 71 National Identity Federations, with little or no administrative involvement. Compliance with the REFEDS Research and Scholarship (R&S) [R3] entity category, the GÉANT *Data Protection Code of Conduct* [R4] and the *Sirtfi security framework* [R5] ensure sufficient user attribute release, as well as operational security, incident response and traceability. Additionally, users without an account on an academic Identity Provider are still able to authenticate themselves by using providers such as ORCID[6], social media (*e.g.,* Google, Facebook, LinkedIn, *etc*) or other external authentication providers. Social media accounts are typically used for accessing EGI and third-party services that do not require a substantial level of assurance[7] (*e.g.,* services that do not require a strict verification of the user identity).

The EGI Check-in service enables users to manage their accounts from a single interface, to link multiple accounts/identities together and to access services based on their roles and group/Virtual Organisation (VO) membership rights.

The adoption of standards and open technologies by EGI Check-in, including *SAML 2.0* [R6], *OpenID Connect (OIDC)* [R7] and *X.509v3* [R8], has facilitated interoperability and integration with the existing Authentication and Authorisation Infrastructures (AAIs) of other e-Infrastructures and research communities, such as B2ACCESS[8] and eduTEAMS[9].

[4] https://www.egi.eu/services/check-in/, see also the webinar at https://indico.egi.eu/event/5494/
[5] https://edugain.org/
[6] https://orcid.org/
[7] https://aarc-community.org/guidelines/aarc-i050/
[8] https://sp.eudat.eu/catalog/resources/d04af0f5-2253-4ee4-8181-3a5a961ccd49
[9] https://eduteams.org/

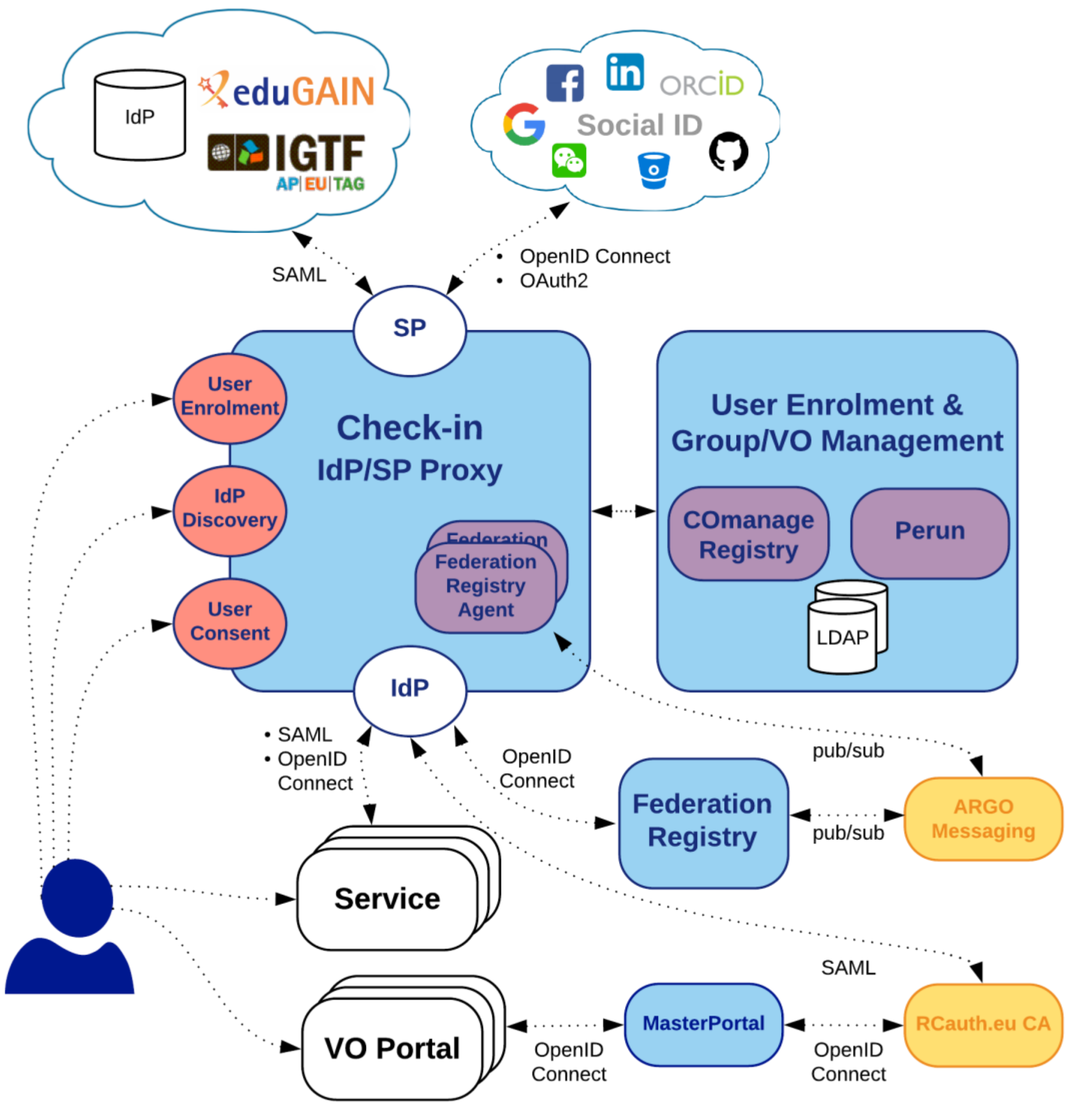


Figure 4. Check-in architecture

Figure 4 shows the EGI Check-in top-level architecture diagram and its interconnections to other AAI services, IdPs and tools. The ***Check-in IdP/SP Proxy*** component acts as a Service Provider towards the external Identity Providers and, at the same time, as an Identity Provider towards the Service Providers (*e.g.,* Applications Database, Cloud providers, *etc*). Through the *Check-in IdP/SP proxy*, users can sign in with the credentials provided by their respective Identity Provider. This is supported by different authentication and authorisation standards, such as SAML 2.0, OpenID Connect 1.0 and OAuth 2.0. The proxy also provides a central Discovery Service (*Where Are You From* – WAYF) for users to select their preferred Identity Provider.

The ***User Enrolment and Group/VO Management*** component, which is based on the **COmanage Registry**[10] tool, supports the management of the full life cycle of user accounts in EGI Check-in. This includes the initial user registration, the acceptance of the terms of use of the infrastructure, account linking, group and VO management, delegation of administration of VOs/groups to authorised users and the configuration of custom enrolment flows for VOs/groups via an intuitive Web interface. Furthermore, the COmanage Registry enables trusted entities to query VO/group information through LDAP. It also provides a REST API that allows authorised clients to manage VO/group membership.

Similarly, EGI Check-in also provides identity management capabilities through the **Perun**[11] tool. Users and groups can be managed directly in the tool by the VO manager or by users themselves following a VO registration flow. Alternatively, Perun can synchronise both users and groups from an external system (*i.e.* VOMS [R9] servers), which can be consequently combined with the aforementioned manual management in Perun itself. All the authorisation information managed in Perun can be provisioned to other systems. To this end, it supports a push model, whereby Perun can be configured to provision the information to the services using standardised mechanisms and customisable connectors. Alternatively, the services can obtain data from Perun using an API or querying some auxiliary services like LDAP. Data from Perun are also available through the *Check-in IdP/SP proxy component*, which releases them as standardised attributes or claims.

The ***MasterPortal*** acts as a caching intermediary service between end-services and the RCauth Online CA[12]. It is designed to provide end-services, such as science gateways or VO Portals, with proxy certificates for their users. The need for an intermediary is two-fold: (i) It is necessary for a scalable trust model based on a single online CA and, secondly, (ii) It hides all the complexity of X.509 certificates and key management for the end-services. In this pure Web flow, the MasterPortal is transparently seen by end-users as a mere 'redirect' between the end-service and the RCauth.eu online CA. In addition to providing portals with proxy certificates, the MasterPortal can also provide end-users with a means to use SSH key authentication to retrieve proxy certificates at the command line. For this functionality, the MasterPortal allows users to upload a SSH public key though a dedicated key management portal.

The ***Federation Registry*** component provides a secure Web interface through which service operators can manage the connection of their EGI Check-in services based on OpenID Connect and SAML. This Web interface covers the whole service lifecycle: the initial registration, reconfiguration, and deregistration. Service configurations are deployed by sending configuration messages to the **Federation Registry Deployment Agents** running on the IdP/SP Proxy component. These configuration messages are exchanged asynchronously through the ARGO Messaging[13] service following Google's Pub/Sub[14] protocol.

[10] https://www.incommon.org/software/comanage/
[11] https://perun-aai.org/
[12] https://rcauth.eu/
[13] https://argoeu.github.io/guides/messaging/
[14] https://cloud.google.com/pubsub/docs

In summary, there are three different options to federate with EGI Check-in to access EGI resources and third-party services:

- Research communities can leverage EGI Check-in for managing their users and their respective roles and other authorisation-related information using either COmanage Registry or Perun.
- Users can authenticate by using a community-specific Identity Provider, such as B2ACCESS, eduTEAMS, or any other Community Authentication and Authorisation service that is compliant with the AARC architectural and policy guidelines[15].
- Resource providers can connect to EGI Check-in to share their resources with the different communities.

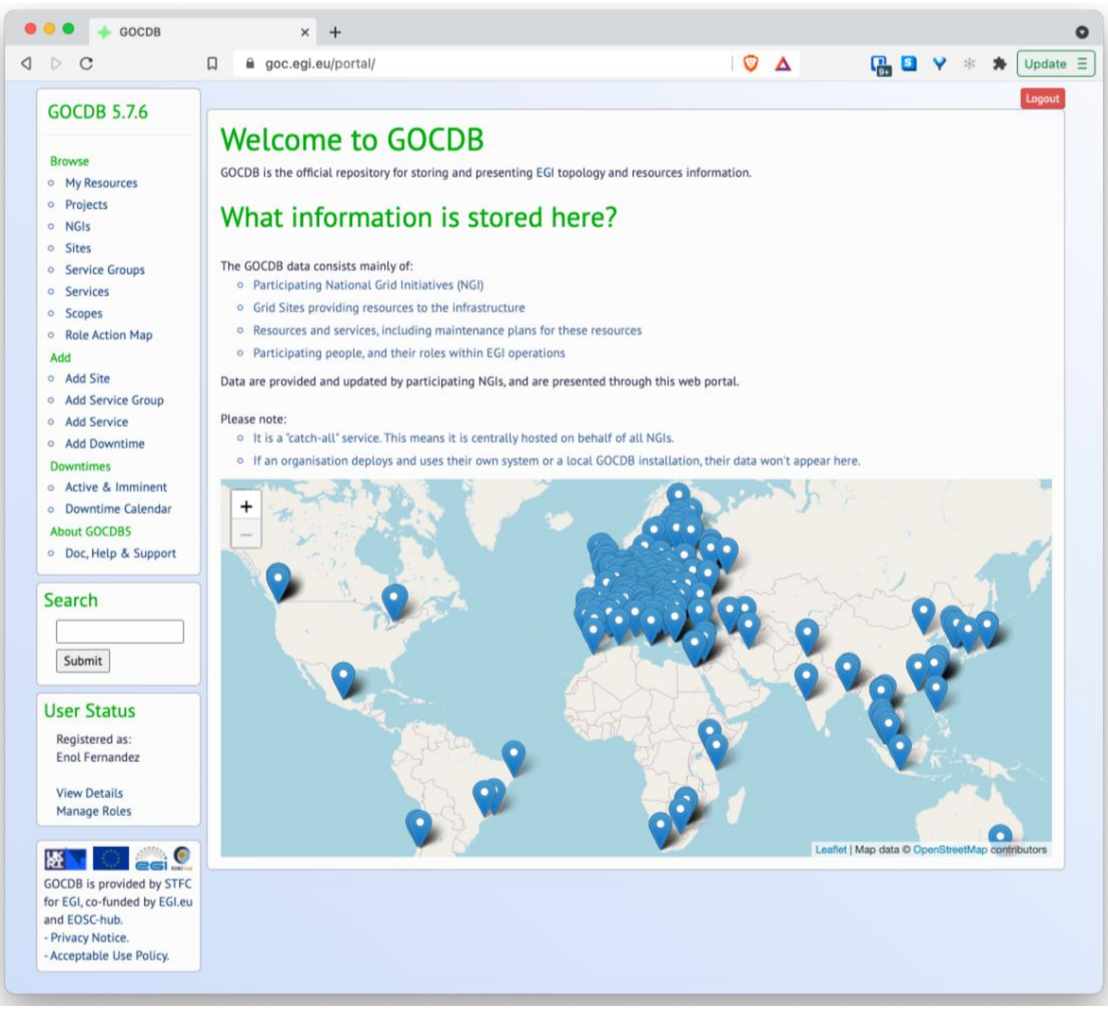
2.2. Configuration Database

The EGI Configuration Database (GOCDB)[16] is a central registry that records topology information about all sites and services participating in the EGI infrastructure. The Configuration Database also provides different rules and grouping mechanisms for filtering and managing the information associated with the resources. This can include entities such as operations and resource centres, service endpoints and their downtimes, contact information and roles of staff responsible for operations at different levels.

The Configuration Database is used by all actors (end-users, site administrators, Resource infrastructure Provider managers, support teams, and VO managers), by other tools, and by third party middleware to discover information about the infrastructure topology.

The tool can be accessed in two ways: (i) Through a Web portal, for inserting/editing information (shown in Figure 5) and (ii) Through a REST style programmatic interface (API), for querying data in XML. Relationships between different objects are defined using a well constrained relational schema that closely resembles a subset of the GLUE 2 [R10] information model. User permissions are controlled by a comprehensive role-based permissions model.

[15] https://aarc-project.eu/guidelines/
[16] https://goc.egi.eu/



GOCDB is provided by STFC for EGI, co-funded by EGI.eu and EOSC-hub.
- Privacy Notice.
- Acceptable Use Policy.

Welcome to GOCDB

GOCDB is the official repository for storing and presenting EGI topology and resources information.

What information is stored here?

The GOCDB data consists mainly of:

- Participating National Grid Initiatives (NGI)
- Grid Sites providing resources to the infrastructure
- Resources and services, including maintenance plans for these resources
- Participating people, and their roles within EGI operations

Data are provided and updated by participating NGIs, and are presented through this web portal.

Please note:

- It is a "catch-all" service. This means it is centrally hosted on behalf of all NGIs.
- If an organisation deploys and uses their own system or a local GOCDB installation, their data won't appear here.

Figure 5. GOCDB web GUI

A flexible tag-cloud mechanism allows objects to be tagged with one or more ‘scope-tags’, which allows resources to be labelled and grouped into multiple categories without duplication of information – this is essential to maintain the integrity of topology information across different infrastructures and projects. Different scope tags can be defined when necessary. For example, tags can be used to reflect different projects, infrastructure groupings and sub-projects. Resources can be flexibly ‘filtered-by-tag’ when querying for data via the Application Programmatic Interface (API). Some tags may be 'reserved', meaning that they can be used only by some authorised sites and services

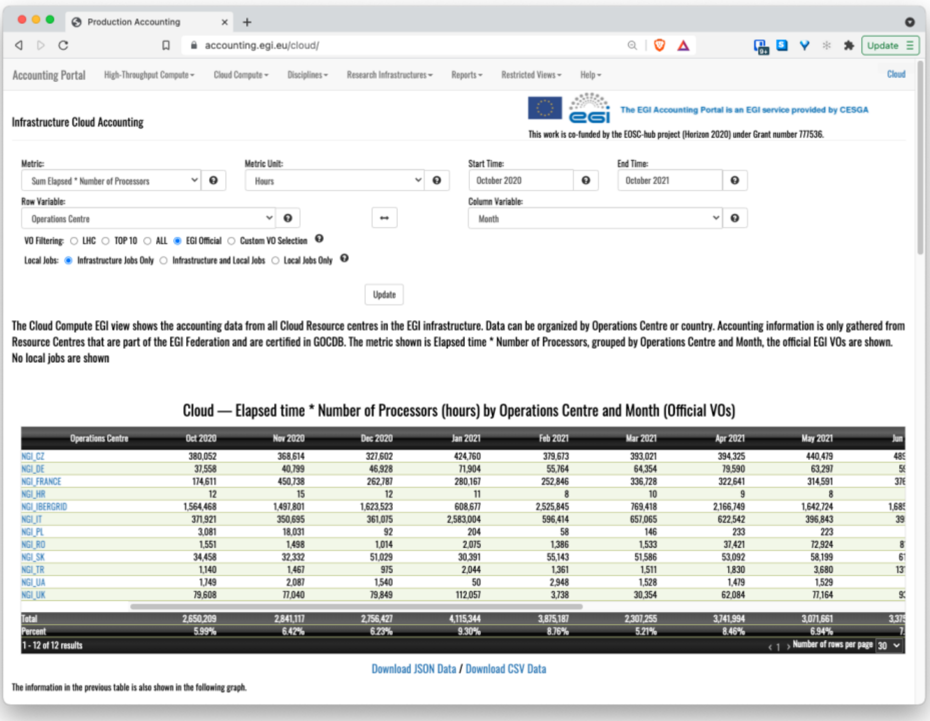
Core objects can also be extended by a powerful extensibility mechanism that allows custom key-value pairs to be added to objects, enabling flexible filtering by custom properties on these objects when selecting and querying data.

2.3. Accounting

EGI Accounting tracks and reports the use of EGI services, offering insights and control over resource consumption. EGI Federation members can use it to account for the resource usage of their own services.

The accounting service collects, stores, aggregates, and displays information about the consumption of resources for High Throughput Compute jobs, Virtual Machines, and online storage providers. Providers supporting these types of services can connect their different service endpoints to the Accounting Service, which is centrally managed and will provide collected data about the service usage.

Probes and sensors that gather accounting information according to certain data formats are deployed locally at the service providers. Data are forwarded from the sensors to a central Accounting Repository where the data are processed to generate various summaries and views for display in the Accounting Portal[17]. Depending on the complexity of the provider, the accounting data may go via intermediate repositories that collate accounting data for particular regions, sub-infrastructures, or communities. EGI-ACE service providers can either directly publish accounting information into the EGI Accounting Repository or can do so via an intermediate repository that serves, for example, a specific region or group of providers. It is up to the provider (group) to use the central repository directly, or to apply an intermediary accounting infrastructure and connect it to EGI.



Infrastructure Cloud Accounting

The EGI Accounting Portal is an EGI service provided by CESGA

This work is co-funded by the EOSC-hub project (Horizon 2020) under Grant number 777536.

Metric: Sum Elapsed * Number of Processors

Metric Unit: Hours

Start Time: October 2020

End Time: October 2021

Row Variable: Operations Centre

Column Variable: Month

VO Filtering: LHC TOP 10 ALL [✓ EGI Official] Custom VO Selection

Local Jobs: [✓ Infrastructure Jobs Only] Infrastructure and Local Jobs Local Jobs Only

The Cloud Compute EGI view shows the accounting data from all Cloud Resource centres in the EGI infrastructure. Data can be organized by Operations Centre or country. Accounting information is only gathered from Resource Centres that are part of the EGI Federation and are certified in GOCDB. The metric shown is Elapsed time * Number of Processors, grouped by Operations Centre and Month, the official EGI VOs are shown. No local jobs are shown

Cloud — Elapsed time * Number of Processors (hours) by Operations Centre and Month (Official VOs)

Operations Centre	Oct 2020	Nov 2020	Dec 2020	Jan 2021	Feb 2021	Mar 2021	Apr 2021	May 2021	Jun
NGI_CZ	380,052	368,614	327,602	424,760	379,673	393,021	394,325	440,479	489
NGI_DE	37,558	40,799	46,928	71,904	55,764	64,354	79,590	63,297	59
NGI_FRANCE	174,611	450,738	262,787	280,167	252,846	336,728	322,641	314,591	376
NGI_HR	12	15	12	11	8	10	9	8	
NGI_IBERGRID	1,564,468	1,497,801	1,623,523	608,677	2,525,845	769,418	2,166,749	1,642,724	1,685
NGI_IT	371,921	350,695	361,075	2,583,004	596,414	657,065	622,542	396,843	39
NGI_PL	3,081	18,031	92	204	58	146	233	223	
NGI_RO	1,551	1,498	1,014	2,075	1,386	1,533	37,421	72,924	8
NGI_SK	34,458	32,332	51,029	30,391	55,143	51,586	53,092	58,199	61
NGI_TR	1,140	1,467	975	2,044	1,361	1,511	1,830	3,680	13
NGI_UA	1,749	2,087	1,540	50	2,948	1,528	1,479	1,529	
NGI_UK	79,608	77,040	79,849	112,057	3,738	30,354	62,084	77,164	93
Total	2,650,209	2,841,117	2,756,427	4,115,344	3,875,187	2,307,255	3,741,994	3,071,661	3,375
Percent	5.99%	6.42%	6.23%	9.30%	8.76%	5.21%	8.46%	6.94%	7.

1 - 12 of 12 results ‹ 1 › Number of rows per page 30

Download JSON Data / Download CSV Data

The information in the previous table is also shown in the following graph.

Figure 6. Accounting portal - cloud view.

[17] https://accounting.egi.eu/

The integration of a service with the Accounting Service requires three steps:

1. Registration of the service in the *Configuration Database* (see Section 2.2), associating it with a geographical or community entity (*e.g.* a country or a community-specific infrastructure).
2. Installation of parsers at the service provider to produce accounting data in the format expected by the Accounting Repository[18]. The parser must be specific to the resource that should be observed from the usage perspective. There are existing parsers already available for:

    a. Grid jobs: https://github.com/apel/apel/tree/dev/apel/parsers
    b. VMs: https://github.com/apel/caso
    c. Storage systems supported by EGI, such as DPM, dCache, StoRM and EOS (see Section 3.2.1), which include the parsers as an integral part of their software releases.

3. Accounting records should be sent to the Accounting Repository (directly or via intermediate repositories) using APEL SSM[19].

The end result of this is the visualisation of information on the service usage (resource consumption) in the Accounting portal. Information from the different individual services can also be used to obtain overall resource usage and generate more useful usage statistics, as shown in the example of Figure 6.

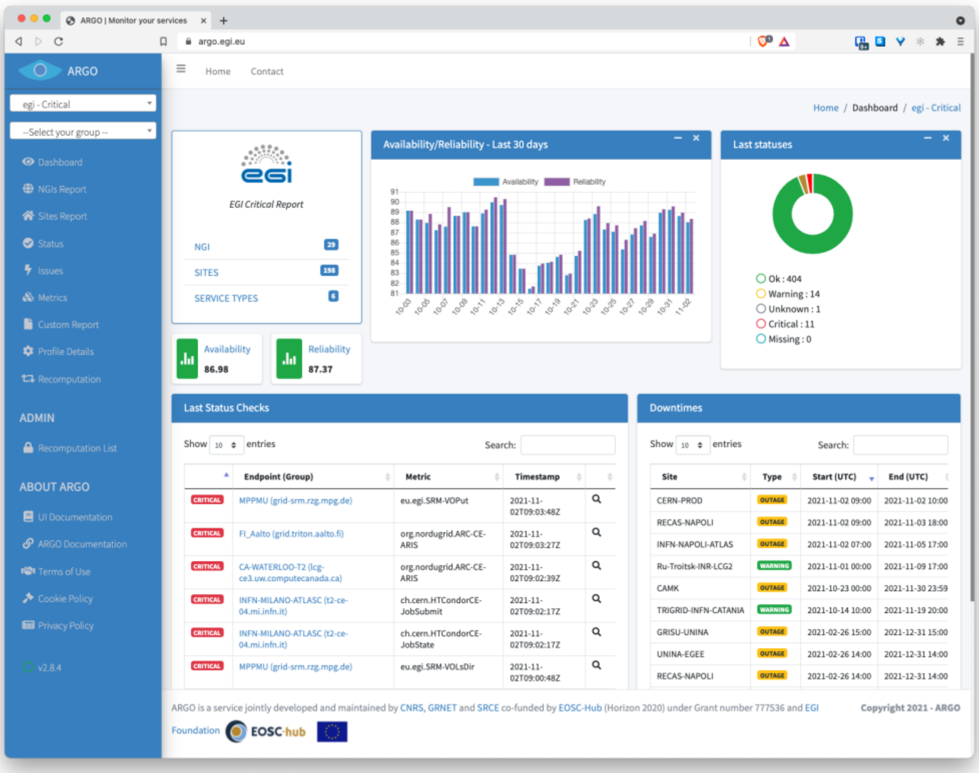
2.4. Monitoring

Monitoring is the mechanism for assessing the state of a service and is a key component needed to gain insights into an infrastructure. It needs to be continuous and on-demand to quickly detect, correlate, and analyse data for a fast reaction to anomalous behaviour. The challenge of this type of monitoring is how to quickly identify and correlate problems before they affect end-users and, ultimately, the productivity of their organisations. The main features of a monitoring system are:

- Monitoring of services.
- Reporting on availability and reliability.
- Visualisation of service status.
- Provision of dashboard interfaces.
- Sending real-time alerts.

Management teams, administrators and service owners can monitor service availability and reliability from a high-level view down to individual system metrics and monitor compliance with multiple SLAs.

[18] Job records format: https://wiki.egi.eu/wiki/APEL/MessageFormat#Job_Records and https://wiki.egi.eu/wiki/APEL/MessageFormat#Summary_Job_Records;
VM usage record format: https://docs.egi.eu/users/getting-started/architecture/#cloud-usage-record;
GPU usage record format: https://docs.egi.eu/users/getting-started/architecture/#gpu-usage-record;
Storage usage record format: https://www.ogf.org/documents/GFD.201.pdf
[19] https://github.com/apel/ssm



	Endpoint (Group)	Metric	Timestamp	
CRITICAL	MPPMU (grid-srm.rzg.mpg.de)	eu.egi.SRM-VOPut	2021-11-02T09:03:48Z	
CRITICAL	FI_Aalto (grid.triton.aalto.fi)	org.nordugrid.ARC-CE-ARIS	2021-11-02T09:03:27Z	
CRITICAL	CA-WATERLOO-T2 (lcg-ce3.uw.computecanada.ca)	org.nordugrid.ARC-CE-ARIS	2021-11-02T09:02:39Z	
CRITICAL	INFN-MILANO-ATLASC (t2-ce-04.mi.infn.it)	ch.cern.HTCondorCE-JobSubmit	2021-11-02T09:02:17Z	
CRITICAL	INFN-MILANO-ATLASC (t2-ce-04.mi.infn.it)	ch.cern.HTCondorCE-JobState	2021-11-02T09:02:17Z	
CRITICAL	MPPMU (grid-srm.rzg.mpg.de)	eu.egi.SRM-VOLsDir	2021-11-02T09:00:48Z	

Site	Type	Start (UTC)	End (UTC)
CERN-PROD	OUTAGE	2021-11-02 09:00	2021-11-02 10:00
RECAS-NAPOLI	OUTAGE	2021-11-02 09:00	2021-11-03 18:00
INFN-NAPOLI-ATLAS	OUTAGE	2021-11-02 07:00	2021-11-05 17:00
Ru-Troitsk-INR-LCG2	WARNING	2021-11-01 00:00	2021-11-09 17:00
CAMK	OUTAGE	2021-10-23 00:00	2021-11-30 23:59
TRIGRID-INFN-CATANIA	WARNING	2021-10-14 10:00	2021-11-19 20:00
GRISU-UNINA	OUTAGE	2021-02-26 15:00	2021-12-31 15:00
UNINA-EGEE	OUTAGE	2021-02-26 14:00	2021-12-31 14:00
RECAS-NAPOLI	OUTAGE	2021-02-26 14:00	2021-12-31 14:00

ARGO is a service jointly developed and maintained by CNRS, GRNET and SRCE co-funded by EOSC-Hub (Horizon 2020) under Grant number 777536 and EGI Foundation Copyright 2021 - ARGO

Figure 7. ARGO Web dashboard

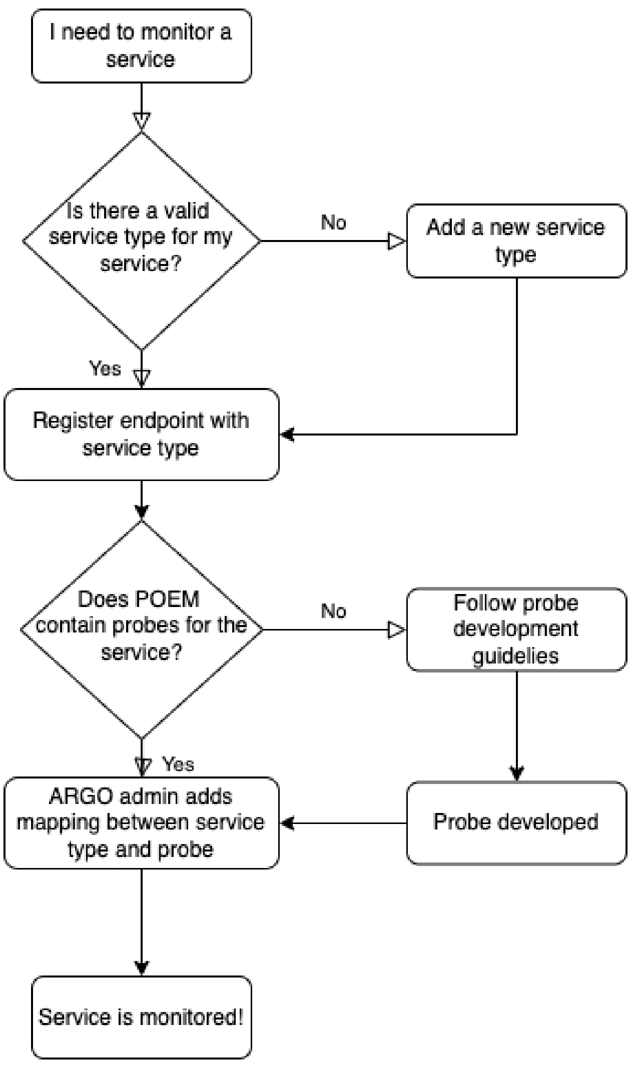
EGI provides a monitoring service based on the ARGO[20] system, which collects status results from one or more monitoring engines and provides status results together with monthly availability (A) and reliability (R) results of distributed services. Both status results and A/R metrics are presented through a Web user interface, shown in Figure 7, which enables users to drill down from the availability of a site to individual services, to individual test results that contributed to the computed figure. ARGO is also capable of sending notifications to service administrators in the event of a failure/warning on one of the monitored services.

For each service monitored by ARGO, there are a set of probes which implement the specific tests for the service to check and at the same time mimic the actual end user behaviour without requiring special privileges or special configurations. These probes are developed by the service owners as they have the inner knowledge about how the service works and how to check its correct behaviour. All the probes intended to be used for monitoring must comply with the ARGO guidelines for monitoring probes [R11].

Figure 8 shows the probe development cycle for any service. ARGO relies on the Configuration Database as a source for the topology of the infrastructure. Every endpoint registered in the database will be monitored according to its service type (*e.g.,* IaaS Cloud provider) using the probe registered in POEM[21]. POEM is ARGO's register of services,

[20] https://argo.egi.eu/, see also EGI webinar https://indico.egi.eu/event/5496/
[21] https://poem.egi.eu/ui/public_probes and https://github.com/ARGOeu/poem-2

repositories, packages and related probes, metric configurations and metric profiles that instruct ARGO monitoring instances what kind of probes/tests to execute.



I need to monitor a service
Is there a valid service type for my service?
No
Add a new service type
Yes
Register endpoint with service type
Does POEM contain probes for the service?
No
Follow probe development guidelies
Yes
ARGO admin adds mapping between service type and probe
Probe developed
Service is monitored!

Figure 8. ARGO probe development workflow

2.5. Helpdesk

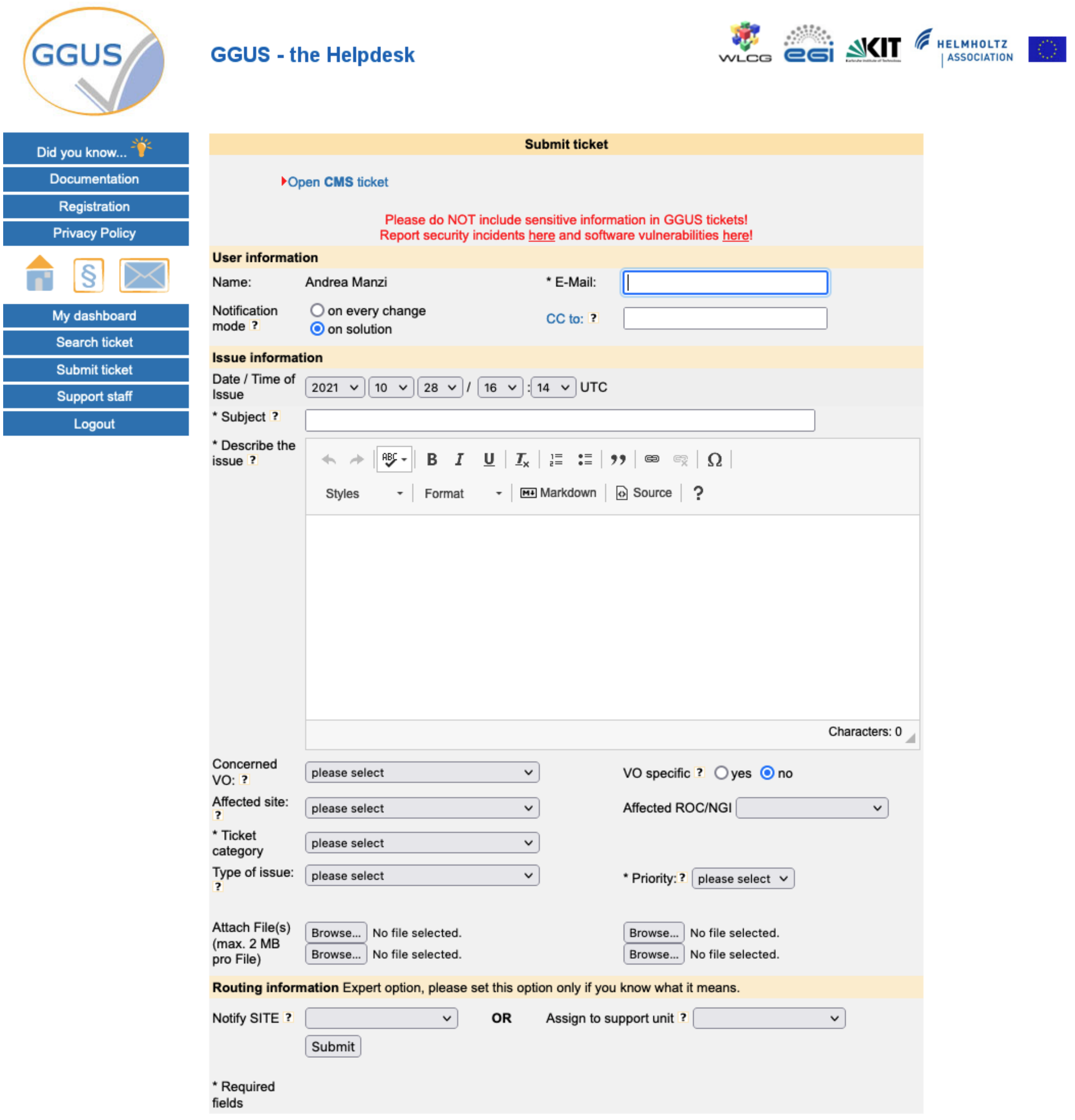
The EGI Helpdesk is a distributed tool with central coordination, which provides the necessary information and support for troubleshooting products and services. It is implemented through the GGUS system[22], where users can report incidents, bugs, and request changes by using the Web user interface shown in Figure 9.

To report issues, the user creates a ticket. Once this ticket is submitted, it is forwarded to the assigned Support Unit (SU) to handle the service support request. The problem is then analysed by a specialist in the scope of the reported problem and the issue is addressed to

[22] https://ggus.eu/

be resolved expeditiously. Depending on the nature of the ticket, interaction with the user may be necessary to request additional information or to confirm the adequacy of the solution.

More information about the EGI Helpdesk can be found in the EGI documentation[23].


GGUS - the Helpdesk

Submit ticket

Open CMS ticket

Please do NOT include sensitive information in GGUS tickets!
Report security incidents here and software vulnerabilities here!

User information

Name:	Andrea Manzi	* E-Mail:	
Notification mode ?	on every change [✓ on solution]	CC to: ?	

Issue information

Date / Time of Issue	2021 / 10 / 28 / 16 : 14 UTC
* Subject ?	
* Describe the issue ?	Characters: 0
Concerned VO: ?	please select
VO specific ?	yes [✓ no]
Affected site: ?	please select
Affected ROC/NGI	
* Ticket category	please select
Type of issue: ?	please select
* Priority: ?	please select
Attach File(s) (max. 2 MB pro File)	No file selected. No file selected. No file selected. No file selected.

Routing information Expert option, please set this option only if you know what it means.

Notify SITE ? OR Assign to support unit ?

* Required fields

Figure 9. The GGUS Helpdesk

[23] https://docs.egi.eu/internal/helpdesk/

3. Federated Resource providers

This section provides an overview of the baseline computing and storage services that deliver access to computing and storage resources for communities to run their workloads on their data.

3.1. Cloud IaaS

Cloud IaaS (Infrastructure as a Service) gives users the ability to deploy and scale Virtual Machines (VMs) on-demand. It offers computational resources in a secure, isolated environment managed through an API, without the overhead of managing physical servers. IaaS supports the following tasks:

- The execution of multiple types of compute- and data-intensive workloads (both batch and interactive).
- Hosting long-running services, such as Web servers, databases or applications servers.
- The creation of disposable testing and development environments on VMs. Users can select specific hardware configurations (CPU, memory, GPU, disk) to run their virtual machines and run any Operating System with complete control of applications and system configuration.

EGI integrates IaaS into a multi-national cloud system that pools resources from a heterogeneous set of providers that deliver an API-controlled service for management of Virtual Machines, associated Block Storage to enable persistence, and Networks to enable connectivity of the VMs. From a technical point of view, each resource centre of the federated infrastructure operates a *Cloud Management Framework* according to its own preferences and constraints. The resource centre joins the Federation by integrating this framework with the different components of the EGI-ACE Core Services and Compute and Data Federation services, according to the preferred integration mode of the provider. Support is granted to OpenStack-based providers (any recent version from OpenStack Mitaka onwards). Complete integration documentation is available at the EGI documentation pages[24].

Beyond the management of VMs, the integration into the EGI Federation brings:

- **Common AAI**, to allow the use of the same identity across all providers.
- **Application sharing,** to facilitate the finding of community-specific software from the providers.
- **Monitoring**, to ensure Availability and Reliability of providers as agreed in Service Level Agreements.
- **Centralised accounting** that collects information on usage in the Federation.

Specific details on Cloud IaaS features are described in more detail in the following sections.

[24] https://docs.egi.eu/providers/cloud-compute/openstack/

3.1.1. AAI integration

IaaS providers integrated with Check-in allow users to access providers with a single identity. Providers in the Federation maintain full control of their services and resources and make authorisation decisions locally by using harmonised user information delivered by Check-in, which includes the Virtual Organisations (VOs) to which each user belongs.

The integration with Check-in is done via *OpenID Connect* and builds on the federated identity features of Keystone[25] (OpenStack's authentication service). EGI provides extensive documentation on how federated user VOs can be mapped to local OpenStack projects[26].

3.1.2. Application Sharing

In a distributed federated IaaS, users need solutions to efficiently manage and distribute their VM images across multiple resource providers. EGI provides *AppDB* (described in Section 4.4) as a catalogue of VM images (VMIs) that can be shared between the different research communities and can be run across the Federation providers.

AppDB allows representatives of research communities (identified by a VO) to generate a list of VMIs following the *HEPiX* image list format[27] to which resource centres can subscribe. This subscription enables the periodic download, conversion, and storage of these images in the local image repository of the provider. The subscription is implemented with the support of the *Cloudkeeper*[28] software package, which also supports the automated synchronisation of images between AppDB and the provider, regardless of the Cloud Management Framework used (OpenStack, OpenNebula, AWS, *etc*).

3.1.3. Compute and Data Federation

The Compute Federation services of EGI-ACE (see Section 4) support OpenStack providers to be manually configured for seamless use by Federation users. Additionally, a piece of software called the Infrastructure Manager, described in Section 4.2, can be used to deploy infrastructure on cloud providers that use systems other than OpenStack, such as OpenNebula, CloudStack, AWS, Azure and GCP.

The Data Federation services of EGI-ACE (see Section 5) are supported by the EGI *DataHub* service, which allows providers to make data available to their local users. To make local data accessible to other EGI providers, the *Oneprovider* component[29], described in Section 5.1, can be deployed. S3-compliant endpoints are also supported by services such as the *File Transfer Service, described in* Section 5.2.

3.1.4. Integration with other core services

In addition to the AAI integration described in Section 3.1.1, providers going through the full integration mode need to complete the *Resource Centre Registration and Certification procedure* (PROC09) [R2] that will result in the creation of new entries in the Configuration

[25] https://docs.openstack.org/keystone/latest/admin/federation/introduction.htmln
[26] https://docs.egi.eu/providers/cloud-compute/openstack/aai/#keystone-setup
[27] https://wiki.appdb.egi.eu/main:faq:vo_image_list_format
[28] https://github.com/the-cloudkeeper-project/cloudkeeper
[29] https://docs.egi.eu/providers/datahub/oneprovider/

Database described in Section 2.2, after meeting all policy, security and quality of service requirements for joining the EGI Federation. Once registered, endpoints will be automatically monitored by ARGO (see Section 2.4). Accounting records can be generated at the provider via cASO[30].

3.2. High Throughput Computing

High Throughput Compute (HTC) is a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks (*e.g.* data analysis jobs). HTC systems execute independent tasks that can be individually scheduled on many distributed computing resources, across multiple administrative boundaries. Users submit these tasks to the infrastructure as *jobs*. After a job has been scheduled and executed, the output can be collected from the service(s) that executed it. The EGI High Throughput Compute[31] infrastructure brings together federated providers delivering HTC resources.

There are two different HTC services considered in EGI-ACE: **Grid providers**, discussed in Section 3.2.1, and **Spider**, discussed in Section 3.2.2.

3.2.1. Grid providers

Grid providers deliver a front-end (ARC-CE[32] or HTCondor-CE[33]) for the submission of jobs to a local batch system (*e.g.* Slurm[34], PBS[35], or HTCondor[36]). These systems rely on X.509 certificate [R8] and VOMS for user authentication and authorisation. Check-in MasterPortal (see Section 2.1) acts as a bridge to interact with grid providers for users relying on federated identity mechanisms.

In addition to job-based computing, Grid providers allow files to be stored in a scalable, fault-tolerant environment and shared with distributed teams. Data can be accessed through multiple protocols (gridFTP and WebDav/HTTP, XRootD and legacy SRM), and can be replicated across different providers to increase fault tolerance. The main implementations supported in EGI's grid storage are: dCache[37], DPM[38] , EOS[39] and StoRM[40].

Grid providers are fully integrated with the Core Services described in Section 2: Configuration Database, Accounting, Monitoring and Helpdesk. Additionally, they deliver information about the resources available via the *BDII* (Berkeley Database Information Index) service. The BDII relies on LDAP to build a hierarchical structure in which all participating providers can be easily discovered.

[30] https://caso.readthedocs.io
[31] https://docs.egi.eu/providers/high-throughput-compute/
[32] https://www.nordugrid.org/arc/ce/
[33] https://opensciencegrid.org/docs/compute-element/htcondor-ce-overview/
[34] https://slurm.schedmd.com/documentation.html
[35] https://www.openpbs.org/
[36] https://htcondor.org/
[37] https://www.dcache.org/
[38] https://twiki.cern.ch/twiki/bin/view/DPM/
[39] https://eos-web.web.cern.ch/eos-web/
[40] https://italiangrid.github.io/storm/

In addition, advanced features are provided by other components described in other sections of the document. Software distribution for supported communities can be enabled by deploying CernVM-FS clients at the providers (see Section 4.3). Advanced job meta-scheduling is handled by the Workload Manager described in Section 4.1, while data transfer and data management are handled by FTS and Rucio, respectively, which are described in Sections 5.2 and 5.3.

3.2.2. Spider

Spider[41] is a versatile high-throughput data-processing platform for processing large, structured data sets provided by SURF[42], running on top of an internal elastic Cloud. It is a feature-rich platform that provides users with a batch processing cluster (based on Slurm) for generic data processing applications, high-performance data access, fast network connectivity to internal and external data centres, support for containers and Jupyter notebooks, and many other user-centric features.

Spider is offered as an alternative for grid users looking for a more customisable and easily accessible system, as the Spider installation shares the same physical data processing infrastructure as SURF's Grid-based processing.

In EGI-ACE, Spider is offered as one of the options for the HTC users and is currently going under integration with the rest of the EGI ecosystem. At the moment of writing, Spider can be registered in the Configuration Database (Section 2.2), it uses CernVM-FS (Section 4.3) for sharing software and can interact with Grid storage systems. Further integration will be provided as the project evolves.

3.3. High Performance Computing

High Performance Computing (HPC) provides highly optimised computing systems that deliver large amounts of parallel computing power to run applications. Similarly, to HTC, it offers a managed service where a fully operational environment runs submitted user jobs. Traditionally, HPC systems are managed in an isolated way and offer limited federation options. In EGI-ACE, a set of pilots explored the integration of several HPC systems into the Federation to make them accessible as part of the EOSC Compute Platform.

In most HPC systems, user access is performed via the Secure Shell Protocol (SSH) [R12] to a set of login nodes where users can interact with the system and submit jobs for their execution. In most cases, SSH credentials used are locally managed, either usernames and password or SSH keys based on public-key cryptography [R13]. For the EOSC Compute Platform, three different access options to these providers are proposed:

- **SSH access with OIDC tokens,** via *ssh-oidc*[43]. This allows SSH access using federated identity technology, as supported by EGI Check-in. This is the preferred mechanism, as it minimises the requirements to the users and does not introduce changes to the traditional mode of operation of HPC centres.

[41] http://doc.spider.surfsara.nl/en/latest/Pages/about.html
[42] https://www.surf.nl/en/research-it
[43] https://github.com/EOSC-synergy/ssh-oidc

- Access via **HTC-like middleware**, where users do not get direct access via SSH, but a set of middleware components handles user access and the interaction with the HPC system. Job submission is performed using the same tools as for an HTC provider, as explained in Section 3.2. While this is the simplest option to implement for the infrastructure, it is not always possible to implement due to the strict policy restrictions of HPC systems.

- **HPC as a Service**. In those providers that offer HPC hardware through IaaS interfaces, small-scale HPC systems can be deployed to run user workloads. While IaaS providers rely on hypervisors[44] to create Virtual Machines, specialised hardware such as GPU accelerators and low-latency networks such as Infiniband can be configured as PCI Passthrough devices without major performance overheads. Integration for these providers would follow the Cloud IaaS model discussed in Section 3.1. Orchestration tools like the *Elastic Cloud Computing Cluster* discussed in Section 6.2 can create elastic virtual clusters on these providers. This approach enables a highly flexible and customisable environment while providing access to HPC features.

Integration with the core services depends on the access mode used for the specific HPC provider and are summarised in Table 2 (refer to D7.3 [R14] for full details).

Table 2 – HPC integration with core services

	ssh access	**HTC access**	**HPC as a Service**
Check-in	ssh-oidc	X.509 certificates via Check-in's MasterPortal	OpenStack integration with Check-in
Configuration Database	edu.kit.ssh_oidc service type	HTC service types (*e.g.,* HTCondor)	OpenStack service types (*e.g.,* nova)
Accounting	Reuse HTC accounting	Reuse HTC accounting	Reuse OpenStack accounting
Monitoring	New dedicated probe for ssh-oidc	Reuse HTC monitoring	Reuse OpenStack monitoring
Helpdesk	Provider registered in EGI Helpdesk		

[44] Software that manages and runs Virtual Machines

4. Federated Compute

The Federated Compute subsystem in EGI-ACE facilitates the use of the underlying computing infrastructure in a homogeneous way, so that users can execute their workloads across the available resource providers in a portable way. This is supported by the following services:

- The **Workload Manager**, powered by *DIRAC*, supports scalable job submission across HTC, HPC and Cloud providers. The Workload Manager takes care of scheduling the user tasks into the most appropriate resources. DIRAC is described in Section 4.1.
- The **Infrastructure Manager** (IM) provides IaaS cloud orchestration by automating the deployment of virtual infrastructure across multiple cloud providers, including public ones, such as AWS, Google Cloud and Azure, amongst others. IM supports hybrid deployments and uses a single description language for all supported backends, thus making applications cloud agnostic. It also provides support with software installation, such as Kubernetes clusters, MinIO and many other tools. IM is described in Section 4.2.
- **CernVM-FS** and **AppDB** support the efficient distribution of software in the federation, so users can just access software from any provider of the infrastructure. They are discussed in Sections 4.3 and 4.4 respectively.
- **DynamicDNS** provides a federation-wide DNS hostname registration service, so users can use memorable names instead of IP addresses for interacting with the virtual infrastructure deployed in the Federation. The service provides tools to dynamically redirect users to the correct virtual machine even when its IP address has changed, which can be a common event in fast-changing virtual environments. The DynamicDNS service is discussed in Section 4.5.

With a few exceptions, most of these services are already integrated (or are in the process of being integrated) with the Core Services described in Section 2: federated users can login using Check-in, services are registered in the Configuration Database, are automatically monitored by ARGO and have support provided through the EGI Helpdesk. The accounting records for these services are not yet defined, so the integration is not yet in place.

Platform services (described in Section 6) can use Federated Compute to abstract the details of each provider and simplify the interaction with the infrastructure. IM serves as the main tool for deploying applications for the *EC3* (Elastic Cloud Computing Cluster), discussed in Section 6.2, and the *PaaS Orchestrator,* discussed in Section 6.3. Services can also be used directly by higher-level thematic services, and the Workload Manager is a service widely used for discipline-specific analytics services that rely on massive submission of jobs to the infrastructure.

4.1. Workload Manager

The Workload Manager Service (WMS) dispatches user's computing tasks in an efficient way while maximising the usage of distributed computational resources. It is built upon the software from the DIRAC Interware project [R15]. The project provides a complete solution for communities needing access to heterogeneous computing and storage resources distributed geographically, integrated in different grid and cloud infrastructures, standalone computing clusters and supercomputers.

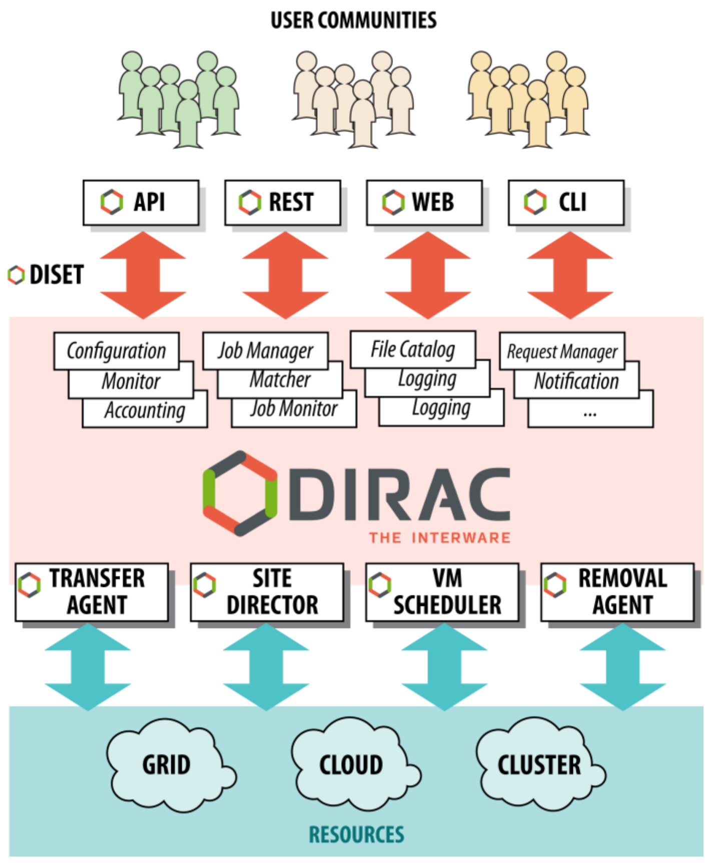


Figure 10. DIRAC Components

Figure 10 illustrates DIRAC components. When user tasks are submitted to the Workload Manager Service, the service performs reservation of computing resources by means of so-called pilot jobs, which are submitted to various computing centres with appropriate access protocols. Once deployed on the worker nodes, pilot jobs verify the execution environment and then request user payloads from DIRAC's central *Task Queue*. Altogether, the pilot jobs and the central Task Queue form a dynamic virtual batch system that overcomes the heterogeneity of the underlying computing infrastructures.

WMS serves multiple scientific communities with a user-friendly interface, and its architecture and job management algorithms provide several crucial advantages:

- Efficient user job execution with a low failure rate.
- Efficient enforcement of resource usage policies for large communities.
- Easy extensions via APIs, to address specific needs.
- A development framework and a set of ready-to-use components to build distributed computing systems of arbitrary complexity.

Examples of the available functionalities are described in the following use cases:

- The workload scheduling architecture allows for easy addition of new resources in a way that is transparent to users. It also facilitates the enforcement of usage policies by defining fine-grained priorities for certain activities. This feature, exploited by the *WeNMR*[45] collaboration during the COVID pandemic, allowed interested sites to share resources to contribute to the COVID-related studies, ensuring a high priority of these studies compared to other regular WeNMR activities.
- The WMS Rest-API may be the submission point of entry for various workflow engines. For example, the *OpenMOLE* open-source platform[46] offers tools to run, explore, diagnose and optimise a numerical model. Singularity jobs prepared with the help of the platform GUI are sent through the DIRAC API to the distributed computing environments dedicated to the exploration of simulation models.
- The DIRAC development framework allows an experiment or a community to integrate a specific tool as a DIRAC WebApp. For instance, the *ConCORDIA* application developed by the *ESCAPE* collaboration[47] provides access to common simulation tools through a GUI integrated into the WMS framework, allowing the conception of Singularity images to run *CORSIKA*[48] simulations (a simulation of extensive air showers induced by high energy cosmic rays) on the distributed computing elements available.

4.2. Infrastructure Manager

The Infrastructure Manager (IM) [R16] is a tool that orchestrates the deployment of complex and customised virtual infrastructures on multiple back-ends. The IM automates the deployment, configuration, software installation, monitoring and update of virtual infrastructures. It supports a wide variety of back-ends, including both public IaaS clouds (Amazon Web Services, Microsoft Azure, *etc*.), on-premises Cloud Management Platforms (OpenNebula, OpenStack, *etc*.), federated platforms (EGI Cloud Compute, Fogbow) and Container Orchestrators (Kubernetes), thus making user applications cloud agnostic. The high-level operation of IM is shown in Figure 11. As depicted in the figure, a typical workflow of IM starts when a user uses the IM Client (Command-Line interface, CLI) or the IM Web (a Web-based GUI) to define the virtual infrastructure to be deployed. This definition is done in text-based software templates in *RADL* (Resource and Application Description Language [R16], the native language of IM) or *TOSCA* (an OASIS standard) [R17] formats. These templates can be created explicitly by the user or automatically (and transparently) generated by IM GUI. IM will then launch the required VMs on the configured cloud provider and, once the virtual hardware has been provisioned, install the required software on it. To support these tasks, SSH connections and Ansible[49] are internally used.

[45] https://www.wenmr.eu/
[46] https://openmole.org/
[47] https://projectescape.eu/news/escape-ossr-enhancing-science-through-sharing-software-benefits-use-cases-post-webinar-report
[48] https://www.iap.kit.edu/corsika/
[49] A tool for automation of IT tasks. https://www.ansible.com/

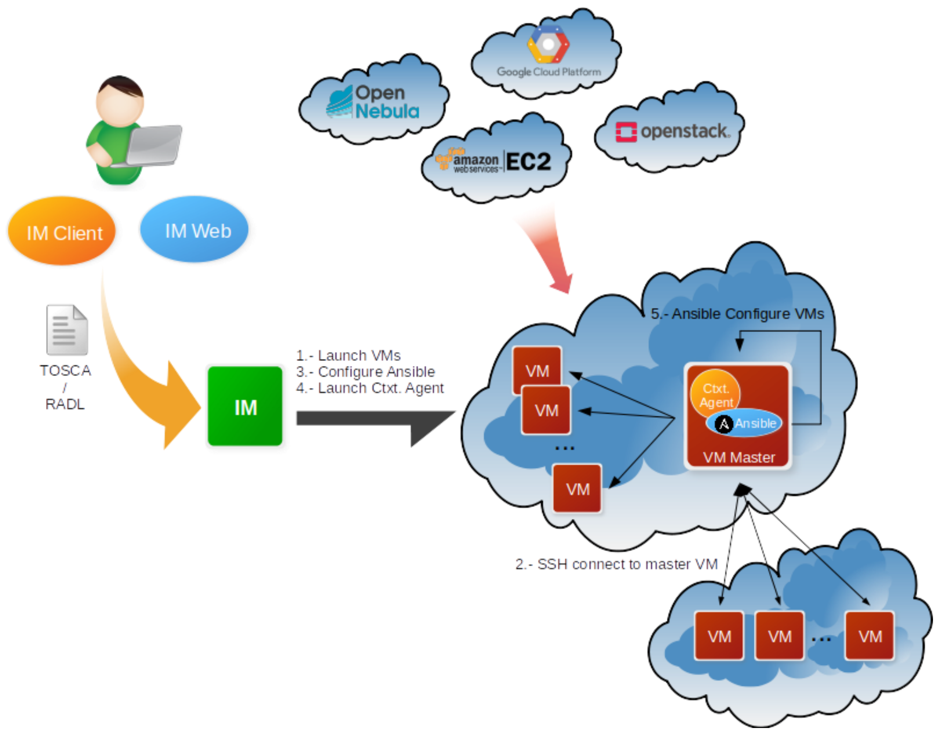


Figure 11. IM Architecture

The use of RADL and TOSCA templates facilitates deterministic repeatability of virtual infrastructure deployments, following an approach called *Infrastructure as Code* (IaC), which is widely adopted by the industry. IaC allows the definition of infrastructure and application architecture as software artefacts, enabling the application of software industry benefits and advanced tools, such as code version control systems, in the management of virtual infrastructure.

IM supports the creation and configuration of clusters consisting of multiple machines with tools such as Hadoop, Slurm, Mesos or Kubernetes, amongst others. Apart from this provision of virtual computing clusters, IM can complement generic Virtual Machine Images by installing and configuring additional software on the Virtual Machine instances, such as MinIO or Galaxy.

The main goal of IM is to provide a set of functions for the effective deployment of all the required virtual infrastructures to deploy an application or service in a Cloud environment, either composed by VMs or by Docker containers. IM considers all the aspects related to the creation and management of virtual infrastructures:

- The software and hardware requirements specification for the users' applications, using a simple language that is easy to understand by non-advanced users who just want to deploy a basic virtual infrastructure, but with enough expressivity so that advanced users can set all the necessary configuration parameters to have the infrastructure fully configured.
- The selection of the most suitable Virtual Machine Images (VMI), is based on the requirements defined by the user.
- The provision of Virtual Machines on a Cloud environment, including public IaaS clouds (Amazon Web Services, Microsoft Azure, *etc.*), on-premises Cloud

Management Platforms (OpenNebula, OpenStack, *etc.*) and Container Orchestrators (Kubernetes).

- Support for hybrid infrastructures, where nodes are spread across different cloud providers, enabling Cloud bursting scenarios.
- The contextualisation of the infrastructure at run-time, by installing and configuring all the required software that may not be available in the pre-configured images (either VMIs or Docker images).
- The elasticity management, both horizontal (adding and removing nodes) and vertical (increasing and reducing node capacity).

IM provides both XML-RPC[50] and REST APIs to enable high-level components to access its functionality. These APIs provide a set of functions for clients to create, destroy, and obtain information about the available infrastructures. It also provides three interfaces:

- **im-client**, which is a Command-line tool available for scripting and automation.
- **im-web**, which is a Web-based interface that has all functionality of IM for advanced users.
- **im-dashboard**, which is a Web-based interface designed with user experience in mind, for users who do not need advanced features and can deploy their virtual infrastructure using the available set of well-tested, predefined TOSCA templates.

More information at https://www.grycap.upv.es/im/ or https://github.com/grycap/im.

4.3. CernVM-FS

The CernVM-File System (CernVM-FS or CVMFS) provides a scalable, reliable, performant, and low-maintenance software distribution service. It was developed to assist High Energy Physics (HEP) collaborations deploy software on the worldwide distributed computing infrastructure used to run data processing applications. CernVM-FS is implemented as a POSIX read-only file system in user space (a *FUSE* module). Files and directories are hosted on standard Web servers and mounted in the universal namespace /cvmfs. Internally, CernVM-FS uses content-addressable storage and Merkle trees to maintain file data and meta-data. CernVM-FS uses only outbound HTTP connections, thereby avoiding network connectivity issues that other network file systems may have when a firewall restricts inbound traffic. It transfers data and meta-data on demand and verifies data integrity using cryptographic hashes [R13].

Comprising a repository service (named CernVM-FS Stratum 0), which provides a single place for users to publish their software, and a global network of replica servers (named Stratum 1), STFC CernVM-FS provides an easy method to publish software and other content, making it instantly available on compute resources around the world.

The integration with EGI Check-in is currently under development, so access is done via *gsi-ssh*, with X.509 certificates used for authorisation. Communication with EGI Check-in to retrieve users' information and map them to the right local Unix account will be done by a PAM module developed by STFC, the Science and Technology Facilities Council, in the UK.

[50] http://xmlrpc.com/

4.4. AppDB

The EGI Applications Database[51] (AppDB) is a service that stores and provides public catalogues of software solutions in the form of native software products and virtual appliances that can run on EGI infrastructure. A *virtual appliance* is simply a preconfigured Virtual Machine image, which has a specific Operating System and relevant software packages pre-installed. AppDB aims to provide end users with scientific software that is both easy to find and easy to deploy in a cloud provider. It also aims to facilitate access to the latest version of scientific software to administrators and VO managers and to monitor availability as well as security issues related to VM image use. As such, AppDB is involved in the distribution, deployment, and management process of virtual appliances on the EGI cloud infrastructure. Its main components are the *Virtual Appliance Catalogue*, the *VO wide image list catalogue*, and the *VMCaster subscription service*, portrayed in Figure 12 and described below.

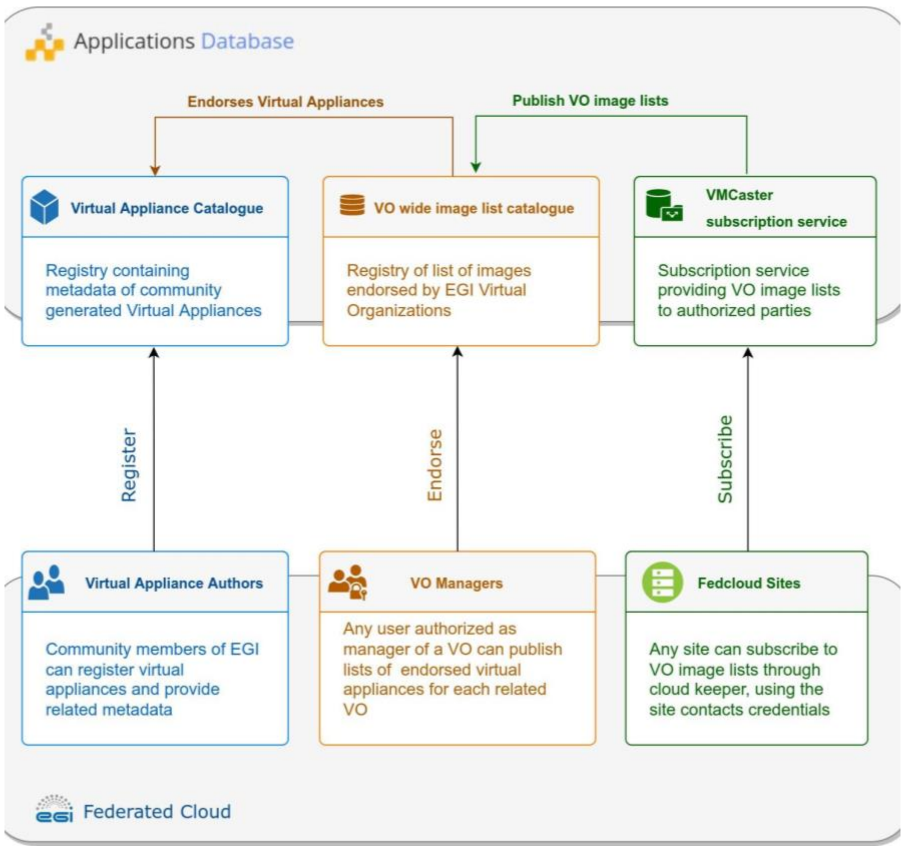

Applications Database
Endorses Virtual Appliances
Publish VO image lists
Virtual Appliance Catalogue
Registry containing metadata of community generated Virtual Appliances
VO wide image list catalogue
Registry of list of images endorsed by EGI Virtual Organizations
VMCaster subscription service
Subscription service providing VO image lists to authorized parties
Register
Endorse
Subscribe
Virtual Appliance Authors
Community members of EGI can register virtual appliances and provide related metadata
VO Managers
Any user authorized as manager of a VO can publish lists of endorsed virtual appliances for each related VO
Fedcloud Sites
Any site can subscribe to VO image lists through cloud keeper, using the site contacts credentials
Federated Cloud

Figure 12. Software distribution components of AppDB

- **Virtual Appliance Catalogue**: It is a community-driven catalogue containing information about virtual appliance solutions, which is authored by its members. VM authors can register their solutions by providing key metadata such as a description, categorisation data, related organisations, projects, *etc*. They may also maintain

[51] https://appdb.egi.eu

version-specific information, such as the physical location of each VM version, its expiration date, resource, and network requirements, supported operating systems / architectures, and so on. To improve quality-of-service, AppDB automatically calculates the checksums of VM images for integrity verification and provides a continuous delivery system that automates the publishing of new VM versions without manual intervention. Virtual Appliances have an expiration date of maximum one year, to ensure that any obsolete software which may contain vulnerabilities is not used in the infrastructure.

- **VO wide image list catalogue**: Virtual Organisations are linked to a list of virtual appliances that the VO is allowed to use. To support this, AppDB integrates with *EGI Operation Portal*[52] to collect information about VOs. VO managers can compose, publish and reuse particular versions of virtual appliances from the Virtual Appliance Catalogue for their specific VO. To help keep image lists up to date with the virtual appliance catalogue, AppDB notifies VO managers about new versions and pending policy-based expiration of endorsed virtual appliance images.

- **VMCaster subscription service**: This component provides authorised third parties, such as EGI cloud providers, with the latest list of virtual appliances assigned to a VO by the corresponding VO managers. This list is in HEPiX format[53] and allows cloud providers to identify the endorsed virtual appliances of a VO. To retrieve this information, the cloudkeeper[54] software or any other compatible solution can be used.

To keep the aforementioned components synchronised with the state of the EGI infrastructure and to keep the distributed information consistent, AppDB has been integrated with various services of the EGI ecosystem, aggregating and correlating information. This has resulted in additional services that can be used by third-party services, in addition to AppDB itself. Such services include the *Information System*[55], which provides cloud infrastructure information, and the *VMOps dashboard*[56], which allows users to deploy and manage VMs on the infrastructure.

4.5. Dynamic DNS

The Domain Name System (DNS) is the mechanism used on the Internet to identify computers by translating network addresses and human-friendly names. It plays a crucial and centric role on the Internet and, especially, on the Web. EGI's Dynamic DNS[57] service provides a unified, federation-wide dynamic DNS support for VMs in EGI's Federated Cloud

[52] https://operations-portal.egi.eu/
[53] https://github.com/hepix-virtualisation/image_list_format_docbook
[54] https://github.com/the-cloudkeeper-project/cloudkeeper
[55] https://is.appdb.egi.eu/rest/
[56] https://dashboard.appdb.egi.eu/vmops
[57] https://docs.egi.eu/users/compute/cloud-compute/dynamic-dns/

infrastructure. Users can register their chosen meaningful and memorable DNS hostnames in given domains (*e.g.* my-server.vo.fedcloud.eu) linking them to the respective public network addresses of their servers (*e.g.* 134.1.2.3). This DNS service is dynamic because it provides a mechanism for dynamic update of network addresses, so that the network address of a server can be updated in the DNS service if it changes, without manual intervention. This is especially useful in a virtual environment, where machines may be frequently removed and recreated, changing their network addresses but needing to keep the same DNS hostname to be located.

The architecture of the Dynamic DNS is depicted in Figure 13. The core component of the service is the *NS-update server,* which consists of the GUI portal and NS-update engine. Users can access the Web portal via EGI Check-in and register hostnames in any of the supported pre-defined domains. After hostname registration, users can assign and update hostname IP addresses via simple commands. All changes are immediately sent to the responsible DNS servers, currently located both at IISAS Bratislava (Slovakia) and LIP Lisbon (Portugal) for high availability. IP changes are refreshed within 60 seconds, so new and modified DNS entries are visible to users in a short time.

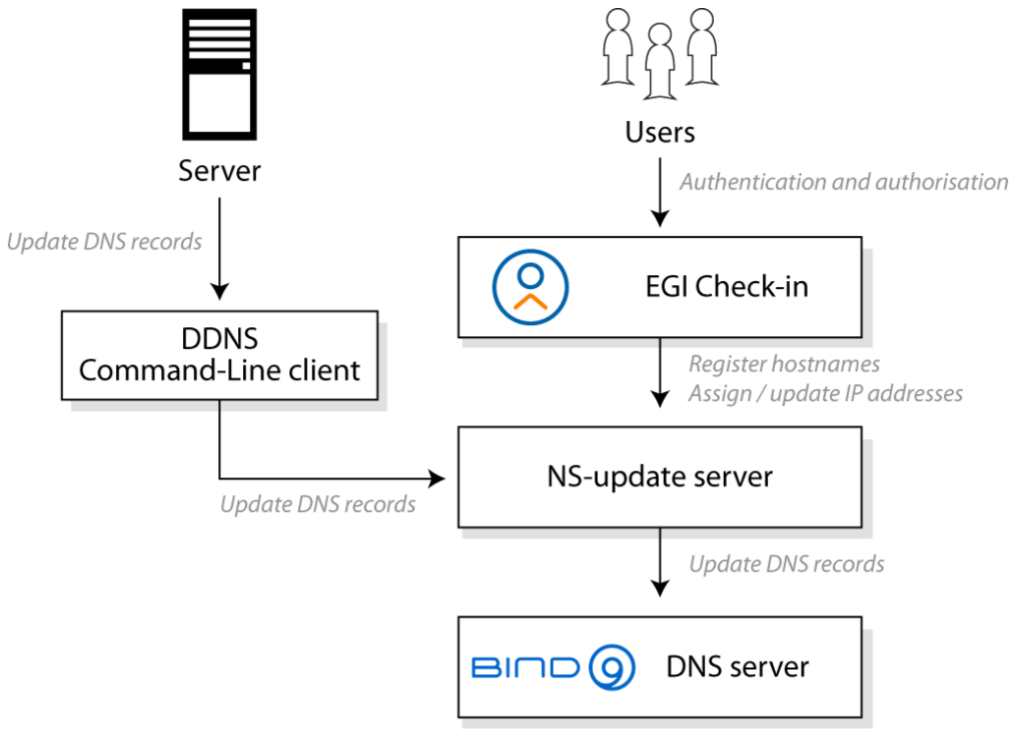


Figure 13. Architecture of Dynamic DNS service.

Some advantages of Dynamic DNS are:

- **Simplifies client and server configuration** for services hosted in the EGI Federated Cloud.
- **Enables a federated approach** in which VMs can be freely moved between providers, maintaining their names without modifying server or client configurations.

- **Facilitates the use of trusted X.509 server certificates**, which are typically linked to DNS names.
- **Provides a user-friendly experience**, with an extremely simple use that requires no additional software installation or support from cloud providers, and network address updates can be performed without special privileges or user credentials.
- **It can be easily integrated** with other tools and services that manage virtual infrastructure, such as the Infrastructure Manager, described in Section 4.2, or Terraform[58].

[58] https://www.terraform.io/

5. Federated Data

The Federata Data Subsystem in EGI-ACE comprises a group of services that provide data management capabilities to enhance the raw storage capacity delivered by the Federated Resource Providers. These services are:

- **DataHub**, a data management solution that enables unified data access to distributed data. This is discussed in Section 5.1.
- **FTS**, a system to address the efficient transfer of data within the data providers of the EGI Federation and is described in Section 5.2.
- **Rucio**, a data management service for managing large volumes of data from multiple sites, which is covered in Section 5.3.
- **openRDM**, which provides advanced data organisation for research projects by offering an integrated environment with data management and digital lab Notebook capabilities. This is discussed in Section 5.4.

These federated data services offer APIs and CLIs that integrate with both the *Thematic Services* and the *Platforms* available in EGI-ACE (see Figure 1). For instance, the *Notebooks service* described in Section 6.1 is integrated with DataHub to allow users to access datasets available in the EGI Infrastructure and share outputs amongst them, and the *PaaS orchestrator* described in Section 6.3 is integrated with Rucio to optimise the deployment of applications close to where the data are located.

Similarly, to Federated Compute services, the Federated Data services are integrated with the Core services described in Section 2, thus allowing Federated Identity access, Configuration Database, Monitoring and Helpdesk. The accounting records for these services are not yet defined, so the integration is not yet in place.

5.1. DataHub

EGI DataHub[59] is a federated service that allows users to access and share their data from any location. Data sets can be kept private to the owner of the data, shared with a restricted number of users, or made publicly available and discoverable via DOI [R18] or PID [R19] handles.

EGI DataHub is based on *Onedata* distributed data access and management system[60]. Onedata is a globally distributed storage solution, integrating storage services from various providers using heterogeneous underlying technologies, such as Ceph, S3, GlusterFS, WebDAV, OpenStack SWIFT, NFS and other POSIX-compliant file systems. It provides access to clients interfaces based on CDMI [R20], REST API and virtually mounted POSIX filesystems.

[59] https://datahub.egi.eu
[60] https://onedata.org

The main functional components of Onedata include:

- **Onezone**, the federation and authentication service. EGI DataHub, as a Onezone instance, provides a single sign-on to a network of connected storage providers. Note that DataHub is integrated with EGI Check-in (described in Section 2.1).
- **Oneprovider**, which is the main data management component of Onedata, is deployed in the data centres and is responsible for provisioning the data and managing data transfers. The main operation of Oneprovider is shown in Figure 14.
- **Oneclient**, which provides access to the virtual filesystem on a VM or host directly, via a FUSE mountpoint.
- **OnedataFS**, which is a Python library supporting the access to the distributed virtual filesystem directly from Python applications (*e.g.,* Jupyter Notebook).

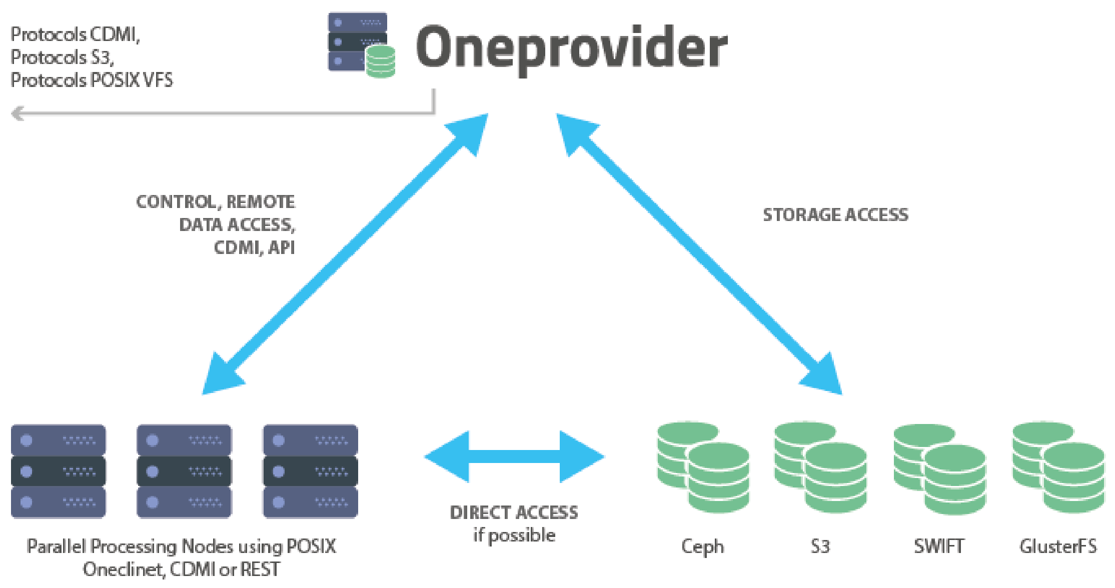

Protocols CDMI,
Protocols S3,
Protocols POSIX VFS
Oneprovider
CONTROL, REMOTE DATA ACCESS, CDMI, API
STORAGE ACCESS
DIRECT ACCESS if possible
Parallel Processing Nodes using POSIX Oneclinet, CDMI or REST
Ceph
S3
SWIFT
GlusterFS

Figure 14. Functional components of Onedata

An important aspect of Onedata is that it has a flexible mechanism for storing metadata in the form of simple key-value pairs, as well as entire metadata documents (currently in JSON and RDF [R21] formats), which can be attached to data resources and used during indexing and querying. On top of this metadata mechanism, Onedata allows users to publish their data as open access content and enables full open data management life cycle management from ingestion through curation to open access thanks to its support for protocols and standards for open data, such as *OAI-PMH* [R22] and handle system integration for registering DOI [R18] identifiers.

Onedata enables seamless data sharing between users, with strict access control. Users can share access to individual files as well as spaces by sending automatically generated access tokens[61]. Onedata has also built-in support for *Let's Encrypt*[62] certificates, encouraging the deployment of secure services with valid X.509 certificates for TLS. Access control is fine-grained and managed by a set of privileges assigned in the form of access control lists at user, group, and space level. Users can create groups for collaborating on a specific space or set of spaces.

All Onedata components have APIs[63] defined using the *OpenAPI specification* [R23], enabling easy integration and automatic generation of client libraries for most existing programming languages and frameworks.
The API's provided by Onedata include:

- **Onezone API**, to allow control and configuration of local Onezone service deployments, in particular: management of users, groups, spaces, shares, providers, services, handles and clusters.
- **Oneprovider API**, to enable access to data through CDMI- compatible endpoints, as well as data management related tasks such as data replication.
- **Onepanel API,** to allow administrators to control deployment of other Onedata components, modifying their configuration - *e.g.* adding more nodes or new storage resources.

For additional information, please refer to Onedata[64] and EGI DataHub[65] user documentation.

5.2. FTS

FTS or FTS3 (File transfer System, version 3.0)[66] is software for efficient scheduled transfers of large data sets. It does so by maximising the usage of available network and storage resources while ensuring that policy limits are respected. FTS3 has the following components:

- **CLI clients**, as the main user interface to interact with FTS3.
- A daemon process for submitting data transfer jobs, retrieving status and general VO and service configuration.
- A daemon process for staging files from tape archives to disk
- A database back-end.
- A message queue daemon which sends monitoring messages to external queue
- A Rest Listener as access points for submitting transfers or administering the service.

[61] Typically, a randomly generated string that acts as a temporary password.
[62] https://letsencrypt.org/
[63] https://onedata.org/#/home/api
[64] https://onedata.org/#/home/documentation/doc/user_guide.html
[65] https://docs.egi.eu/users/datahub/
[66] https://fts.web.cern.ch/fts/, see also EGI webinar https://indico.egi.eu/event/5711/

This architecture allows the service to be easily scalable by adding additional resources with identical configuration into an FTS3 cluster. Figure 15 identifies the main architecture components of FTS.

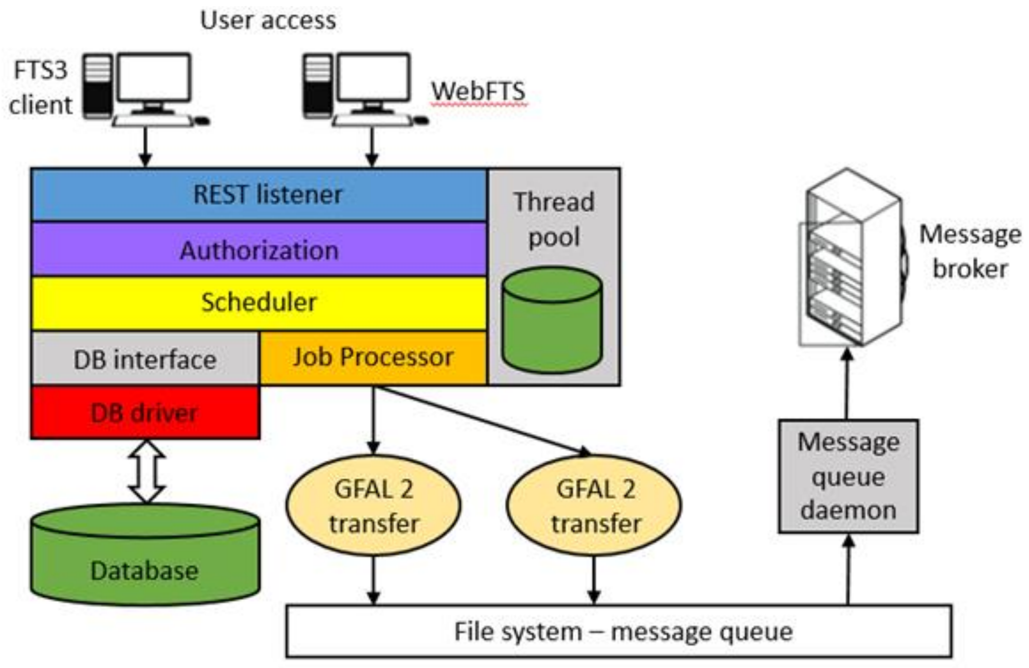


Figure 15. FTS3 service architecture

The optimisation algorithm is influenced by both the throughput achieved and the success rate of the transfers, and automatically adjusts the number of concurrent transfers accordingly. This allows transfers between two random endpoints to be executed with good reliability and throughput, with minimal manual intervention. FTS3 also includes a multi-dimensional scheduler, which ensures fair network data bandwidth between VOs that share a data link. This default behaviour can be overwritten if necessary, in multiple ways, such as giving more transfer slots to a particular VO, splitting the allocated slots by a specific weighting, or prioritising specific transfers to give them a transfer boost.

Some additional features of FTS3 are:

- **Multi-protocol support** through the Grid File Access Library, version 2 (GFAL 2[67]).
- **A REST interface** to submit jobs and query their state.
- **Third party data copy** support, in particular GridFTP[68] and XrootD[69] and, for certain storage implementations, HTTP TPC[70].
- **Session reuse**, which is particularly suitable for cases where many small files are transferred between two endpoints.
- Staging files from tape systems and monitoring of archive to tape operations

[67] https://dmc-docs.web.cern.ch/dmc-docs/gfal2/gfal2.html
[68] https://gridcf.org/gct-docs/latest/gridftp/key/index.html
[69] https://xrootd.slac.stanford.edu/
[70] https://twiki.cern.ch/twiki/bin/view/LCG/ThirdPartyCopy

- Web **monitoring**.

The core functionality of the service is extended by the inclusion of ***WebFTS***, a standalone Web-based interface that provides easy access to FTS3 features for end-users. WebFTS provides the same multi-protocol support as FTS3, but with a two-panel interface (shown in Figure 16) through which users can submit and monitor data transfers instead of using the command-line client, making it a very useful tool for transferring files between Grid and non-Grid resources.

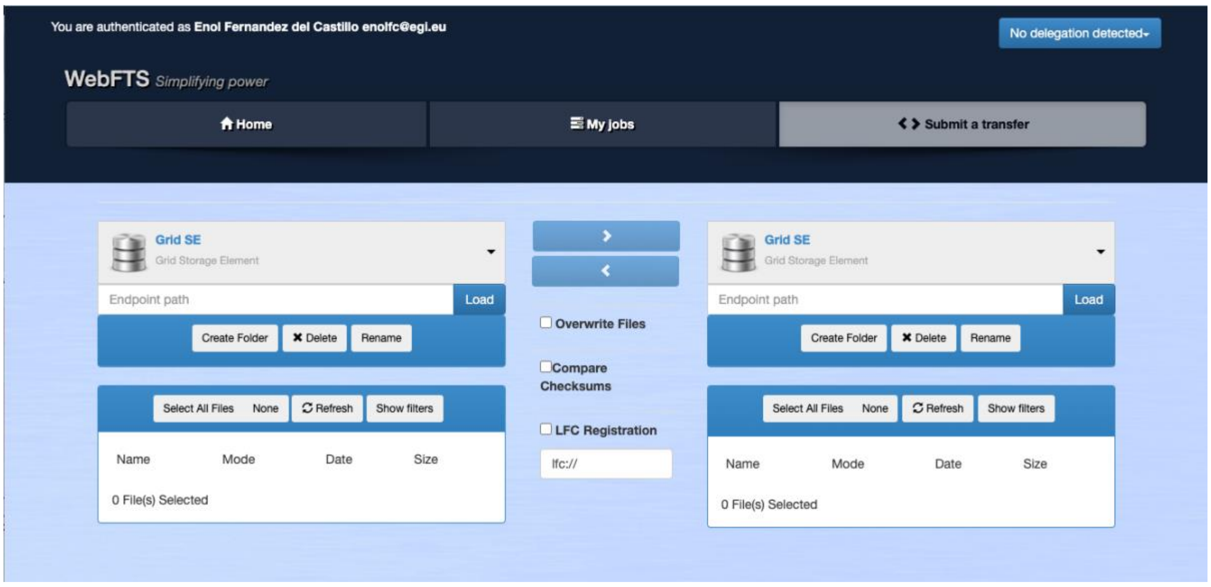


Figure 16. WebFTS transfer submission interface

Access to FTS3 is currently done through X.509 certificates or proxy[71], but OIDC support is expected to be added, which will allow integration with EGI Check-in.

5.3. Rucio

Rucio[72] is a data management piece of software designed to manage large volumes of data across multiple sites and a variety of storage endpoints. It was developed by ATLAS[73] as an open source project but is now used and developed by a variety of communities inside and outside of the High Energy Physics field. The Rucio instance at *Rutherford Appleton Laboratory* (RAL) can support multiple VOs at the same time, leveraging the power of Rucio to support the EGI-ACE use cases with a robust and performant data management solution that is being developed with the future of data and users in mind.

As depicted in Figure 17, a Multi-VO Rucio instance has the following components:

- The **Rucio database**, which stores the state of the service and the file namespace.

[71] See RFC 3820 X.509 Proxy Certificate Profile https://www.ietf.org/rfc/rfc3820.txt
[72] https://rucio.cern.ch/, see also EGI webinar https://indico.egi.eu/event/5711/
[73] https://atlas.cern/

- The **Rucio server**, is implementing the service logic.
- A **CLI client**, which is the main user interface to interact with Rucio server.
- **Daemon processes**, implements the connection with the FTS server or directly with storage systems.
- **A tape archive**, is used to backup the Rucio database.

Rucio orchestrates the transfer of data according to the rules created by the users. Rucio moves data by communicating with the FTS instance at STFC (the Science and Technology Facilities Council in UK, see Section 5.2 for more information about FTS) to transfer the data between storage endpoints.

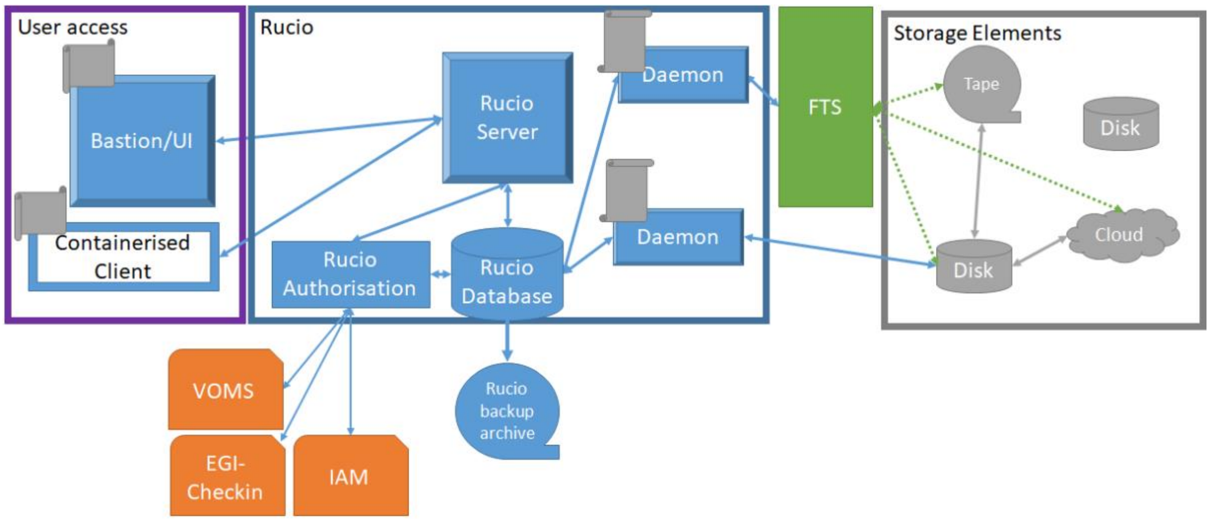


Figure 17. Architecture of the Rucio service. Check-in/IAM integration is under development.

Users can access Rucio in two ways:

- Using the Bastion, a VM hosted at RAL.
- Using a containerised client that the user can install on their own machines.

Both systems provide the same CLI interface, but the containerised client allows local volumes to be mounted for easy uploading and downloading of data. Rucio also has a Web-based interface, which is planned to be functional within Multi-VO Rucio and will allow users to manage their data, account, and account properties using a Web interface.

Authentication and authorisation in Multi-VO Rucio can be done by means of X.509 certificates, an X.509 proxy, or the traditional (and less secure) username and password. Integration with EGI-Check-in (see Section 2.1) is planned by the end of 2022, which will allow a wider range of users to authenticate with Rucio as a service.

To ensure redundancy and data preservation, Rucio's database is backed up daily in an external archive. There are also plans to implement a failover database for faster recovery from downtime caused by a database outage.

Each VO will inform Rucio of its storage endpoints by configuring them via the admin interface. Rucio and FTS support a wide range of protocols and endpoints. VOs can customise the permissions model and the schema for data placement with the use of Rucio policy packages, which are Python packages written and maintained by the VO. Generic policy packages are provided with Rucio, which can be reused and adapted by the VO.

Multi-VO Rucio's goal is to assist EGI communities with its powerful Rucio software as a service, helping them prepare to use Rucio, registering accounts, configuring storage endpoints, customising policy packages, and providing a point of contact for troubleshooting and assistance.

5.4. openRDM

openRDM is a service for storing inventory and datasets associated with the experiments conducted in a laboratory during a research activity. It implements a digital notebook to describe the computational experiments and link it with the materials, samples and protocols stored in the inventory. Additionally, it also allows to perform data analysis (*e.g* via Jupyter) and to execute workflows in computing clusters.

The service is based on the openBis Active Research Data Management (ARDM) platform[74], which has been developed over the last twelve years by the Scientific IT Services of Informatikdienste (ID SIS) at ETH Zurich. ARDM is the process of organising data during an ongoing research project, consisting of annotation, storage and backup of the data.

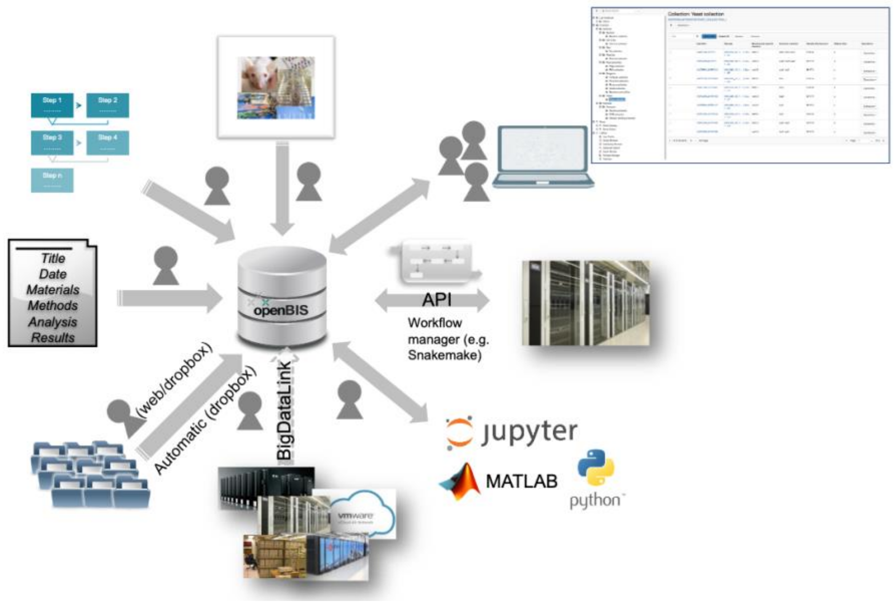


Figure 18. openRDM functionalities

As presented in Figure 18, openBIS is a server-client application: the remote server hosts the database and storage backends that users access from their local machines via a Web browser. It enables scientists to meet the increasingly demanding requirements of funding

[74] https://openbis.ch/

agencies, journals, and academic institutions to publish data according to the FAIR data principles – according to which data must be Findable, Accessible, Interoperable and Reusable[75]. The system is available in two versions: (i) a specific one for life sciences and (ii) a generic and customisable one for other scientific disciplines.

The openRDM service consists of the following service components:

- A *Preview* openBIS server installed and configured on cloud infrastructure, intended for end-users to learn the service, and eventually plan a deployment on-premises and/or on a private cloud.
- Consultancy and support for the deployment of openBIS on-premises and/or on a private cloud.
- User support, including generation of data models, import of data into openBIS and training in the use of openBIS as a data management platform.

The *Preview* instance is currently running on the EGI Cloud IaaS, with a fully automated deployment procedure based on Ansible and integrated with EGI Check-in. Access to this cloud-based *Preview* instance is provided for end-user testing and on-boarding. In addition, consultancy and support are provided for on-premises deployment of the openRDM platform. In the case of the cloud-based Preview instance, no data backup and retention are provided.

[75] https://www.go-fair.org/fair-principles/

6. Platforms

EGI-ACE platforms offer generic services that are widely used and not discipline-specific, covering the following functionality:

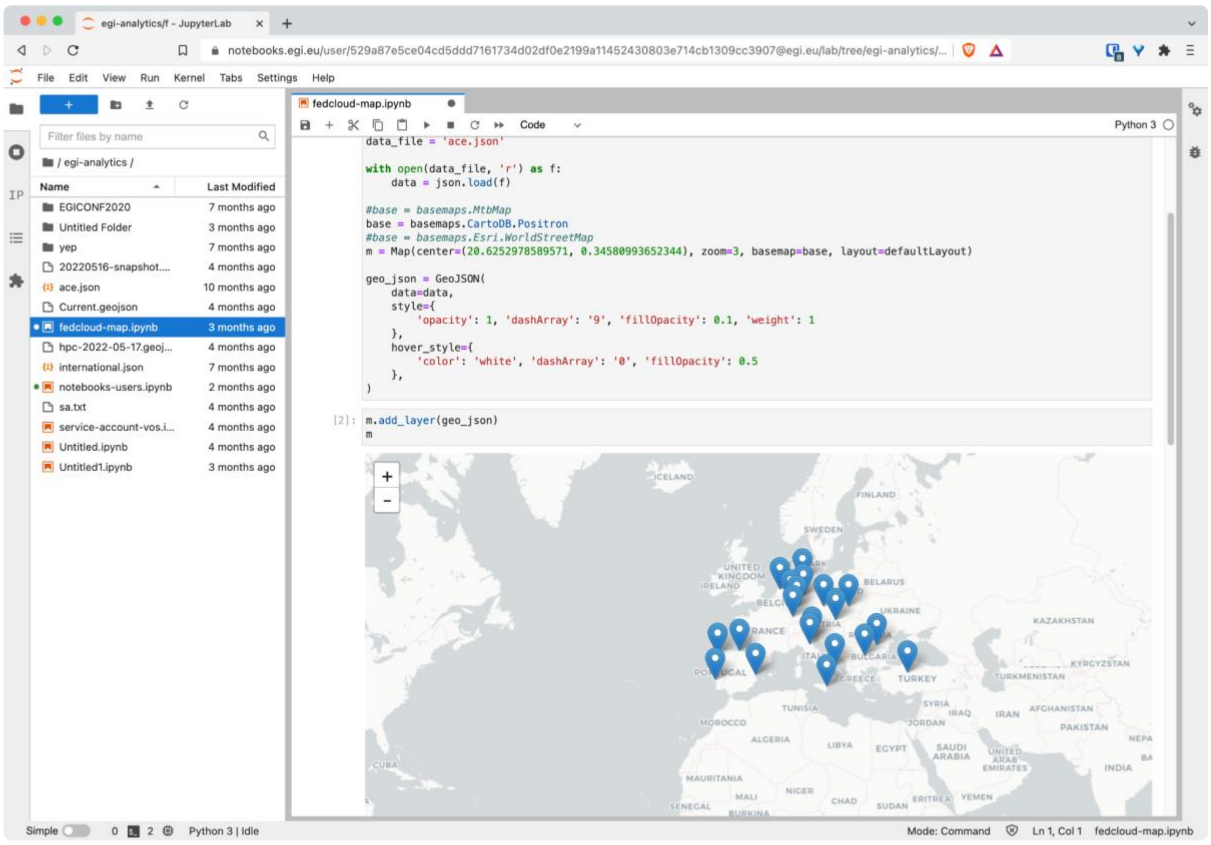
- **Interactive Computing**, with a Jupyter-based service that allows users to create live documents that run on the EGI infrastructure. This is discussed in Section 6.3.
- Scalable **deployment of big data tools and clusters**, supported by EC3, PaaS Orchestrator and DODAS, enable the creation of virtual infrastructures on top of IaaS resources. Each tool has its own levels of abstractions and features, and are discussed in Sections 6.2, 6.3 and 6.4.
- **AI/ML training,** with the DEEP training facility service that supports the *train-test-evaluation* cycle for prototyping AI models and applications.

As with the Federated Compute and Federated Data areas, these services are partially integrated with the Core Services described in Section 2. All services integrate with Federated Identity access, Monitoring, Configuration Database, and Helpdesk. Accounting (including the definition of new records) is in progress.

6.1. Notebooks

EGI Notebooks[76] is a Web-based interactive development environment based on the JupyterHub[77] technology that runs on a cloud provider of EGI-ACE (storage and compute infrastructures). This environment offers a flexible tool where users can create and share documents containing live code, equations, narrative text and rich media output (see Figure 19 for an example session). It is accessible to both individual users and members of a Virtual Organisation. Computing resources are provisioned from providers integrated with the EGI Federated Cloud. Basic storage capacity is equally provisioned from EGI cloud sites in the form of IaaS resources, but additional storage can be made available on a case-by-case basis from a variety of storage capacity providers, including: (i) The Onedata-based EGI DataHub, (ii) Object storage interfaces exposed by other providers members, and (iii) Specialised, area-specific dataset providers such as the Sentinels Collaborative Ground Segment for Earth Observation data.

[76] https://notebooks.egi.eu
[77] https://jupyter.org



Name	Last Modified
EGICONF2020	7 months ago
Untitled Folder	3 months ago
yep	7 months ago
20220516-snapshot....	4 months ago
ace.json	10 months ago
Current.geojson	4 months ago
fedcloud-map.ipynb	3 months ago
hpc-2022-05-17.geoj...	4 months ago
international.json	7 months ago
notebooks-users.ipynb	2 months ago
sa.txt	4 months ago
service-account-vos.i...	4 months ago
Untitled.ipynb	4 months ago
Untitled1.ipynb	3 months ago

```
data_file = 'ace.json'

with open(data_file, 'r') as f:
    data = json.load(f)

#base = basemaps.MtbMap
base = basemaps.CartoDB.Positron
#base = basemaps.Esri.WorldStreetMap
m = Map(center=(20.6252978589571, 0.34580993652344), zoom=3, basemap=base, layout=defaultLayout)

geo_json = GeoJSON(
    data=data,
    style={
        'opacity': 1, 'dashArray': '9', 'fillOpacity': 0.1, 'weight': 1
    },
    hover_style={
        'color': 'white', 'dashArray': '0', 'fillOpacity': 0.5
    },
)
```

```
[2]: m.add_layer(geo_json)
     m
```

Figure 19. EGI Notebooks interface showing a notebook execution with code and map visualisation.

Users are authorised to use EGI Notebooks through the EGI Check-in service described in Section 2.1. An essential set of Notebook images is provided universally to all incoming users, and additional images can be made available on request supporting various programming languages like Python, Julia, R, Octave or MATLAB. The EGI notebooks storage is backed-up and restored from a S3 service provided by CESNET. User-defined environments built on-the-fly is facilitated through the *Binder* service[78], which shares the underlying infrastructure of EGI Notebooks. Furthermore, Notebook's full integration with GitHub and Zenodo allows users to easily engage with the concept of Open Science. This relation is shown in the EGI Notebooks architecture depicted in Figure 20.

[78] https://binder.notebooks.egi.eu

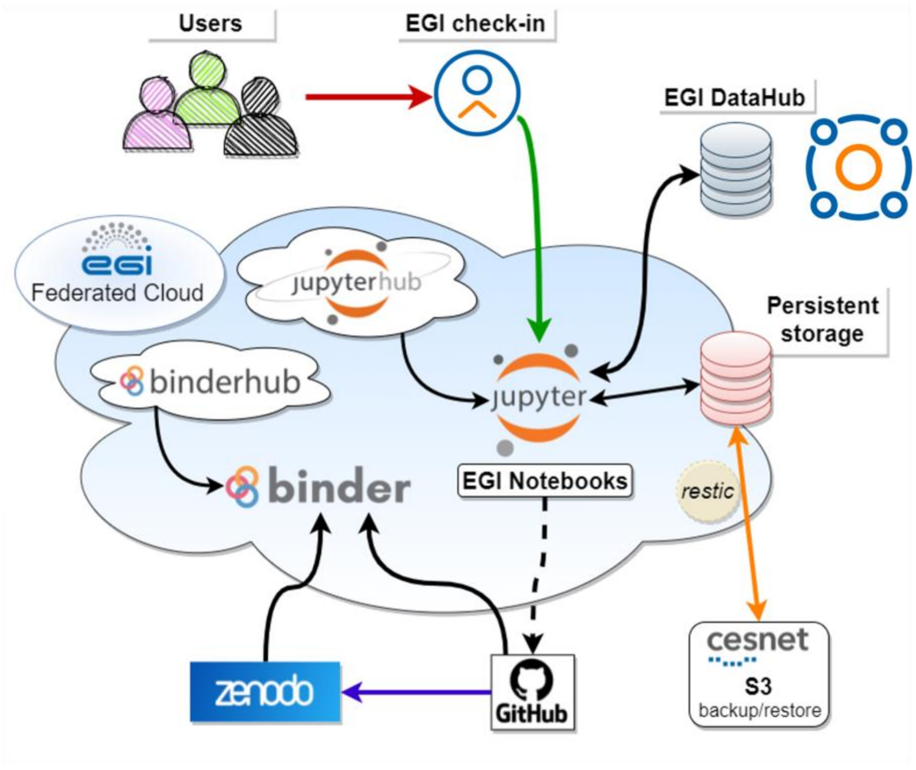


Figure 20. EGI Notebooks architecture

6.2. EC3

Elastic Cloud Computing Cluster (EC3)[79] is a tool for creating elastic virtual clusters on top of Infrastructure as a Service (IaaS) providers, either public (such as Amazon Web Services, Google Cloud or Microsoft Azure) or local (such as OpenNebula and OpenStack). It does so with a combination of Green computing, Cloud computing and HPC techniques.

By default, EC3 has the capacity to deploy different types of clusters: TORQUE [R24], SLURM [R25], SGE [R26], HTCondor [R27], Mesos [R28], OSCAR [R29], ECAS [R30], Nomad[80] and Kubernetes. These clusters are self-managed with CLUES [R31], which starts with a single-node cluster and dynamically deploys additional working nodes to satisfy the demanded load, measured in the number of jobs at the *Local Resource Management System* (LRMS). Working nodes are undeployed when they are idle, which introduces a cost-efficient approach for Cluster-based computing on top of an IaaS Cloud. EC3 cluster infrastructures are therefore automatically scaled out (adding new nodes on demand, up to a maximum size specified by the user) and scaled in (removing existing nodes whenever idle resources are detected, according to some predefined policies). This creates the illusion of a real cluster, without requiring an investment beyond actual usage.

[79] https://servproject.i3m.upv.es/ec3/
[80] https://www.nomadproject.io/

Figure 21 summarises the main EC3 architecture. The deployment of the virtual elastic cluster consists of two phases:

- The first one involves the creation of a VM in the Cloud to act as the cluster's front-end, which is done by a component called the *EC3 Launcher* using the Infrastructure Manager described in Section 4.2. Once the front-end and the elasticity manager (CLUES) have been deployed, the virtual cluster becomes fully autonomous, and each user will be able to submit jobs to the LRMS, either from the cluster front-end or from an external node that provides job submission capabilities.
- The second is the automatic management of the cluster size, depending on the workload and the specified policies. Users have the perception of a cluster with the maximum number of nodes specified. CLUES monitors the working nodes and intercepts job submissions before they reach the LRMS, allowing the system to dynamically manage the cluster size transparently to the LRMS and the user, scaling in and out according to demand. Just like in the deployment of the front-end, CLUES internally uses the Infrastructure Manager to support the provisioning of VMs that will be used as working nodes for the cluster. Once these nodes are available, they are automatically integrated in the cluster as new available nodes for the LRMS.

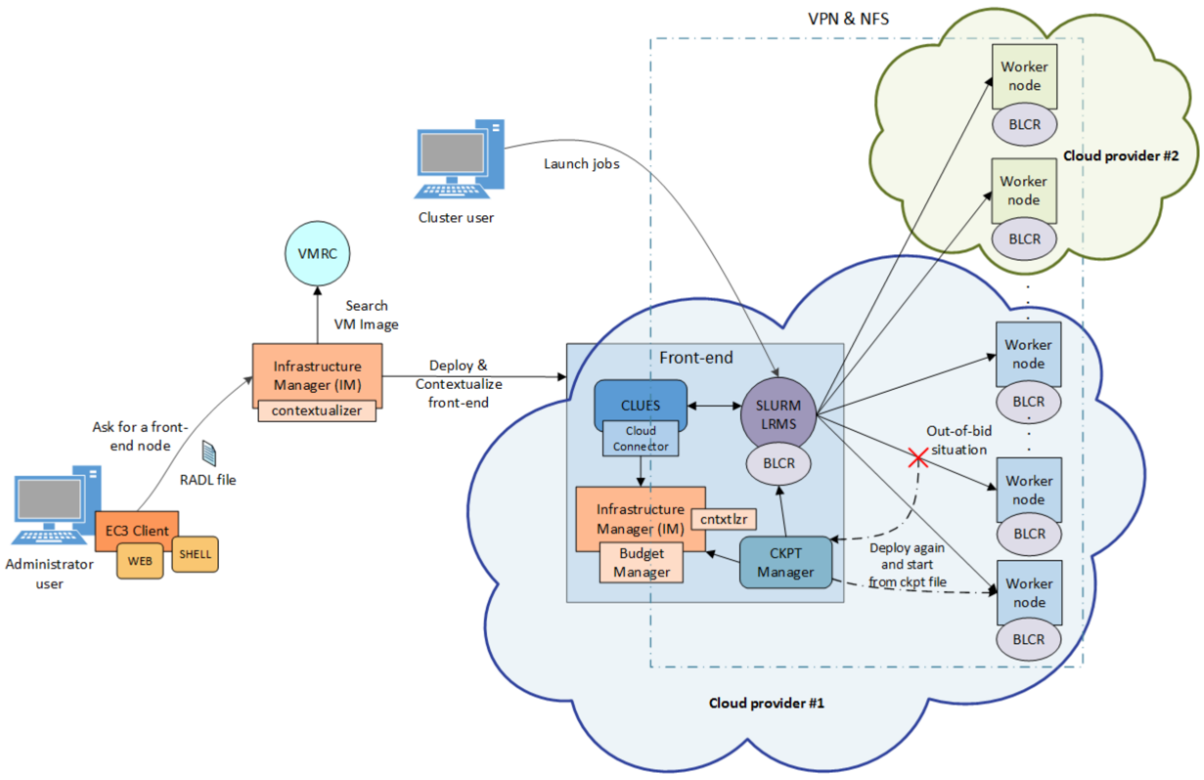


Figure 21. EC3 Architecture

EC3 supports three deployment models, as depicted in Figure 22:

- **Basic structure** (homogeneous cluster), which is composed of working nodes that have the same characteristics (in terms of hardware and software). This is the basic deployment model of EC3, where only one type of working nodes is used.
- **Heterogeneous cluster**, which allows working nodes with different characteristics (in terms of hardware and software). This is of particular interest when nodes with different configuration or hardware specifications must work together in the same cluster. Working nodes can be assigned to different queues according to their characteristics.
- **Cloud Bursting** (Hybrid clusters), which consists of launching nodes in two or more different Cloud providers, to overcome limitations on user quotas or saturated resources. When a limit is reached and no more nodes can be deployed inside the first Cloud Provider, EC3 launches new nodes in a second Cloud provider. Nodes deployed at different providers are automatically interconnected with VPN or SSH tunnelling techniques, and can also be different, so that heterogeneous clusters with cloud bursting capabilities can be deployed and automatically managed with EC3.

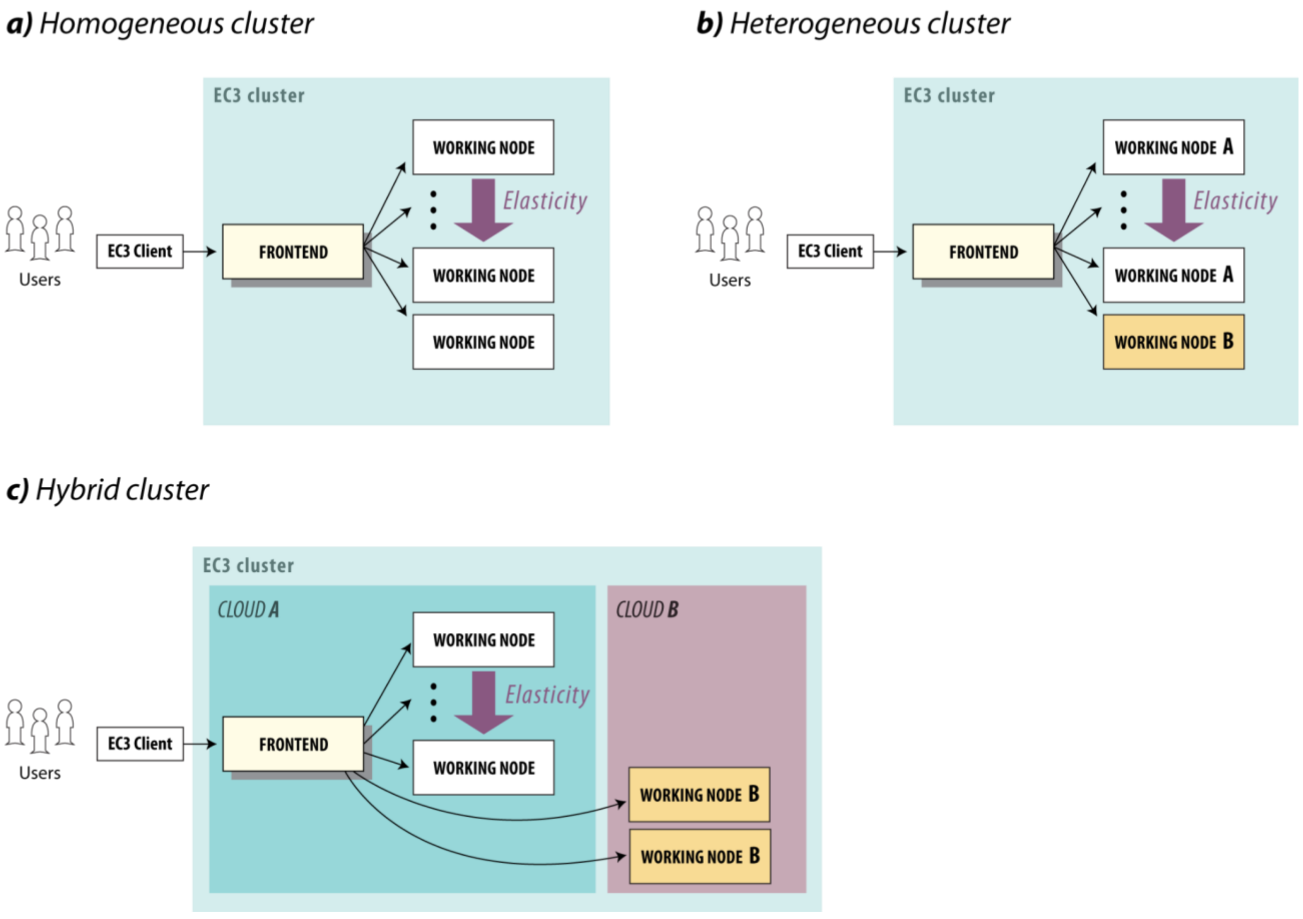


Figure 22. EC3 Deployment models

More information regarding EC3 can be found at EC3's GitHub repository[81].

6.3. PaaS Orchestrator

The PaaS Orchestrator[82] is the core component of *INDIGO PaaS* [R32], an abstraction and federation layer on top of heterogeneous distributed computing environments that orchestrates and coordinates:

- The provisioning of virtualised compute and storage resources on both private and public Cloud Management Frameworks, such as OpenStack, OpenNebula, AWS, *etc*.
- The deployment of containerised long-running services and batch jobs on Container Orchestration Platforms such as Apache Mesos [R28] and Kubernetes.
- The submission and monitoring of HPC jobs on HPC sites through a QCG [R33] gateway.
- The access to storage services to implement data placement, via the Rucio and DataHub Connector

These communications are illustrated in Figure 23.

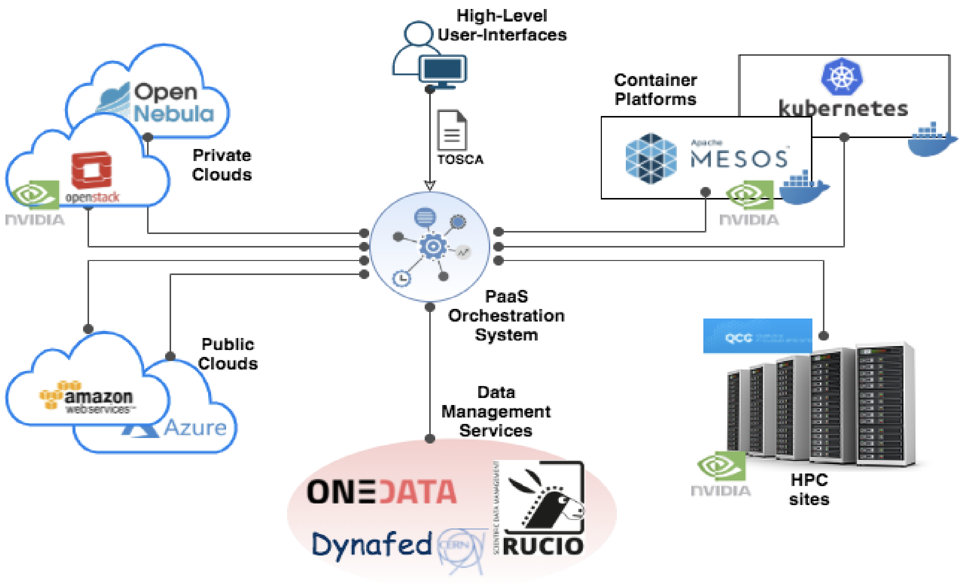


Figure 23. PaaS Orchestrator architecture

As depicted in Figure 24, the Orchestrator receives the deployment requests, expressed through templates written in TOSCA [R17], and coordinates the deployments on the best available cloud sites. To do so, the Orchestrator implements a complex workflow: it collects (i) Information on Service Level Agreements (SLAs) signed by providers and the user, (ii) Monitoring data about the availability of the compute and storage services, and (iii) The location of the data requested by the user (if any).

---

[81] https://github.com/grycap/ec3.

[82] See also EGI webinar https://indico.egi.eu/event/5720/

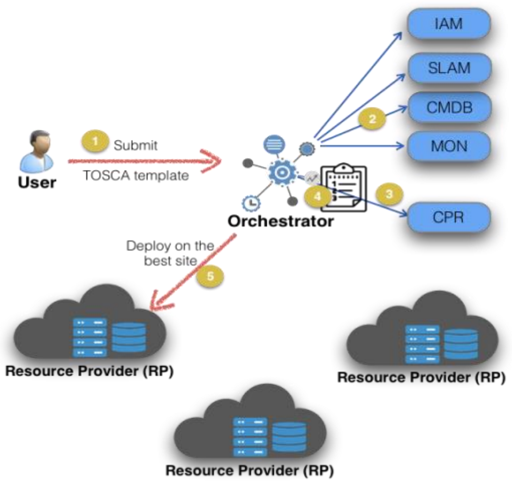


Figure 24. Orchestrator deployment workflow

Once the best site has been identified, the Orchestrator starts the actual deployment workflow through one of its provider plugins:

- **The Cloud/IaaS adapter,** which implements the interfaces with the relevant Cloud Management Frameworks through the Infrastructure Manager (described in Section 4.2).
- **The Mesos connectors**, which implement the interfaces that manage the interactions with the corresponding cluster framework, Marathon[83] (to manage long-running services) and Chronos[84] (to manage batch jobs).
- **The Kubernetes adapter**, which implements the interfaces for managing deployments on a Kubernetes cluster.
- **The HPC adapter**, which implements the interfaces for submitting jobs to HPC sites through a QCG Gateway.

Figure 25 displays the internal architecture of the PaaS Orchestrator, showing these available plugins in the lower layer. The top layer shows a REST API available for users to interact with the platform. A relational database (RDBMS in the figure) is used to store data about the state of the service.

[83] https://mesosphere.github.io/marathon/
[84] https://mesos.github.io/chronos/

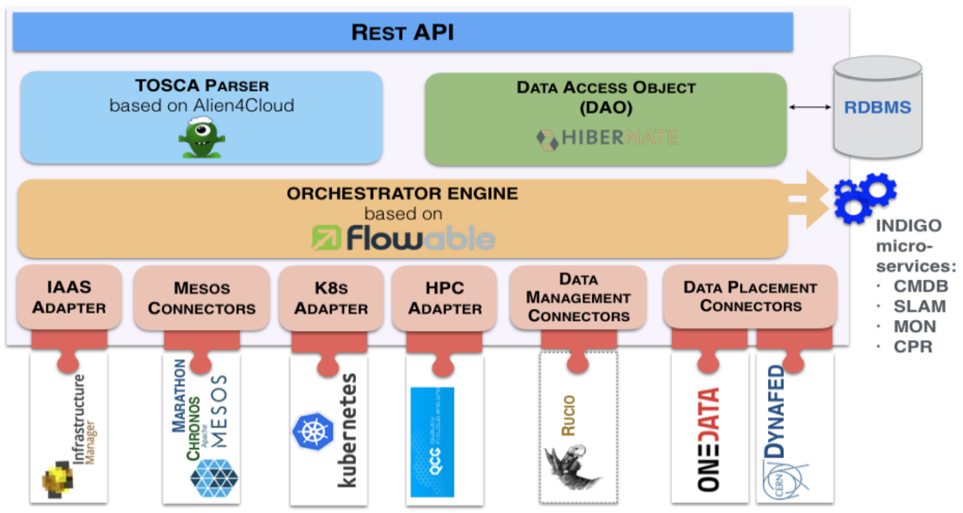


Figure 25. PaaS Orchestrator internal architecture

The PaaS Orchestration system provides the following key features:

- Support for deployments that need specialised hardware resources, such as GPUs and Infiniband.
- Support for hybrid deployments and network orchestration.
- Support for isolated deployments, *i.e.,* hosted on private networks (with no inbound connection) that can be reached by users via VPN.
- Automatic retry in case of deployment failure or timeout.
- Integration with *Hashicorp Vault*[85] to manage public cloud credentials.
- Support for multiple OIDC Identity Providers.
- Multi-tenancy support.
- RESTful API endpoints for handling deployments.
- Command-line and Web interfaces.

6.4. DODAS

Dynamic On-Demand Analysis Software (DODAS)[86] [R34] is a service for running user analysis code based on batch jobs, via the command-line or a Jupyter-based interface. DODAS is open-source, can be deployed in production and demanding environments is highly customisable, and has a modular design that allows users to configure the service to efficiently address their specific use cases. These modules are called “blocks”. Existing features of blocks include:

- Deployment and use of an HTCondor cluster over heterogeneous Cloud resources.
- Combination of Jupyter and HTCondor.
- Use of the Jupyter interface for analysis.
- Deployment of Big Data Pre-Post processing facilities.

---

[85] https://www.vaultproject.io/
[86] See EGI webinar https://indico.egi.eu/event/5695/

To do so, resource and software provisioning, configuration and management is transparently conducted for the user. Other DODAS blocks support data management through caches (to optimise the processing of remote data) and Posix or S3-compatible storage systems.

DODAS operates on Docker containers orchestrated at the IaaS layer by Kubernetes. When it is used for batch system processing, it automatically deploys and manages both the HTCondor central services (treating them as Long Running Services) and the worker nodes, which can be made automatically and dynamically scalable.

DODAS itself is run on container technology, and it can be easily customised and adapted to specific needs. Since DODAS interfaces natively with Kubernetes, its building blocks may also be composed via Kubeapps[87], a Web-based dashboard for Kubernetes.

A DODAS key element is the integration of a flexible, standard-based, and federated Authentication and Authorization system, based on *INDIGO-IAM,* so it uses OIDC token-based mechanisms at all levels of the computing and data stacks. From the user's perspective, this translates into a greatly simplified single sign-on experience. EGI Check-in is integrated as an external authentication mechanism of IAM, and IAM is used as an AAI harmonisation layer, so that Check-in grants access to EGI cloud resources, while the IAM harmonised identity is used to manage DODAS domain-specific use cases.

Figure 26 highlights the high modularity of the technical implementation of DODAS, which allows a strong integration with EGI-ACE services: Identity and Access Management (INDIGO-IAM), PaaS Orchestrator and Infrastructure Manager.

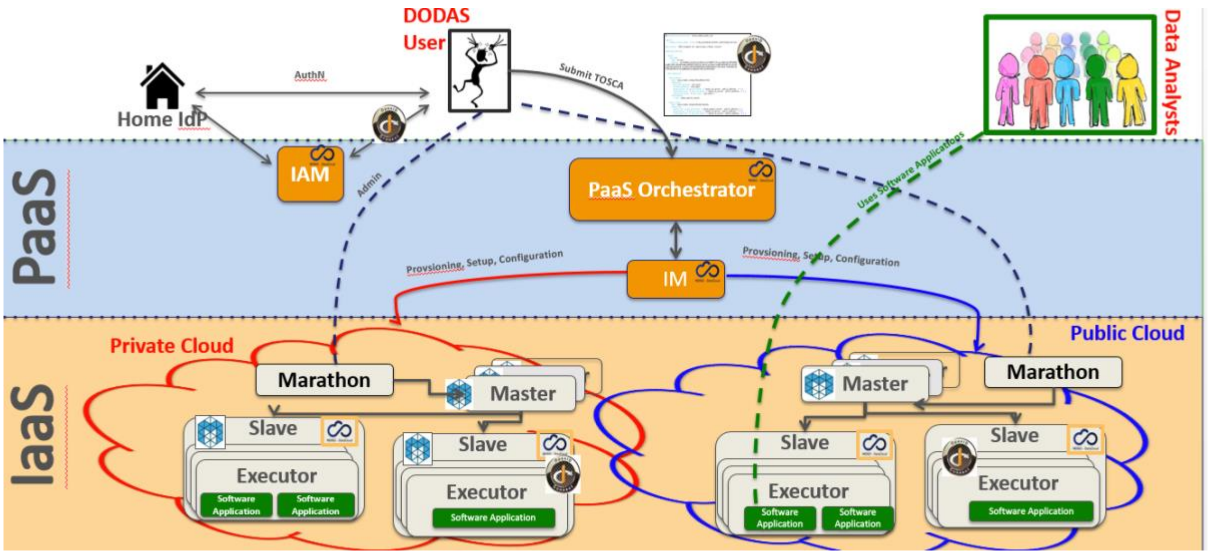


Figure 26. DODAS architecture

[87] https://kubeapps.dev/

6.5. DEEP training facility

The DEEP platform was developed in the DEEP-Hybrid-DataCloud project[88] and enhanced with services developed in the EOSC-Synergy project[89] and with continuous contributions from DEEP partners. It offers a complete framework for users, practitioners, and developers of Artificial Intelligence (AI) with various levels of expertise. The framework allows Machine Learning and Deep Learning (ML/DL) applications to be transparently trained, shared, and served, both locally and on hybrid cloud systems, in the context of EOSC. The provided set of tools and services uses a DevOps approach [R35] to cover the whole ML/DL development cycle, ranging from model creation, data processing, training, validation, and testing of models serving as a service (through a serverless architecture), sharing and publication. Developers of the services can focus on domain-specific challenges, while the DEEP platform takes care of additional support tasks (AAI, resource management, marketplace, CI/CD software quality assurance, *etc*).

The DEEP training facility[90] allows AI model prototypes and applications to undergo through the train-test-evaluation cycle of the ML lifecycle, performing this phase on production-grade resources required for each of the training steps (see Figure 27). This facility therefore allows access to the underlying Cloud, HTC and HPC resources exploiting accelerators, transparently to the user through a user-friendly dashboard.

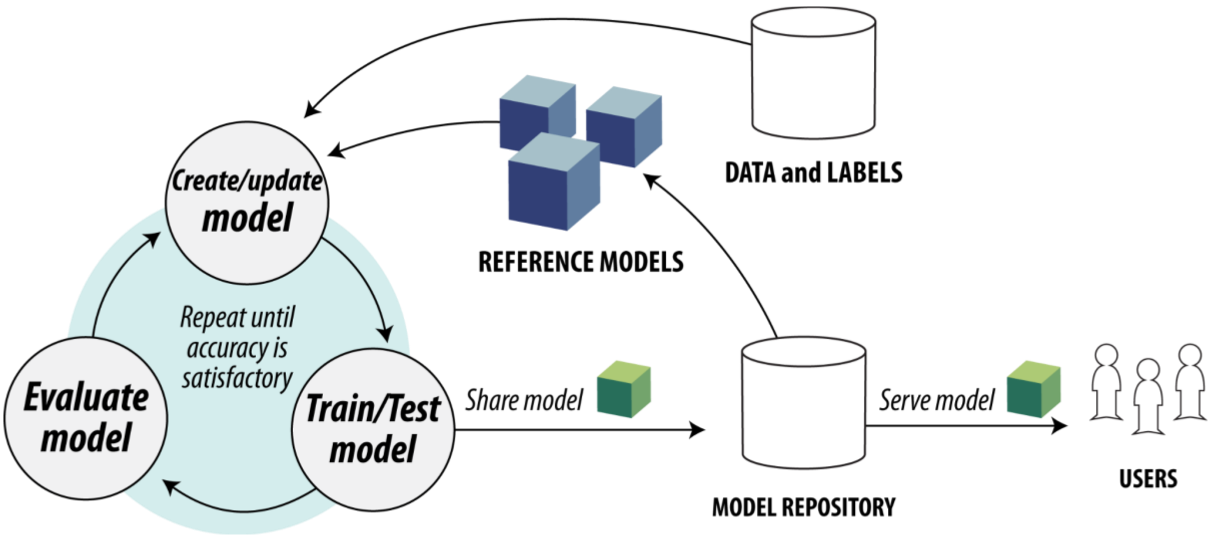
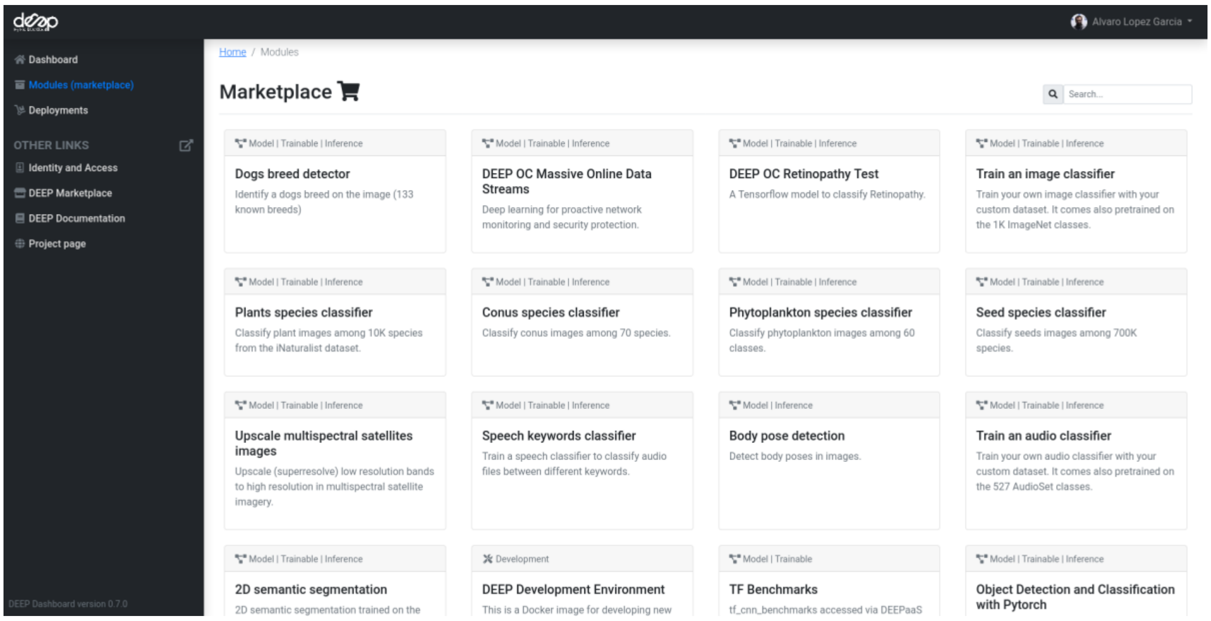


Figure 27. DEEP training model workflow

Once a model has been initially built, the dashboard allows users to access computing resources and train it. The dashboard hides the complexity of making a deployment in the DEEP framework, letting users to easily interact with resources through a simple GUI.

88 https://deep-hybrid-datacloud.eu/
89 https://www.eosc-synergy.eu/
90 https://train.deep-hybrid-datacloud.eu/

Similarly, the dashboard allows users to interact with modules hosted at the *DEEP Open Catalogue*[91], as well as deploy external Docker images hosted in Docker Hub[92]. For all deployments, users can select the desired hardware (CPU or GPU), the amount of memory and other relevant parameters. Another useful feature is the ability to store the history of previous training sessions, allowing the training status to be monitored directly from the Dashboard, while tracking the results of experiments.



Model | Trainable | Inference
Dogs breed detector
Identify a dogs breed on the image (133 known breeds)

Model | Trainable | Inference
DEEP OC Massive Online Data Streams
Deep learning for proactive network monitoring and security protection.

Model | Trainable | Inference
DEEP OC Retinopathy Test
A Tensorflow model to classify Retinopathy.

Model | Trainable | Inference
Train an image classifier
Train your own image classifier with your custom dataset. It comes also pretrained on the 1K ImageNet classes.

Model | Trainable | Inference
Plants species classifier
Classify plant images among 10K species from the iNaturalist dataset.

Model | Trainable | Inference
Conus species classifier
Classify conus images among 70 species.

Model | Trainable | Inference
Phytoplankton species classifier
Classify phytoplankton images among 60 classes.

Model | Trainable | Inference
Seed species classifier
Classify seeds images among 700K species.

Model | Trainable | Inference
Upscale multispectral satellites images
Upscale (superresolve) low resolution bands to high resolution in multispectral satellite imagery.

Model | Trainable | Inference
Speech keywords classifier
Train a speech classifier to classify audio files between different keywords.

Model | Inference
Body pose detection
Detect body poses in images.

Model | Trainable | Inference
Train an audio classifier
Train your own audio classifier with your custom dataset. It comes also pretrained on the 527 AudioSet classes.

Model | Trainable | Inference
2D semantic segmentation
2D semantic segmentation trained on the

Development
DEEP Development Environment
This is a Docker image for developing new

Model | Trainable
TF Benchmarks
tf_cnn_benchmarks accessed via DEEPaaS

Model | Trainable | Inference
Object Detection and Classification with Pytorch

Figure 28. DEEP marketplace

The training facility relies on modules published on the DEEP Marketplace, shown in Figure 28, so it requires user applications to be installed in a container image and integrated with the DEEP API[93] to expose its functionality. The training facility uses the INDIGO PaaS Orchestrator described in Section 6.3 to integrate with infrastructure systems (*e.g.,* OpenStack, Kubernetes, Mesos or HPC systems).

[91] https://marketplace.deep-hybrid-datacloud.eu/
[92] A repository for container images. https://hub.docker.com
[93] https://docs.deep-hybrid-datacloud.eu/projects/deepaas/en/stable/

7. Conclusions

EGI-ACE delivers the EOSC Compute Platform following a layered architecture where Cloud, HTC and HPC resource providers are federated to enable processing and analysis for all kinds of research needs.

A set of Core Services has been described in Section 2, which provides federated operations (such as common configuration management, authentication and authorisation mechanisms, monitoring, accounting, and support) for those providers that participate as full members of the EGI Federation, becoming an integral part of the services delivered by the Federation to the EOSC Marketplace. Providers that are not part of the Federation can still contribute to the EGI-ACE architecture, supporting the different services provided by the Compute Federation, Data Federation, Platforms and Thematic Service layers, although their use will not be managed via EGI's SLA/OLA framework. Providers not fully integrated in the EGI Federation can rely on common AAI, shared applications, and shared data. Providers can choose the level of integration that better suits their needs and the communities they serve.

The offer of compute services of the EOSC Compute Platform has been described in Section 3, and covers different types of applications and use cases: from Virtual Machines with full user-control to a fully managed distributed platform to run jobs:

Table 3 – Description of EOSC Compute Services

Type	**Description**	**Use Case**
Cloud IaaS	Provides access to VM-based computing with associated storage. It delivers a customisable platform where users have complete control over the software and the supporting compute capacity.	The flexibility of the computing platform enables a variety of workloads: user gateways or portals, interactive computing platforms and almost any kind of data- and/or compute-intensive workloads.
High Throughput Compute	Provides access to large, shared Grid Computing systems for running computational jobs at scale.	Analysis of large datasets in an "embarrassingly parallel" fashion, *i.e.* by splitting the data into small pieces, and executing thousands, or even more independent computing tasks simultaneously, each processing one piece of data.
High Performance Compute	Supports highly optimised applications that need massively parallel computing with low latency and a high bandwidth interconnection network.	Complex computational problems using tightly coupled parallel processing: simulations, analysis of large datasets or AI/ML workloads. Typically, these applications rely on MPI[94] for supporting interprocess communication.

[94] Message Passing Interface (MPI) https://www.mpi-forum.org/

The Federated Compute and the Federated Data services have been discussed in Sections 4 and 5 respectively and support the execution of research workloads by delivering agnostic ways to run applications on the heterogeneous set of providers. These support both exploiting data locality, by moving computing near data, and seamless access to remote data, with replication and caching whenever necessary.

A set of platform-level services has been described in Section 6, which provides further generic tools to exploit the compute and storage resources of EGI-ACE. These include Interactive Notebooks, Scalable deployment of big data tools and AI/ML training.

All these layers support the discipline-specific thematic services that bring additional analytics and data for specific communities.

The compute services of EGI-ACE described in this document contribute to the adoption of the European Open Science Cloud (EOSC) by European researchers, innovators, companies, and citizens, serving as a valuable introduction of EOSC capabilities and the potential advantages that it can bring to advanced research. Researchers can benefit from a federated and open multi-disciplinary environment that facilitates the publication, search and reuse of data, tools and services, with the ultimate goal of improving research, innovation and education.

8. References

No	Description/Link
R1	**EGI-ACE D2.3 Technical specifications for compute common services** https://documents.egi.eu/document/3816
R2	**PROC09 Resource Centre Registration and Certification** https://confluence.egi.eu/display/EGIPP/PROC09+Resource+Centre+Registration+and+Certification
R3	**REFEDS Research and Scholarship (R&S)** https://refeds.org/category/research-and-scholarship
R4	**GÉANT Data Protection Code of Conduct** https://www.geant.org/uri/Pages/dataprotection-code-of-conduct.aspx
R5	**A Security Incident Response Trust Framework for Federated Identity (Sirtfi) Version 2** https://refeds.org/wp-content/uploads/2022/08/Sirtfi-v2.pdf
R6	**Security Assertion Markup Language 2.0 (SAML 2.0)** https://wiki.oasis-open.org/security/FrontPage
R7	**OpenID Connect (OIDC)** https://openid.net/connect/
R8	**ISO/IEC 9594-8:2020 Open systems interconnection — Part 8: The Directory: Public-key and attribute certificate frameworks** https://www.iso.org/standard/80325.html
R9	**VOMS, an Authorization System for Virtual Organizations.** R. Alfieri et al. In Grid Computing. AxGrids 2003. Lecture Notes in Computer Science, vol 2970. Springe DOI: 10.1007/978-3-540-24689-3_5

R10	**GLUE Specification v. 2.0.** Edited by S.Andreozzi, http://www.ogf.org/documents/GFD.147.pdf
R11	**ARGO Guidelines for monitoring probes** https://argoeu.github.io/argo-monitoring/docs/monitoring/guidelines
R12	**SSH Mastery: OpenSSH, PuTTY, Tunnels and Keys**. M. W. Lucas. IT Mastery, 2018. ISBN: 978-1642350029
R13	**Serious Cryptography. A Practical Introduction to Modern Encryption.** JP Aumasson. No Starch Press, 2017. ISBN:978-1-59327-826-7
R14	**EGI ACE D7.3 HPC integration handbook** https://documents.egi.eu/document/3804 (last accessed: Sept 2022)
R15	**The DIRAC interware: current, upcoming and planned capabilities and technologies.** F. Stagni, A. Tsaregorodtsev, A. Sailer, C. Haen. EPJ Web Conf. 245 03035 (2020) DOI: 10.1051/epjconf/202024503035
R16	**Dynamic Management of Virtual Infrastructures**. M. Caballer, I. Blanquer, G. Moltó, C. de Alfonso, J. Grid Comput., 13, 53–70, (2015). DOI: 10.1007/s10723-014-9296-5
R17	**TOSCA Simple Profile in YAML Version 1.0**. Edited by Derek Palma, Matt Rutkowski, and Thomas Spatzier. 21 December 2016. OASIS Standard. http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/os/TOSCA-Simple-Profile-YAML-v1.0-os.html
R18	**ISO 26324:2022 Digital object identifier system** https://www.iso.org/standard/81599.html
R19	**Persistent Identifiers.** Davidson, J. (2006). DCC Briefing Papers: Introduction to Curation. Edinburgh: Digital Curation Centre. Handle: 1842/3368.
R20	**ISO/IEC 17826:2022. Cloud Data Management Interface (CDMI) Version 2.0.0**. https://www.iso.org/standard/83451.html

R21	**Resource Description Framework (RDF) 1.1 Primer**. Schreiber G., Raimond Y. 2014. W3C Recommendation. http://www.w3.org/TR/rdf11-primer/
R22	**The Open Archives Initiative Protocol for Metadata Harvesting.** Edited by C. Lagoze and H. van de Sompel. 2015 https://www.openarchives.org/OAI/openarchivesprotocol.html
R23	**OpenAPI Specification v3.1.0.** Edited by D. Miller, J. Whitlock, M. Gardiner, M. Ralphson, R. Ratovsky, U. Sarid. The Linux Foundation 2021. https://spec.openapis.org/oas/v3.1.0
R24	**TORQUE resource manager.** Garrick Staples. In Proceedings of the 2006 ACM/IEEE conference on Supercomputing (SC '06). Association for Computing Machinery, New York, NY, USA, 8–es. 2016 DOI: 10.1145/1188455.1188464
R25	**SLURM: Simple Linux Utility for Resource Management.** A.B. Yoo, M.A. Jette, M.A., M. Grondona. Job Scheduling Strategies for Parallel Processing. JSSPP 2003. Lecture Notes in Computer Science, vol 2862. Springer. DOI: 10.1007/10968987_3
R26	**Sun Grid Engine: Towards Creating a Compute Power Grid.** W. Gentzsch. IEEE International Symposium on Cluster Computing and the Grid, Brisbane, Australia, 2001 pp. 35. DOI: 10.1109/CCGRID.2001.923173
R27	**Distributed Computing in Practice: The Condor Experience.** D. Thain, T. Tannenbaum and M. Livny, Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pages 323-356, February-April, 2005. DOI: 10.1002/cpe.938
R28	**Apache Mesos Cookbook** D. Blomquist, T. Janiszewski. Packt Publishing Ltd, 2017. ISBN: 978-178588-4627
R29	**On-Premises Serverless Computing for Event-Driven Data Processing Applications.** A. Pérez, S. Risco, D. M. Naranjo, M. Caballer and G. Moltó, IEEE 12th International Conference on Cloud Computing (CLOUD), 2019, pp. 414-421, DOI: 10.1109/CLOUD.2019.00073
R30	**ENES Climate Analytics Service (ECAS).** S. Bendoukha, T. Weigel, S. Fiore, A. D'Anca, 20th EGU General Assembly, EGU2018, Proceedings from the conference held 4-13 April, 2018 in Vienna, Austria, p.12549 Bibcode: 2018EGUGA..2012549B
R31	**Multi-elastic Datacenters: Auto-scaled Virtual Clusters on Energy-Aware Physical Infrastructures.** C. de Alfonso, M. Caballer, A. Calatrava, et al. J Grid Computing 17, 191–204 (2019). DOI: 10.1007/s10723-018-9449-z

R32	**INDIGO-DataCloud: a Platform to Facilitate Seamless Access to E-Infrastructures.** D. Salomoni, I. Campos, L. Gaido, et al. J Grid Computing 16, 381–408 (2018) DOI: 10.1007/s10723-018-9453-3
R33	**Development of Science Gateways Using QCG -- Lessons Learned from the Deployment on Large Scale Distributed and HPC Infrastructures.** T. Piontek, et al. J. Grid Comput. 14, 4 (December 2016), 559–573. DOI: 10.1007/s10723-016-9384-9
R34	**DODAS: How to effectively exploit heterogeneous clouds for scientific computations.** D. Spiga *et al*. International Symposium on Grids and Clouds 2018 in conjunction with Frontiers in Computational Drug Discovery (ISGC 2018 & FCDD) - Virtual Research Environment (VRE) DOI: 10.22323/1.327.0024.
R35	**The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.** G. Kim, P. Debois, J. Willis, J. Humble. IT Revolution Press 2016. ISBN: 978-1942788003