D2.3 Technical specifications for compute common services

Lead partner:	EGI.eu
Version:	1
Status:	Under EC review
Dissemination Level:	PUBLIC
Keywords:	Technical Architecture, Compute Services
Document Link:	https://documents.egi.eu/document/3816

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. This document describes the different layers of the EGI-ACE technical architecture and details each of the services that compose those layers, covering the central services that enable the federation, resource providers delivering access to the actual computing and storage infrastructure, and higher level Platform and Software as a Service layers that provide compute and data orchestration alongside tools to facilitate the execution of workloads in the distributed infrastructure.

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

go.egi.eu/egi-ace

COPYRIGHT NOTICE

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

DELIVERY SLIP

Date	*Name*	*Partner/Activity*
From:	Enol Fernandez	EGI.eu/WP2
Moderated by:	Malgorzata Krakowian	EGI.eu/WP1
Reviewed by:	Jerry Horgan Andrea Cristofori	Walton Institute (IE) EGI.eu
Approved by:	SDS	

DOCUMENT LOG

Issue	*Date*	*Comment*	*Author*
v.0.1	2021-09-27	Initial ToC proposal	Enol Fernández (EGI.eu) Gergely Sipos (EGI.eu)
v0.2	2021-10-27	First complete version	Marica Antonacci (INFN), Valeria Ardizzone (EGI.eu), Priyasma Bhoumik (ETH Zurich), Miguel Caballer (UPV), Amanda Calatrava (UPV), Ian Collier (STFC), Rose Cooper (STFC), Enol Fernández (EGI.eu), William Karageorgos (IASA), Bartosz Kryza (CYFRONET), Nicolas Liampotis (GRNET), Slávek Licehammer (CESNET), Alvaro López (CSIC), Andrea Manzi (EGI.eu), Gino Marchetti (CNRS), Germán Moltó (UPV), Timothy Noble (STFC), Viet Tran (IISAS), Petr Pospíšil (CESNET),

			Daniele Spiga (INFN), Zdeněk Šustr (CESNET), Richard Wartenburger (ETH Zurich)
V1	2021-11-05	Tackled reviewer comments	Enol Fernández (EGI.eu), Gergely Sipos (EGI.eu)

TERMINOLOGY

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

Contents

Executive summary

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 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 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 deliverable describes the different layers of the EGI-ACE technical architecture and details each of the services that compose those layers.

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.

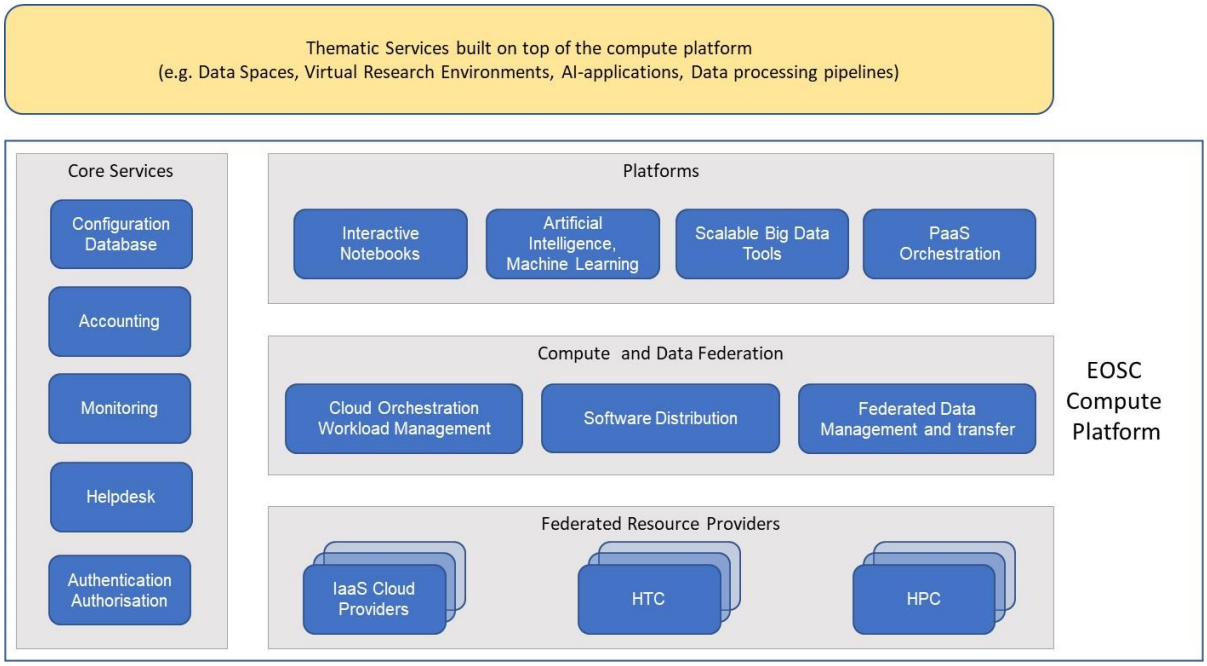

Figure 1 EOSC Compute Platform functional block diagram

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 included in this layer:

- **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** (High Throughput Compute) provide access to large, shared computing systems for running computational jobs at scale. These allow researchers to analyse 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. HTC means the execution and management of many independent tasks over longer times.
- **HPC** (High Performance Compute) (to be available in 2022) supports very optimised application of machines that have lot of interconnected processing unit. with many dependent tasks that need large amounts of parallel computing along with a low latency and high bandwidth interconnection network.

The Service Management Tools pillar delivers the functionality for services of all other areas to be integrated in the Federation. They support the operation of the EOSC Compute platform and integrate and interoperate with the EOSC Core that is run and further developed in the EOSC Future project. EGI's **Authentication and Authorisation** service, called Check-in, is a key component of the architecture that enables using a common identity across all the layers and services of the EOSC Compute platform. **Configuration Database**, **Monitoring**, **Accounting**, and **Helpdesk** services are also included in this area alongside with other non-technical services and coordination activities like Operations Management, and Security and Incident Response.

The Compute Federation services orchestrate the execution of user workloads in the Federated Resource Providers. They exploit data locality by moving computing near data and facilitate application portability with the support of a diverse range of computing platforms (Cloud IaaS, HTC, HPC) and the interaction with software distribution tools (as VM images, container images or binaries directly). There are three services in this layer of 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 Federated Data services support exposing discoverable datasets and staging data into/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 can be transferred among the EGI-ACE providers, and between EGI-ACE provides and external data repositories. The Federated Data Management function uses the the **Data Transfer** service to perform the transfers.

A Platforms service area provides generic added-value services for scientific communities to build thematic services for end-users (typically for researchers). The platforms rely on the existing Compute Federation and Data Federation services to access the Federated Resource Providers and deliver **Interactive Notebooks**, **PaaS Orchestration** to facilitate the deployment of complex applications, and **Artificial Intelligence and Machine Learning** and **Scalable Big Data Tools** that can be reused in several research disciplines.

Thematic Services are built by combining services of all these areas to provide capabilities for simulation, machine learning and data analytics that are tailored to the needs of a specific research domain. Thematic Services build on the overall EGI-ACE service stack by external communities and projects. Each thematic service has a custom combination of applications/tools and data specific for the given disciplines or scientific questions they address. EGI-ACE also includes such thematic services in WP5 (Called Data Spaces). Thematic Services (incl. EGI-ACE Data spaces) share the EOSC Compute Platform as a common architecture, but the rest of their setup is custom therefore they are covered in this deliverable.

The rest of the document covers the different functional areas of the EGI-ACE technical architecture: Service Management Tools, Federated Resource Providers, Federated Compute, Federated Data and finally Platforms.

2. Service Management Tools

This section describes the components of the EGI-ACE technical architecture that enable the operation of the federation. The Service Management Tools form a vertical pillar that integrates with all layers and provides a uniform access and management of the services. Some of these tools play a similar operational role in the EOSC Core (the Helpdesk, same as integrated with the equivalent counterparts in EOSC Core as they become available, thus making it possible to integrate EGI-ACE with EOSC.

2.1. Check-in: Authentication and Authorisation

Check-in[1] is the authentication, authorisation and user management service for the EGI infrastructure. It enables users to access EGI, and third-party services (web and non-web based), using existing credentials managed by the Identity Providers (IdPs) of their home organisations. To enable access through academic identity providers, Check-in has been registered in eduGAIN[2] as a Service Provider (SP). Through eduGAIN, EGI and third-party services connected to Check-in can become available to more than 4500 Universities and Institutes from 71 National Identity Federations with little or no administrative involvement. Compliance with the REFEDS Research and Scholarship (R&S)[3] entity category, the GÉANT Data Protection Code of Conduct[4] and the Sirtfi[5] security framework ensures sufficient attribute release, as well as operational security, incident response, and traceability. Complementary to this, users without an account on an academic Identity Provider are still able to use ORCID[6], social media or other external authentication providers for accessing EGI and third-party services that do not require substantial level of assurance. The 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 Virtual Organisation (VO)/group membership rights. The adoption of standards and open technologies by Check-in, including SAML 2.0, OpenID Connect and X.509v3, has facilitated interoperability and integration with the existing Authentication & Authorisation Infrastructures (AAIs) of other e-Infrastructures and research communities, such as B2ACCESS and eduTEAMS.

[1] https://www.egi.eu/services/check-in/, see also webinar https://indico.egi.eu/event/5494/
[2] https://edugain.org/
[3] https://refeds.org/category/research-and-scholarship
[4] https://www.geant.org/uri/Pages/dataprotection-code-of-conduct.aspx
[5] https://refeds.org/sirtfi
[6] https://orcid.org/

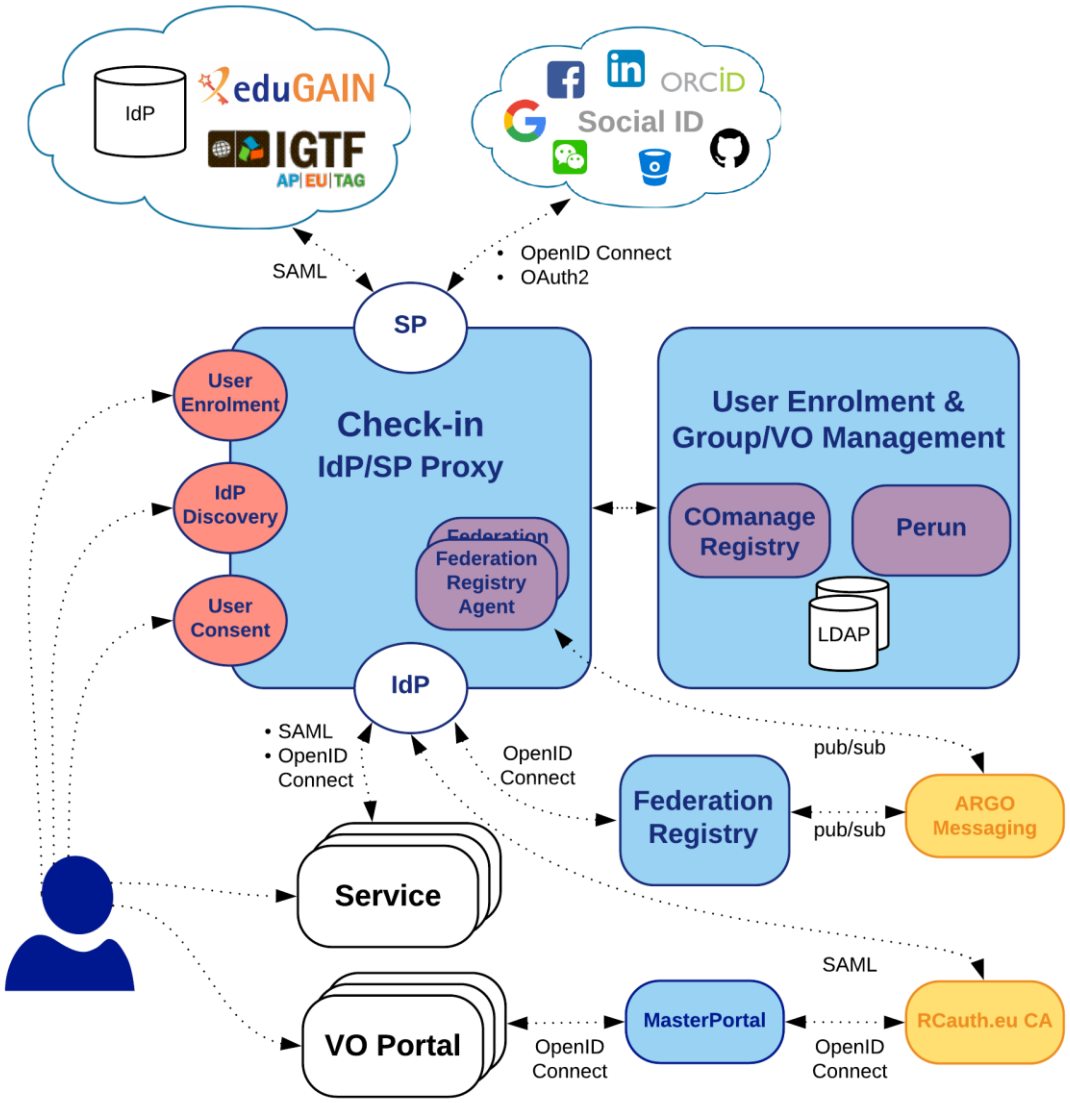

Figure 2 Check-in architecture

Figure 2 shows the Check-in top-level architecture diagram and 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, FedCloud providers etc). Through the Check-in IdP/SP proxy, users are able to sign in with the credentials provided by their Identity Provider. To achieve this, the proxy supports different authentication and authorisation standards, such as SAML 2.0, OpenID Connect (OIDC) 1.0 and OAuth 2.0. The proxy also provides a central Discovery Service (Where Are You From – WAYF) for users to select their preferred Identity Provider.

The **User Enrolment and Group/VO Management** component, which is based on the **COmanage Registry**[7] tool, supports the management of the full life cycle of user accounts in Check-in. This includes the initial user registration, the acceptance of the terms of use of the infrastructure, account linking, group and VO management, delegation of administration of VOs/Groups to authorised users and the configuration of custom enrolment flows for VOs/Groups via an intuitive web interface.

Check-in also provides identity management capabilities through the **Perun**[8] 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 synchronize both users and groups from an external system, which can be consequently combined with aforementioned manual management in Perun itself. Perun also manages user profiles including an identity linking service, which can deal with various types of identities including federated identities, X.509 certificates or local username and passwords. Furthermore, Perun supports registering facilities, which represent anything from single machines, clusters, storage elements to even software licenses in various infrastructure sizes. The facility manager can provide a resource to any VO manager, by which he/she defines roles, access rights and rules controlling how the VO can utilise the facility. The VO manager then decides which users within the VO will be allowed to use the resource. 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 or 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 standardized attributes or claims.

The **MasterPortal** acts as a caching intermediary service between end-services and the RCauth Online CA[9]. 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: it is necessary for a scalable trust model based on a single online CA, and secondly, it hides all the complexity of X.509 certificates and keys handling for the end-services. In this pure webflow, the MasterPortal is seen by the end-users only as a 'redirect' between the end-service and the RCauth.eu online CA. 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 on 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 OpenID Connect and SAML based services to Check-in. The web interface of the Federation Registry covers the whole service lifecycle, including the initial registration, reconfiguration, and deregistration. The service configurations are deployed by sending configuration messages to the **Federation Registry Deployment Agents** running on the

[7] https://www.incommon.org/software/comanage/
[8] https://perun-aai.org/
[9] https://rcauth.eu/

IdP/SP Proxy component. These configuration messages are exchanged asynchronously through the ARGO Messaging[10] service following the Google pub/sub[11] protocol.

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

- Research communities can leverage Check-in for managing their users and their respective roles and other authorisation-related information using either COmanage Registry or Perun.
- Users authenticate using a community-specific identity provider, for example B2ACCESS, eduTEAMS, or any other Community Authentication & Authorisation service which is compliant with the AARC architectural and policy guidelines.
- Resource providers can connect to Check-in to share their resources to different communities.

2.2. Configuration Database

The EGI Configuration Database (GOCDB)[12] 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 the actors (end-users, site managers, NGI managers, support teams, and VO managers), by other tools, and by third parties' middleware to discover information about the infrastructure topology.

The tool provides a web portal for inserting/editing information and 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[13] information model. A comprehensive role-based permissions model controls user permission.

[10] https://argoeu.github.io/guides/messaging/
[11] https://cloud.google.com/pubsub/docs
[12] https://goc.egi.eu/
[13] http://www.ogf.org/documents/GFD.147.pdf

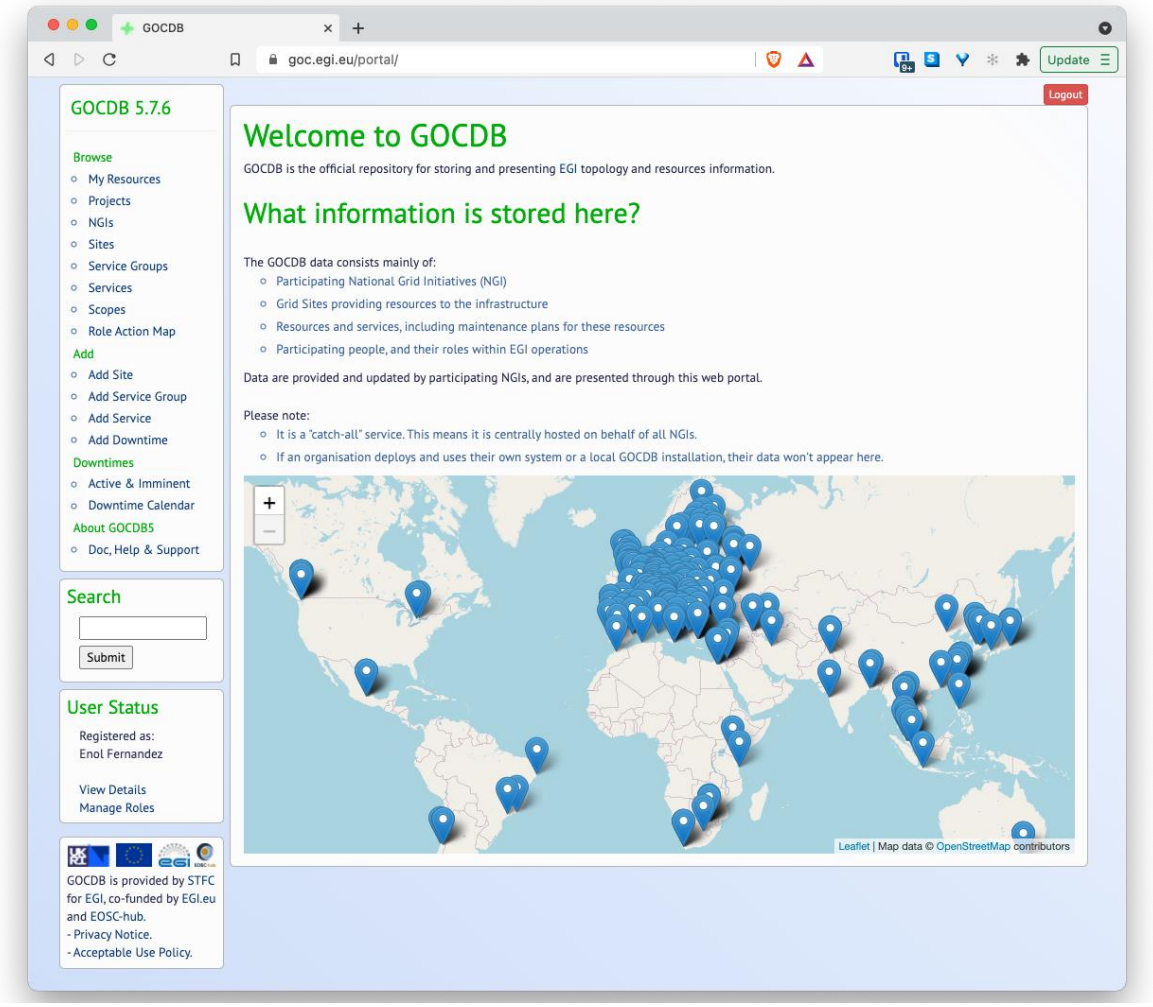

Figure 3 GOCDB web GUI

A flexible tag-cloud mechanism allows objects to be tagged with one or more ‘scope-tags’. This allows resources to be tagged 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' which means they are protected as they are used to restrict tag usage and prevent non authorised sites/services from using tags not intended for them.

Core objects can also be extended using a powerful extensibility mechanism that allows custom key-value pairs to be added to objects. These objects can then be flexibly ‘filtered-by-custom-property’ when selecting / querying data.

2.3. Accounting

EGI Accounting tracks and reports usage 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 usage information about the consumption of resources for High Throughput Compute jobs, Infrastructure as a Service cloud virtual machines and online storage providers. This usage data is collected from those providers that support those types of services, and that connect their service endpoints to the centrally managed Accounting Service.

Probes and sensors that gather accounting information according to certain data formats, are deployed locally at the service providers. Data is forwarded from the sensors into a central Accounting Repository where those data are processed to generate various summaries and views for display in the Accounting Portal[14]. 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.

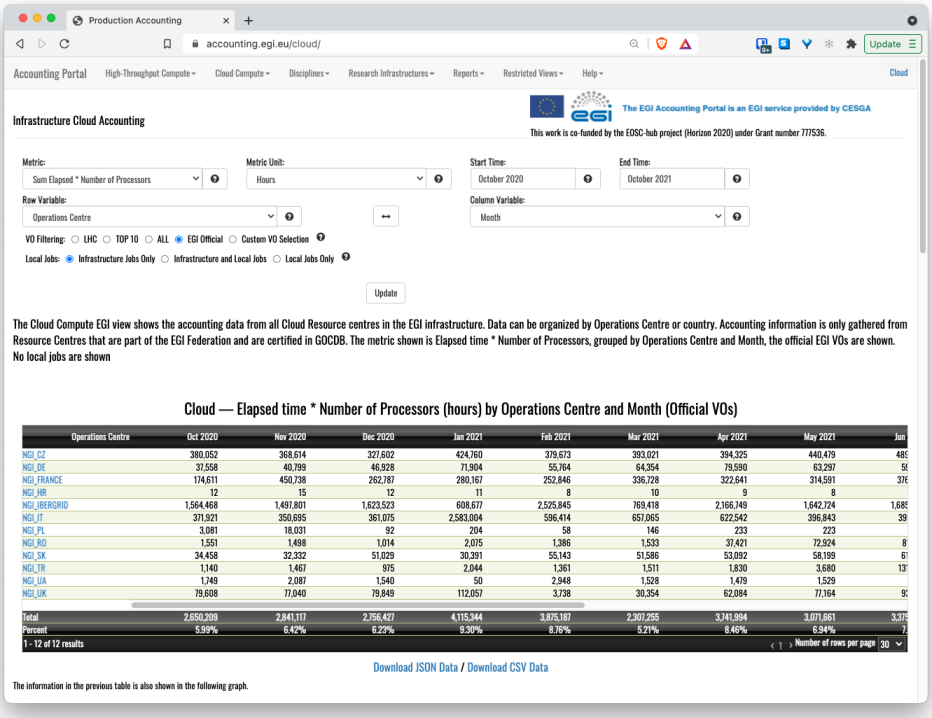

Figure 4 Accounting portal - cloud view.

[14] 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 geographical or community entity (e.g., a country, a community-specific infrastructure).
2. Installation of parsers at the service provider to produce accounting data in the format expected by the Accounting Repository[15]. The parser must be specific to the resource that should be observed from the usage perspective. Ready-to-use parsers are available for:
    a. Grid jobs: https://github.com/apel/apel/tree/dev/apel/parsers
    b. VMs: https://github.com/apel/caso
    c. DPM and dCache storage systems: Part of the relevant storage software release.
3. Accounting records should be sent to the Accounting Repository (directly or via intermediate repositories) using APEL SSM[16].

2.4. Monitoring

Monitoring is a key service 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 organizations. The features of a monitoring system are:

- monitoring of services,
- reporting availability and reliability,
- visualization of the services status,
- providing dashboard interfaces,
- and sending real-time alerts.

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

[15] 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

[16] https://github.com/apel/ssm

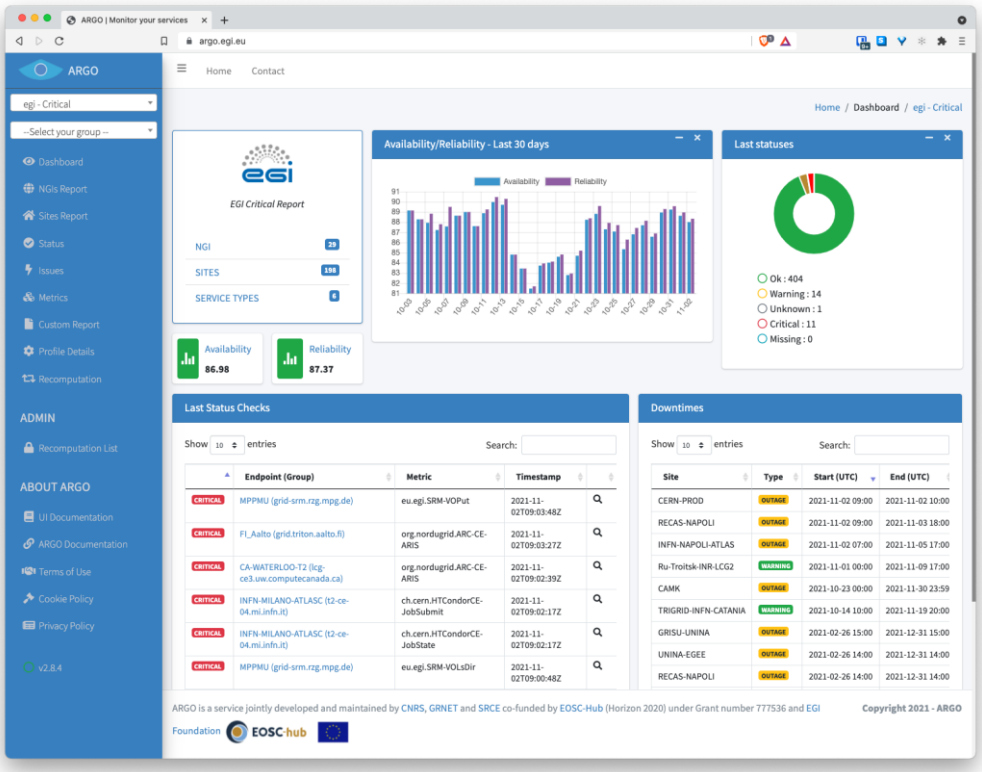

Figure 5 ARGO web dashboard

EGI provides a monitoring service based on the ARGO[17] system. This ARGO Service collects status results from one or more monitoring engine(es) and delivers status results and/or monthly availability (A) and reliability (R) results of distributed services. Both status results and A/R metrics are presented through a Web UI, with the ability for a user to drill-down from the availability of a site to individual services, to individual test results that contributed to the computed figure. ARGO is capable also to send notifications to the service admins in case of a failure/warning on one of the services monitored.

2.5. Helpdesk

The EGI Helpdesk is a distributed tool with central coordination, which provides the information and support needed to troubleshoot product and service problems. Users can report incidents**,** bugs **or** request changes using the GGUS system[18] via its graphical interface as shown in fig.5.

[17] https://argo.egi.eu/, see also EGI webinar https://indico.egi.eu/event/5496/

[18] https://ggus.eu/

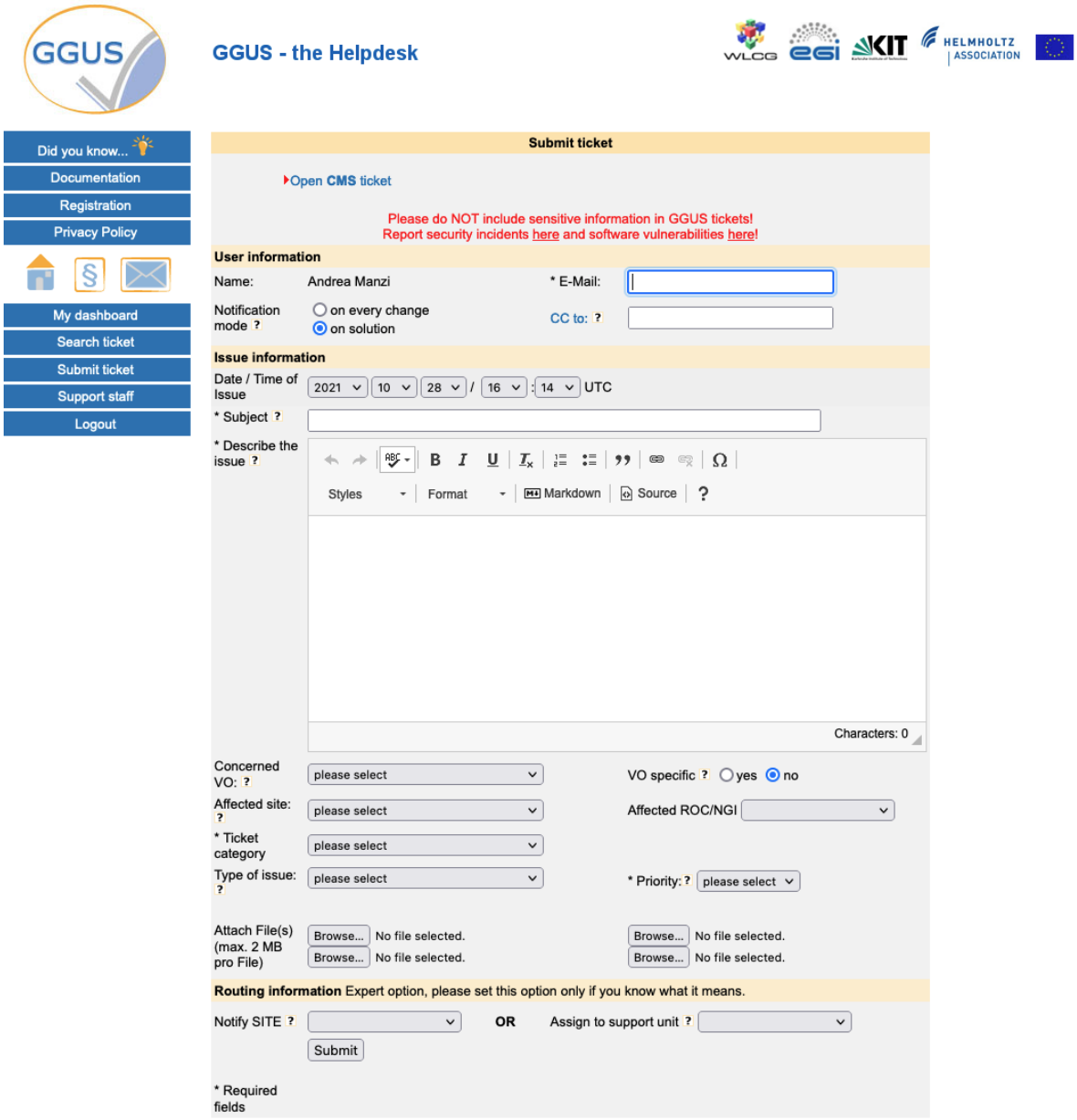

Figure 6 The GGUS Helpdesk

The ticket will be routed to the related Support Unit (SU) assigned to handle the service support request. More information about the EGI Helpdesk is available in the EGI documentation[19].

[19] 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 against their data.

3.1. Integration Modes

The EOSC Compute Platform allows for different modes of integration for the Federated Resource Providers that deliver the compute and storage resources:

- **Full integration**. These providers are fully integrated in the EGI Federation via the Service Management Tools. They are registered in the Configuration Database after a certification process, they deliver usage metrics to the Accounting service, provide support via the Helpdesk, make software available by synchronising with AppDB or deploying CernVM-FS (see Sections 4.3 and 4.4), and rely on Check-in for the authentication and authorization of federated users. This level of integration allows for the seamless usage of the providers in all the upper layers of the EGI-ACE technical architecture and the inclusion of the providers in the SLM (Service Level Management) process of EGI for supporting SLA/OLAs of EGI services.
- **AAI Integration**. In this case, providers use Check-in for authentication and authorization of federated users, but do not necessarily integrate with the rest of the Service Management Tools. While these services are not part of the EGI federation, the hosting organisation must comply with the EGI security requirements to assure that they 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 interaction of the user with the service. Check-in integration allows usage from the compute and data federation layers but it requires individual and manual configuration as these providers cannot be easily discovered. Lack of certification, monitoring and accounting also prevents the inclusion of the services for supporting SLAs/OLAs as the Availability and Reliability of the providers cannot be monitored by EGI.
- **Application Sharing**. Providers that are interested in offering software available in EGI's AppDB and/or CernVM-FS to their local users can configure the synchronization to AppDB and/or the CernVM-FS client and proxy. This allows for local users to easily access software available to EGI communities.
- **Compute/Data Federation integration**. Providers can be individually configured at the Compute and/or Data Federation services and tools to form a hybrid setup where these manually configured providers can be used alongside fully integrated providers, e.g. a oneprovider (see Section 4.1) can be deployed at an external provider and therefore be able to share data with the rest of the federation. Similarly, to other not fully integrated providers, these providers cannot be included in the SLAs/OLAs.

Table 1 Integration modes summary

Integration Mode	Full Integration	AAI integration	Application Sharing	Compute/Data Federation
Check-in	Yes	Yes	No	No
Monitoring/ Accounting/ Helpdesk	Yes	No	No	No
AppDB/ CernVM-FS	Yes	No	Yes	Optional
SLM	Yes	No	No	No
Support for compute and data federation tools	Yes	Yes	No	Yes

Figure 7 depicts the different integration options: fully integrated resource providers shown in the centre of the figure are integrated with Check-in and the rest of EGI Service Management Tools and can be seamlessly used from the Federated Compute and Federated Data Management services available. Non fully federated providers, shown at the top and bottom for compute providers and data providers respectively, can support EGI users by using EGI Check-in (dashed line in the Figure) and can be integrated with the Federated Compute or the Federated Data services.

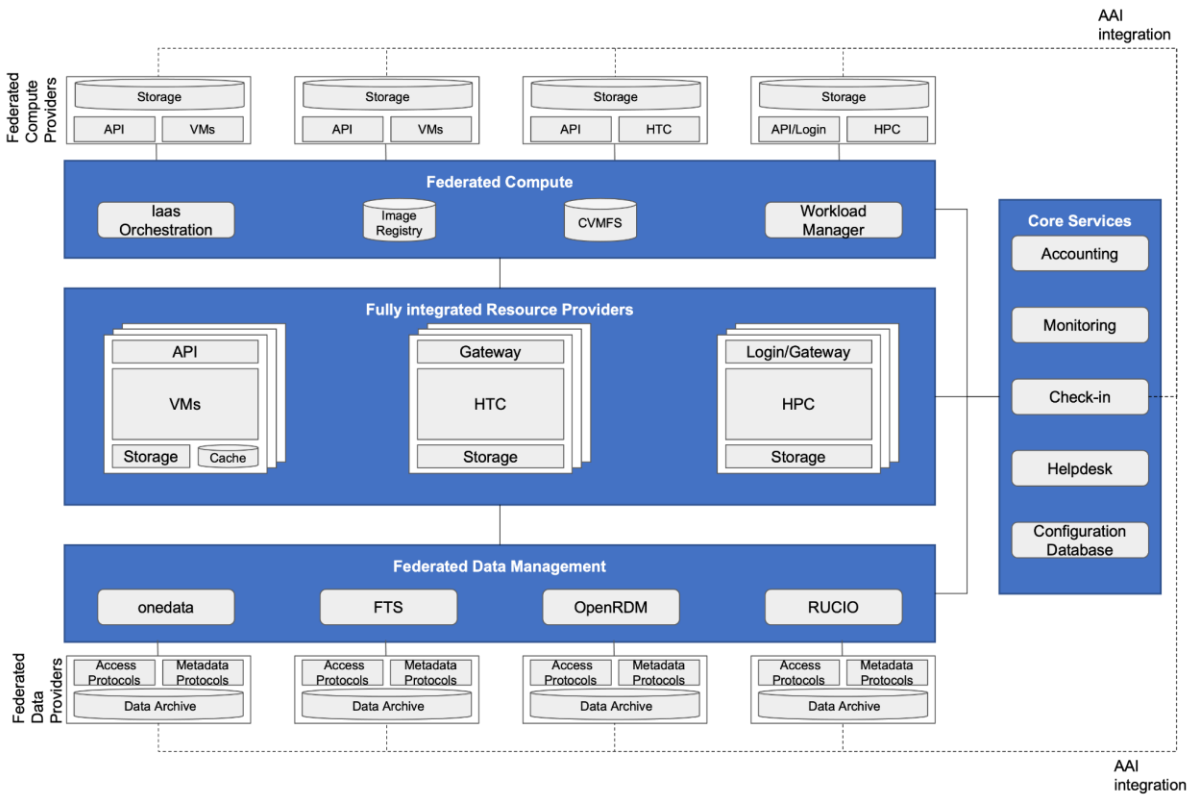

Figure 7 Integration options of the Resource Providers

3.2. Cloud IaaS

Cloud IaaS gives users the ability to deploy and scale virtual machines on-demand. It offers computational resources in a secure and isolated environment managed via an API, without the overhead of managing physical servers. IaaS supports executing multiple kinds of compute- and data-intensive workloads (both batch and interactive); hosting long-running services (e.g. web servers, databases or applications servers); and creating disposable testing and development environments on virtual machines. Users can select the hardware configurations (CPU, memory, GPU, disk) to run their virtual machines and run any Operating System with complete control of the applications and configuration of the system.

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 (VMs), associated Block Storage to enable persistence, and Networks to enable connectivity of the VMs. From a technical perspective, each resource centre of the federated infrastructure operates a Cloud Management Framework (CMF) according to its own preferences and constraints. The resource centre joins the federation by integrating this CMF with the different components of the EGI-ACE Service Management Tools 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[20].

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

- common AAI to enable the use the same identity at all providers;
- application sharing to facilitate finding the community-specific software at the providers;
- monitoring to ensure Availability and Reliability of providers as agreed in Service Level Agreements; and
- central accounting that collects usage information in the federation.

3.2.1. AAI integration

IaaS providers integrated with Check-in allow users to access providers with a single identity. Providers in the federation keep complete control of their services and resources and take authorization decisions locally by using the harmonized user information delivered by Check-in, which includes the Virtual Organizations (VOs) that each user belongs to.

The integration with Check-in uses OpenID Connect and relies in the federated identity features of Keystone[21] (OpenStack authentication service). EGI provides extensive documentation on how the federated user VOs can be used to mapped to local OpenStack projects[22].

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

[21] https://docs.openstack.org/keystone/latest/admin/federation/introduction.htmln

[22] https://docs.egi.eu/providers/cloud-compute/openstack/aai/#keystone-setup

3.2.2. Application Sharing

In a distributed federated IaaS, users need solutions for efficiently managing and distributing their VM images across multiple resource providers. EGI provides AppDB (see Section 4.3.2) as a catalogue of VM images (VMIs) that allows any user to share their VMIs, and communities to select those VMIs relevant for distribution across providers.

AppDB allows representatives of research communities (VOs) to generate a VM image list following the HEPiX image list format[23] that resource centres can subscribe to. The subscription enables the periodic download, conversion, and storage of those images to the local image repository of the provider. The cloudkeeper[24] software provides the support for subscribing to lists and automated synchronisation of images between AppDB and the provider with backends to OpenStack and other Cloud Management Frameworks (OpenNebula, AWS).

3.2.3. Compute and Data Federation

The Compute Federation services of EGI-ACE (see Section 4) support OpenStack providers to be manually configured so they can be used seamlessly by users of the federation layer. The IM orchestrator also supports IaaS cloud providers beyond EGI's OpenStack (e.g OpenNebula, CloudStack, AWS, Azure, GCP among others), see Section 4.2 for more information.

Providers willing to make data available in EGI DataHub service accessible to local users or make local data accessible to other EGI providers can deploy the Oneprovider component[25] (see Section 5.1). Providers with an S3 compatible endpoint can also be used within the File Transfer Service (see Section 5.2).

3.2.4. Full integration

Besides the AAI integration described above, providers going through the full integration mode need to complete the Resource Centre Registration and Certification procedure (PROC09[26]) that will result in the creation of new entries in the Configuration Database after meeting all the policy, security and quality of service requirements for joining the EGI Federation. Once registered, endpoints will be automatically monitored by ARGO. Accounting records can be generated at the provider using cASO.

3.3. 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 different 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 the

[23] https://wiki.appdb.egi.eu/main:faq:vo_image_list_format
[24] https://github.com/the-cloudkeeper-project/cloudkeeper
[25] https://docs.egi.eu/providers/datahub/oneprovider/
[26] https://confluence.egi.eu/display/EGIPP/PROC09+Resource+Centre+Registration+and+Certification

job. The EGI High Throughput Compute[27] infrastructure brings together federated providers delivering HTC resources.

There are two different HTC services considered in EGI-ACE: Grid providers and Spider.

3.3.1. Grid providers

Grid providers deliver a front-end (ARC-CE[28] or HTCondor-CE[29]) for the submission of jobs to a local batch system (e.g. Slurm, PBS, or HTCondor). These systems rely on X.509 certificates and VOMS[30] for authentication and authorization of users. Check-in MasterPortal (see Section 2.1) acts as a bridge for interacting with the grid providers for users relying on federated identity mechanisms.

Besides job-based computing, Grid providers enable storage of files in a fault-tolerant and scalable environment and sharing it 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. Main implementations supported in EGI grid storage are: dCache[31], DPM[32] and StoRM[33].

The grid providers are fully integrated with 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 where all the participating providers can be easily discovered.

CernVM-FS (see Section 4.3) clients can be deployed at the providers to deliver relevant software for the supported communities.

The Workload Manager (see Section 4.1) delivers meta-scheduling of jobs for the distributed Grid providers while FTS (Section 5.2) and Rucio (Section 5.3) deliver Data Transfer and Data Management at the federation level respectively.

3.3.2. Spider

Spider[34] is a versatile high-throughput data-processing platform aimed at processing large structured data sets provided by SURF that runs 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 centers, support for containers, Jupyter notebooks and many other user-centric features.

---

[27] https://docs.egi.eu/providers/high-throughput-compute/
[28] https://www.nordugrid.org/arc/ce/
[29] https://opensciencegrid.org/docs/compute-element/htcondor-ce-overview/
[30] https://italiangrid.github.io/voms/index.html
[31] https://www.dcache.org/
[32] https://twiki.cern.ch/twiki/bin/view/DPM/
[33] https://italiangrid.github.io/storm/
[34] http://doc.spider.surfsara.nl/en/latest/Pages/about.html

Spider is offered as an alternative for Grid users who look for a more customizable system with a low-barrier entry with the same data-processing capabilities as the Spider installation shares the same physical data-processing infrastructure as the Grid-based processing of SURF.

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, it uses CernVM-FS for sharing software and can interact with Grid storage systems. Further integration will be provided as the project evolves.

3.4. High Performance Computing

High Performance Computing (HPC) provides highly optimised computing systems that deliver large amounts of parallel computing power to run applications. In EGI-ACE, a specific task (7.3 HPC Integration) provides interoperability guidelines for HPC systems with the EOSC Cloud Compute platform. The task explores how HPC systems should be exposed to the EOSC portal and how users should interact with them. Four scientific pilot use cases with combined cloud and HPC needs are used for exploring and identifying gaps for the execution of workloads on HPC systems: (1) ELI-NP pilot will explore the configuration of HPC-capable systems on IaaS and compare them to HPC systems, (2) HEP pilot will explore benchmarking, data transfer and execution of codes using federated authentication, (3) ENES pilot will execute docker-based jobs accessing to DataHub on HPC; and (4) PROMINENCE pilot will facilitate running containerised workflows on HPC resources from the PROMINENCE service. The upcoming iteration of this deliverable will provide a complete specification of the integration mechanisms foreseen for HPC providers.

4. Federated Compute

The Federated Compute subsystem in EGI-ACE includes services that facilitate the use of the underlying computing infrastructure in a homogeneous way so users can execute their workloads across the available resource providers in a portable way:

- 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.
- IM (Infrastructure Manager) delivers orchestration of IaaS clouds by automating the deployment of virtual infrastructure on multiple cloud providers. IM supports hybrid deployments and uses a single description language for all supported backends, thus making applications cloud agnostic.
- CernVM-FS and AppDB support the efficient distribution of software in the federation so users can just access the software from any provider of the infrastructure.
- DynamicDNS provides a federation-wide DNS hostname registration service so users can use memorable names instead of IP addresses for interacting with the virtual infrastructures deployed in the infrastructure.

These services are integrated (or are in the process of being integrated) with the Service Management Tools, so they allow federated users to login using Check-in, they are registered in the Configuration Database, they are automatically monitored by ARGO and provide support via EGI's Helpdesk. Accounting records for these services are not yet defined thus integration is not yet in place.

The Federated Compute can be used by the Platform services (described in Section 6) to abstract the individual provider details and simplify the interaction with the infrastructure. IM serves as the main tool for deploying applications for the EC3 (see Section 6.2) and PaaS Orchestrator (see Section 6.3). Services can be also 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 job submission 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[35]. 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 or standalone computing clusters and supercomputers.

[35] http://diracgrid.org

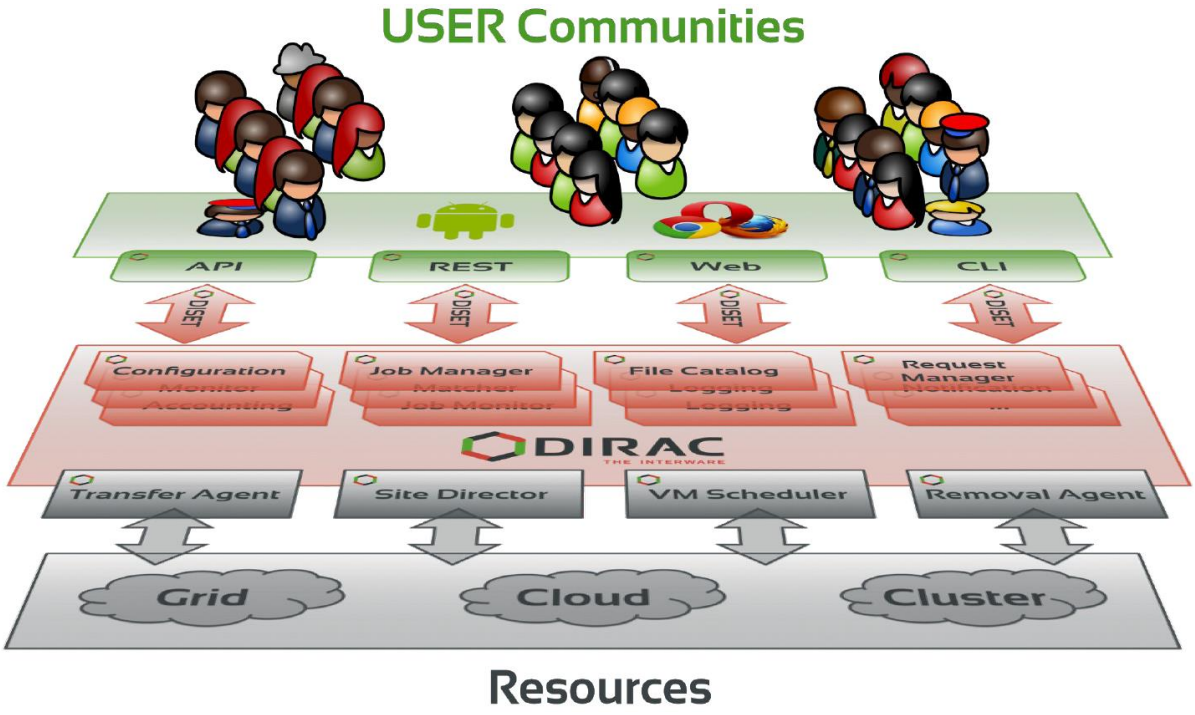

Figure 8 DIRAC Components

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 worker nodes, pilot jobs verify the execution environment and then request user payloads from the central DIRAC 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.

To summarize, WMS serves multiple scientific communities with a user-friendly interface. 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 experiences specific needs,
- Development framework and a set of ready-to-use components to build distributed computing systems of arbitrary complexity.

CC-IN2P3/CNRS hosts the service since April 2021, replacing the previous hosting facility. Service migration was completed in less than 3 months with no loss of previous historical data. During this period, services were deployed on high performance, high memory machines and special care was taken to ensure redundancy and backup of the File Catalog, Accounting and other databases.

These are some use cases as examples of the available functionalities:

- The workload scheduling architecture allows to easily add new resources transparently for the users. It also makes it easy to apply usage policies by defining fine grained priorities to certain activities. This feature, exploited by the WeNMR[36] Collaboration during the COVID pandemic, allowed to quickly make available resources of the sites willing to contribute to the COVID-related studies and ensure 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[37] offers tools to run, explore, diagnose and optimize 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[38] provides access to common simulation tools through a GUI integrated into the WMS framework, allowing the conception of Singularity images to run CORSIKA[39] simulations (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)[40] is a tool that orchestrates the deployment of complex and customized 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 architecture of IM is shown in Figure 8.

[36] https://www.wenmr.eu/
[37] https://openmole.org/
[38] https://projectescape.eu/news/escape-ossr-enhancing-science-through-sharing-software-benefits-use-cases-post-webinar-report
[39] https://www.iap.kit.edu/corsika/
[40] https://www.grycap.upv.es/im/index.php see also EGI webinar https://indico.egi.eu/event/5495/

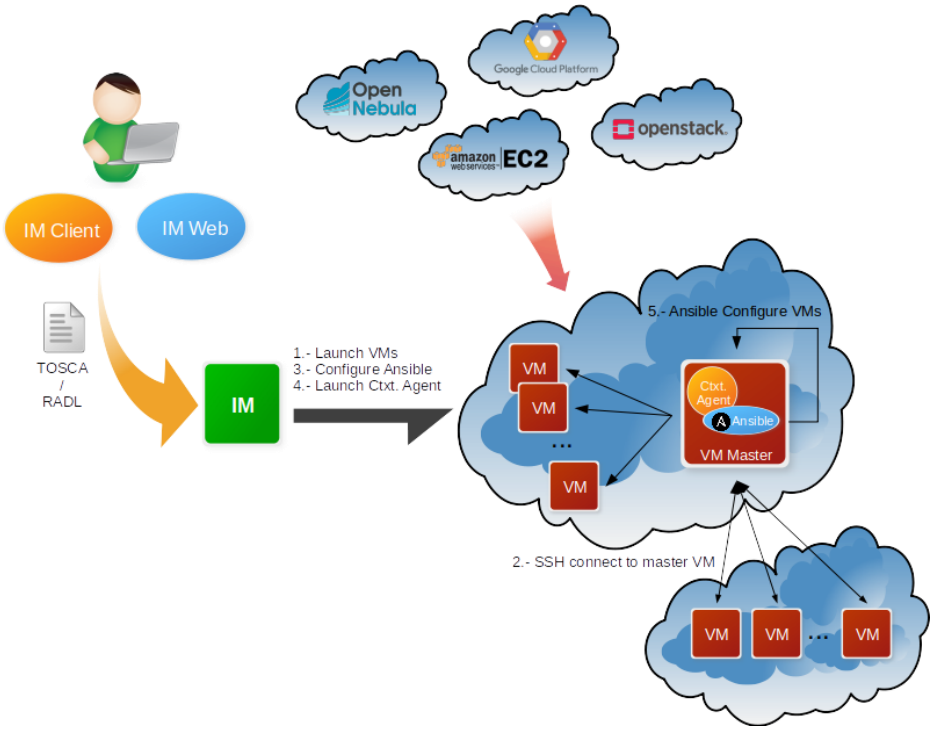

Figure 9 IM Architecture

In order to facilitate deterministic repeatability of virtual infrastructure deployments, the IM service adopts an Infrastructure as Code (IaC) approach that is being widely adopted by the industry. IaC allows to define application architecture using high-level recipes and resort to automated procedures both for virtual computing resources provision and automated configuration of said resources. In addition, it features DevOps capabilities, based on Ansible, enabling the contextualization of the infrastructure at run-time by installing and configuring all the required software that may not be available in the Virtual Machine Images, thus providing the user with a fully functional virtual infrastructure. The virtual infrastructures can be defined using its native language called RADL (Resource and Application Description Language)[41] or the TOSCA OASIS standard (YAML version 1.0)[42].

The main goal of the 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. The IM considers all the aspects related to the creation and management of virtual infrastructures:

- The software and hardware requirements specification for the user applications, using a simple language defined to be easy to understand by non-advanced users who just want to deploy a basic virtual infrastructure, but with enough expressivity for advanced users to set all the configuration parameters needed to get the infrastructure fully configured.

[41] https://imdocs.readthedocs.io/en/latest/radl.html

[42] https://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.html

- The selection of the most suitable Virtual Machine Images (VMI) based on the user expressed requirements.
- The provision of Virtual Machines on the Cloud deployments (or Docker containers in Kubernetes, for example) available to the user, including both public IaaS Clouds (Amazon Web Services, Microsoft Azure, etc.), on-premises Cloud Management Platforms (OpenNebula, OpenStack, etc.) and Container Orchestrators (Kubernetes).
- Support of hybrid infrastructures, where nodes are spread among different cloud providers, enabling Cloud bursting scenarios.
- The contextualization of the infrastructure at run-time by installing and configuring all the required software that may not be available in the images (either VMIs or Docker images).
- The elasticity management, both horizontal (adding/removing nodes) and vertical (growing/shrinking the capacity of nodes).

The IM provides both XML-RPC 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 get information about the infrastructures. It also provides a command line tool (im-client) and two web interfaces: the first one (im-web) has almost all the IM functionality available for advanced users, and the second one (im-dashboard) is designed for the end user that only wants to deploy a set of predefined, and well tested, TOSCA templates with a set of clicks.

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 to 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 in order to maintain file data and meta-data. CernVM-FS uses outgoing HTTP connections only, thereby it avoids most of the firewall issues of other network file systems. It transfers data and meta-data on demand and verifies data integrity by cryptographic hashes.

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

Currently access is by gsi-ssh - with X.509 certificates used for authorization. Support for SAML and OIDC will be added to the service in the coming months which will in turn enable integration with EGI Check-in and Indigo IAM. Optimised support for distributing container images will also be added.

4.4. AppDB

The EGI Applications Database (AppDB)[43] is a service that stores and provides public catalogues about software solutions in the form of native software products and virtual appliances that can be run on the EGI infrastructure, aiming at providing end users with easy to find scientific software and ease of use for cloud VM deployment. It also aims at making it easier for administrators and VO managers to have the most recent version of scientific software readily available on sites, and to monitor availability as well as security issues related to VM image use. As such, the AppDB is involved in the distribution, deployment, and management process of virtual appliances on the EGI FedCloud infrastructure. Its main related components are the *Virtual Appliance Catalogue*, the *VO-wide image list catalogue*, and the *VMCaster subscription service*, portrayed in Figure 10.

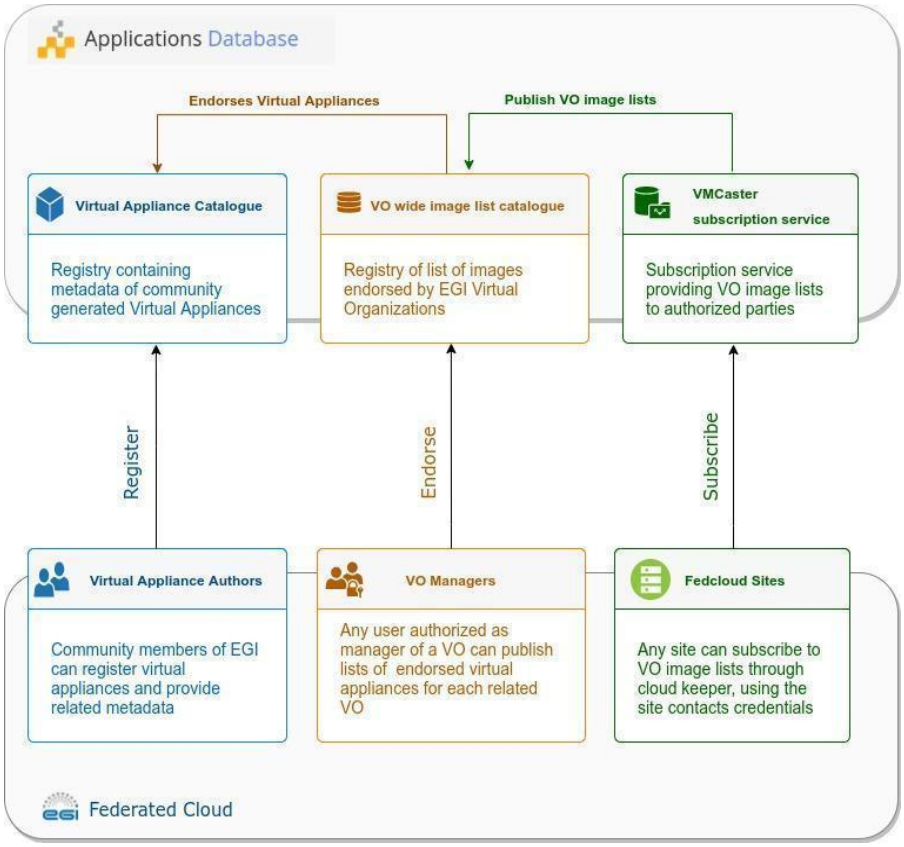

Figure 10 Software distribution components of AppDB

- **Virtual Appliance Catalogue**: A community driven catalog, containing information about virtual appliance solutions, authored by its members. VM authors can register their solutions by providing key metadata such as a description, categorization data, related organizations, projects etc. They may also maintain version-specific information, such as the physical location of each VM version, its expiration date, resource and network requirements, supported operating systems / architectures etc. To improve quality-of-service, AppDB automatically calculates the VM image checksums and provides a continuous delivery system which automates the publishing of new VM versions, without

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

the need to visit the AppDB portal.

- **VO wide image list catalogue**: AppDB integrates with the EGI Operations Portal[44] to collect information about Virtual Organizations. Virtual Organization managers can compose and publish a list of VM images for their VO, by endorsing specific VM versions from the available virtual appliance catalogue, thus declaring which of them are supported by the virtual organization in question. To help keep the image lists up-to-date with the virtual appliance catalogue, AppDB notifies VO managers about new versions and pending policy-based expiration of images of the endorsed virtual appliances

- **VMCaster subscription service**: This component provides authorized third parties, such as EGI Fedcloud providers, with the latest iteration of each VO image list composed by Virtual Organization managers, in a HEPiX-based format. More specifically, a cloud provider that wants to retrieve the endorsed virtual appliances of a virtual organization, may retrieve it from this service, either by using the cloudkeeper[45] software or any other compatible solution. The service authorizes access to site contacts as per the configuration database, in order to protect potentially sensitive information of private virtual appliances.

In order to keep the aforementioned components in sync with the state of the EGI infrastructure and coherently maintain the distributed information, AppDB has been integrated with various services of the ecosystem, aggregating and correlating information. This has given rise to additional services which may be used by other relevant services and 49 parties, apart from AppDB itself. Such services include the Information System[46] which provides infrastructure information regarding the FedCloud and the VMOps dashboard[47] which enables users to deploy and manage VMs on the infrastructure.

4.5. Dynamic DNS

The Dynamic DNS[48] service provides a unified, federation-wide Dynamic DNS support for VMs in EGI Federated Cloud infrastructure. Users can register their chosen meaningful and memorable DNS host names in given domains (e.g. my-server.vo.fedcloud.eu) and assign to public IPs of their servers.

The architecture of the Dynamic DNS is described in Figure 10. The core component of the service is the NS-update server which consists of the GUI portal and NS-update engine. Users can log in to the portal via EGI Check-in account and register hostnames in the supported domains. After hostname registration, users can assign/update the IP addresses of the hostnames via simple commands. All the changes will be sent immediately to DNS servers which are currently located at

[44] https://operations-portal.egi.eu/
[45] https://github.com/the-cloudkeeper-project/cloudkeeper
[46] http://is.marie.hellasgrid.gr/rest/
[47] https://dashboard.appdb.egi.eu/vmops
[48] See EGI webinar https://indico.egi.eu/event/5495/

IISAS Bratislava and LIP Lisbon for high availability. The DNS servers have current TTL (Time-to-Live) for hostnames set as 60s. It means that IP assignments/updates will be refreshed on all computers within the 60s.

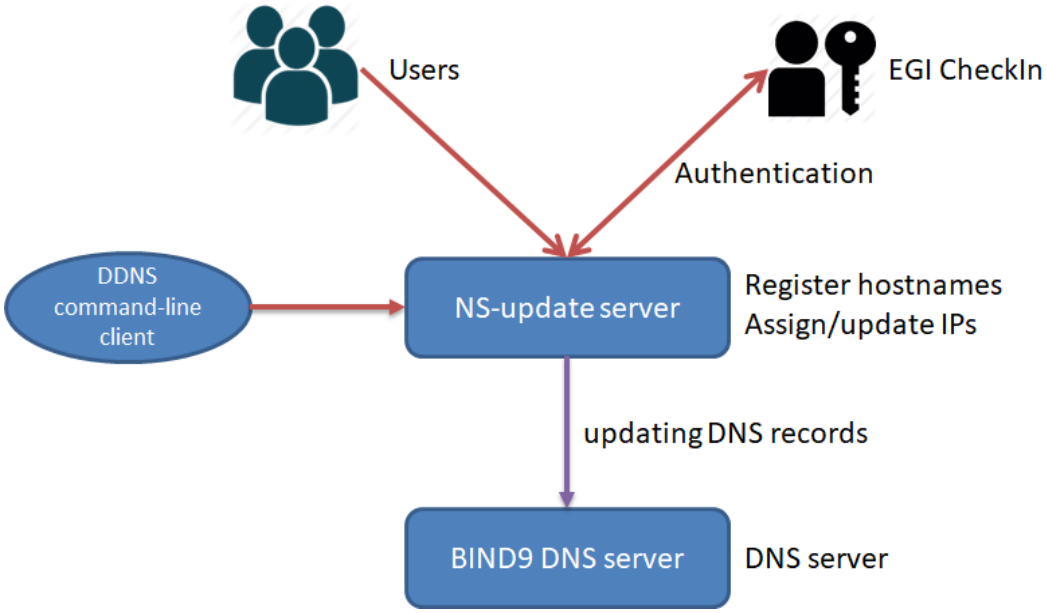

Figure 11Architecture of Dynamic DNS service

By using Dynamic DNS, users can host services in the EGI Federated Cloud with their meaningful service names, can simplify the configuration of clients/servers, can freely move VMs from sites to sites without modifying server/client configurations (federated approach), can request valid server certificates in advance (critical for security) and many other advantages.

The most important thing: the Dynamic DNS service has extremely simple usage, it does not require any additional installations or support from Cloud providers, no special privileges on users' clients or servers, and no user credentials for assigning/updating IP address. That enables easy integration of Dynamic DNS services with other deployment tools/services, e.g. Infrastructure Manager or Terraform, for deploying services in Cloud with registered hostnames from Dynamic DNS service.

5. Federated Data

The Federata Data Subsystem in EGI-ACE comprises a group of services which are providing data management capabilities to enhance the raw storage capacity delivered by the Federated Resource Providers (e.g. DataHub, FTS and Rucio). Moreover, one of the services in this area (openRDM) offers advanced organization of data during ongoing research projects as an integrated environment offering data management and digital lab notebook.

The Federated data services offer APIs and CLIs that are integrated both by the Thematic services and by the Platforms available in EGI-ACE. For instance, the Notebooks service is integrated with DataHub to allow users to access datasets available on the EGI Infrastructure or share outputs among them, and the PaaS orchestrator is integrated with Rucio in order to optimize the application deployment close to where the data is located.

The services are integrated or planned to be integrated within the first year of the EGI-ACE project with the Service Management Tools described in Section 2, thus for instance allowing Federated Identity access, monitoring and accounting.

5.1. DataHub/Onedata

EGI DataHub[49] is a federated service, integrated with EGI Check-in, allowing users to access and share their data from anywhere using either fully restricted access based on access tokens or publicly shared data sets with integrated discovery based on DOI or PID handles.
EGI DataHub is provisioned based on the Onedata distributed data access and management system[50]. Onedata is a globally distributed storage solution, integrating storage services from various providers using possibly heterogeneous underlying technologies, such as NFS or other POSIX-compliant file systems as well as Ceph, S3, GlusterFS, WebDAV and OpenStack SWIFT and provides to clients interfaces based on CDMI, REST API and virtually mounted POSIX filesystem.

[49] https://datahub.egi.eu
[50] https://onedata.org

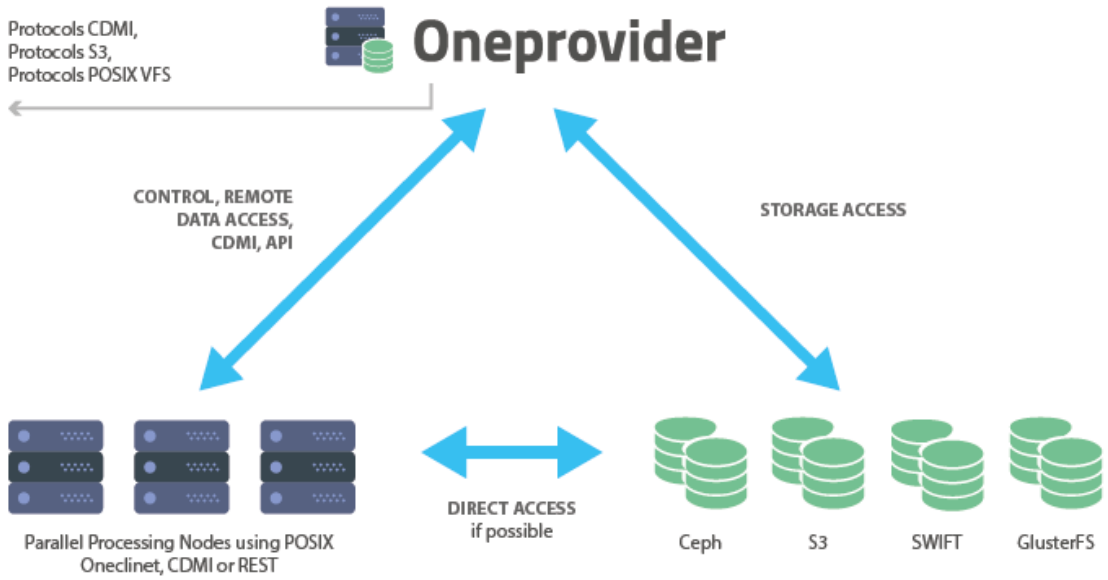

Figure 12 Functional components of Onedata

The main functional components of Onedata include:

- **Onezone** - the federation and authentication service, each Onezone instance (e.g. EGI DataHub) provides a single-sign on to a network of connected storage providers.
- **Oneprovider** - is the main data management component of Onedata, which is deployed in the data centers and is responsible for provisioning the data and managing transfers.
- **Oneclient** - provides access to the virtual filesystem on a VM or host directly via a Fuse mountpoint.
- **OnedataFS** - Python library, allowing access to the distributed virtual filesystem directly from Python applications (e.g. Jupyter Notebooks).

An important aspect of Onedata is a flexible metadata mechanism allowing for storing metadata in the form of simple key value pairs, as well as entire metadata documents (currently in JSON and RDF formats), which can be attached to data resources and used during indexing and querying. Building on top of this metadata mechanism, Onedata enables users to publish their data as open access content. Onedata supports several protocols and standards for open data such as: OAI-PMH[51] for publishing data, handle system integration for registering DOI[52] identifiers, enabling full open data management life cycle management from ingestion through curation to open access.

Onedata enables seamless sharing of data between users, with strict access control. Users can share access to individual files as well as spaces by sending automatically generated access tokens. Authentication in Onedata is based on OpenID Connect[53] standard and enables users to reuse their existing social accounts to authenticate with Onedata services without additional password.

[51] https://www.openarchives.org/pmh/
[52] https://www.doi.org/
[53] https://openid.net/connect/

Onedata has also built-in support for Let's Encrypt[54] certificates, fostering deployment of secure services with valid SSL certificates. Access control is fine grained and managed by a set of privileges assigned in the form of access control lists on the level of users, groups and spaces. Users can create groups for collaborating on a specific space or set of spaces.

All Onedata components have APIs defined using OpenAPI (formerly known as Swagger) specification (version 2.0), 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** - allows control and configuration of local Onezone service deployment, in particular management of users, groups, spaces, shares, providers, handle services, handles and clusters,
- **Oneprovider API** – allows data access through CDMI compatible endpoint as well as data management related tasks including data replication,
- **Onepanel API** – allows administrators to control deployment of other Onedata components, modifying their configuration, e.g. scaling to more nodes or adding new storage resources.

The API documentation can be found at:

- https://onedata.org/#/home/api

The user documentation can be found at:

- https://onedata.org/#/home/documentation/doc/user_guide.html
- https://docs.egi.eu/users/datahub/

5.2. FTS

FTS3 (File transfer System, version 3.0)[55] is a bulk data mover designed to efficiently schedule data transfers. Its purpose is to maximise the usage of available network and storage resources whilst ensuring any policy limits are respected. The components in FTS are: CLI clients, a daemon process for transfer submission, status retrieval and general VO and service configuration, an additional daemon process for staging files from archive, and a database back end. This architecture allows the service to be easily scalable by adding additional resources with identical configuration into an FTS3 cluster.

[54] https://letsencrypt.org/
[55] https://fts.web.cern.ch/fts/, see also EGI webinar https://indico.egi.eu/event/5711/

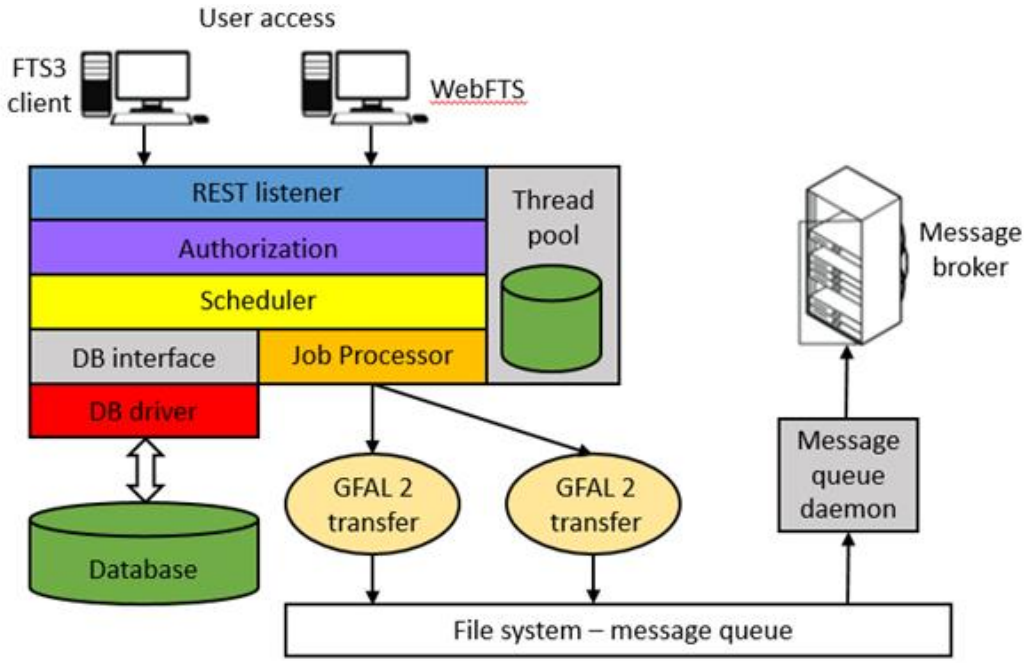

Figure 13 FTS3 service architecture

One of the key features of FTS3 is its adaptive optimization and channel-less transfer model, which allows it to operate as a mesh of network links instead of a predefined structured topology. The optimization algorithm is influenced by both the achieved throughput and success rate of the transfers, and will adjust the number of concurrent transfers accordingly. This makes it possible to run transfers between two random endpoints with good reliability and performance with minimal manual intervention. FTS3 also includes a multidimensional scheduler, which ensures fair share between VOs that share a given link. This default behaviour can be overwritten if necessary in multiple ways: give more slots to a given VO, divide assigned slots by a specific weighting, or prioritise different transfers to give them a boost.

Some additional features of FTS3 are:

- Multiprotocol support through the Grid File Access Library, version 2 (GFAL 2) library.
- REST interface to submit jobs and query their state.
- Third party copy, in particular GridFTP and XrootD, and for certain storage implementations HTTP TPC.
- Session reuse, which is particularly suited to cases where many small files are transferred between two endpoints.
- Staging from tape and archive monitoring.
- Web monitoring.

The core functionality of the service is extended by the inclusion of WebFTS, a standalone web-orientated interface that provides an easily accessible option for end users wishing to utilize the features of FTS3. WebFTS provides the same multiprotocol support as FTS3, but with a two-panel

interface through which users can submit and monitor transfers instead of the command-line client, making it a very useful tool for transferring files between Grid and non-Grid resources.

Currently access to FTS3 is through X.509 certificates or proxy. Moreover, the OIDC authorization support will be added to the service over the coming months which will allow for integration with IAM and EGI Check-in. Work has begun to develop an instance of FTS3 at STFC that is configured for the EGI-ACE communities and will be integrated with EGI Check-in.

5.3. Rucio

Rucio[56] is a data management software designed to manage large volumes of data across multiple sites and a variety of storage end points. It was developed by ATLAS[57] as an open source project, but is now used and developed by a variety of communities within and outside of High Energy Physics. The Rucio instance at Rutherford Appleton Laboratory (RAL) has been developed to be capable of supporting multiple VOs at the same time. This Multi-VO Rucio is a service that aims to leverage the power of Rucio to support the EGI community with a robust and performant data management solution that is being developed with the future of data, and users in mind.

The components within Rucio are the database, server, CLI client, daemon processes, and a tape archive. Rucio orchestrates the movement of data by fulfilling rules created by users. Rucio moves data for users by communicating with the FTS instance at STFC (see above section 5.2) to transfer the data between storage end points. The architecture of Multi-VO Rucio is described in the figure below (Figure 14).

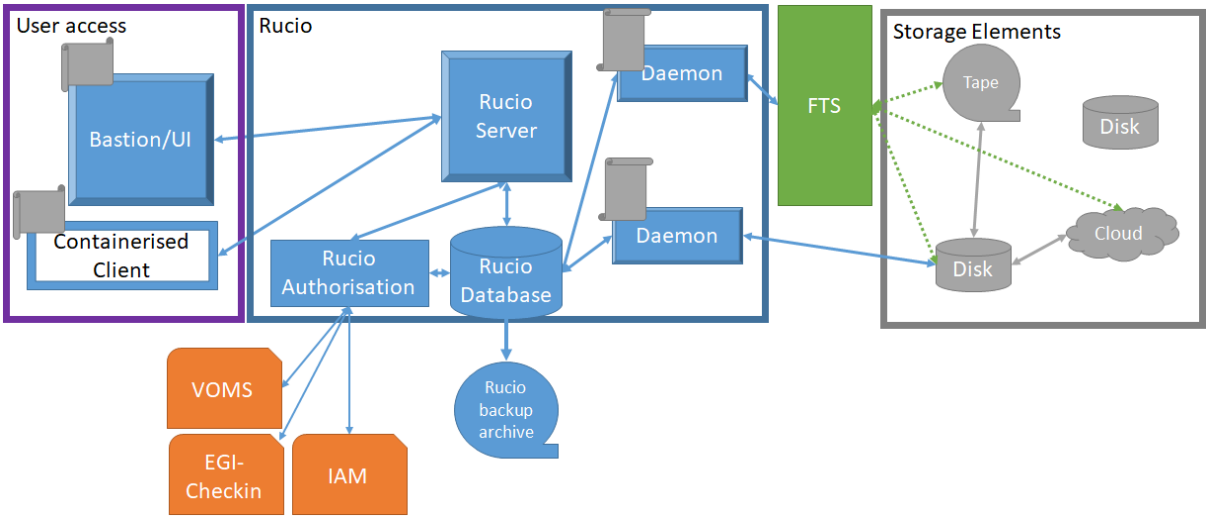

Figure 14 Architecture of the Rucio service. Check-in/IAM integration is under development.

Users access Rucio via the Bastion, a VM hosted at RAL or a containerised client that the user installs on their own machines. Each provides the same CLI interface but the containerised client allows for

[56] https://rucio.cern.ch/, see also EGI webinar https://indico.egi.eu/event/5711/
[57] https://atlas.cern/

mounting of local volumes to facilitate upload and download of data. Further work is planned to make the Rucio WebUI functional within Multi-VO Rucio, this will provide a webpage that allows users to manage their data, account, and review account properties.

Multi-VO Rucio currently mediates authentication and permissions through username and password, X.509, or X.509 proxy. Though, username and password is not recommended. There are development plans over the next few months to support Security Assertion Markup Language (SAML) and OpenID Connect (OIDC) and begin integration with EGI-Checkin (see section 2.1) and the IAM authentication pathways. The integration with EGI-Checkin and IAM will allow for a wider variety of users to authenticate with Rucio as a service.

For redundancy and data preservation, the Rucio database is backed up to an offsite archive daily. There are also plans to have a failover database put into place for quicker recovery of downtime that would be caused by a database outage.

Each VO will inform Rucio of their storage endpoints. Rucio and FTS support a wide range of protocols and endpoints. VOs can customise the permissions model, and schema for data placement with the use of Rucio policy packages. Policy packages are python packages written and maintained by the VO with the support of the Rucio service owner. These Policy Packages, when needed, can be adapted from the generic policy packages within Rucio.

Multi-VO Rucio aims to assist the EGI communities with its powerful Rucio software as a service, helping them to get ready to use Rucio, registering accounts, storage endpoints set up, and customising the policy packages, and providing a point of contact for troubleshooting and assistance.

5.4. openRDM

The *openRDM* service is based around the active research data management (ARDM) platform openBIS[58], developed for the last 12 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 (data annotation, storage and backup).

[58] https://openbis.ch/

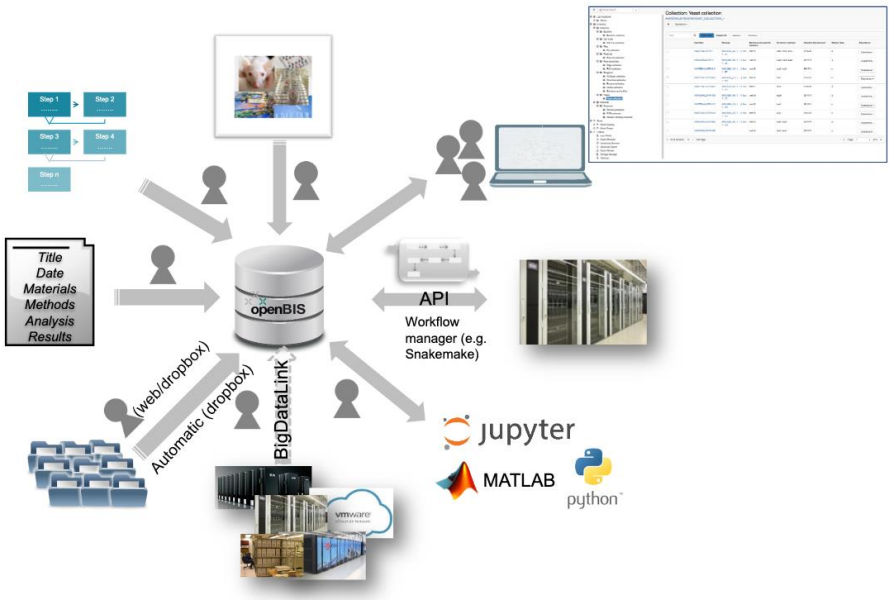

Figure 15 openRDM functionalities

openBIS is a server-client application: the remote server hosts the database and storage backends which are accessed by the users from their local machines via a web browser. openBIS combines a data management platform with a digital lab notebook and a sample and protocol management system. It enables scientists to meet the ever-increasing requirements from funding agencies, journals, and academic institutions to publish data according to the FAIR data principles – according to which data should be Findable, Accessible, Interoperable and Reusable. The system is available in a version specific for life sciences and in a generic version, customizable for other scientific disciplines.

The openRDM.eu service consists of the following service components:

- Installation and configuration of a preview openBIS server on cloud infrastructure: preview is intended for end-users to learn the service & eventually plan on-premise and/or own cloud based deployment.
- Consulting & support for on-premise and/or own cloud-based deployment of openBIS.
- User support including data model generation, data import into openBIS & training for the use of openBIS as a data management platform.

The preview service is currently running on the EGI Cloud IaaS. The virtual machine on which the application is running is deployed in a fully automated procedure based on Ansible scripts. The installation and configuration of the application itself as well as of all third-party dependencies (e.g. the configuration of the shibboleth daemon to allow the application to use EGI Check-in for single sign-on authentication) is also automated through Ansible.

Access to the cloud-based preview instance is provided for end-user testing and onboarding, and in addition the service will be provided as support and consulting for on-premise deployment of the openRDM platform. For the cloud-based preview instance, no data backup nor retention is provided (in line with the concept of a preview instance).

6. Platforms

The services of the platforms area of the EGI-ACE architecture deliver widely used platform-level features that are not discipline specific. They cover the following functionality:

- Interactive Computing, with a Jupyter-based service that allows users to create live documents that run on the EGI infrastructure.
- Scalable deployment of big data tools and clusters: EC3, PaaS Orchestrator and DODAS support the creation of virtual infrastructures on top of IaaS resources, each tool with different levels of abstractions and features.
- AI/ML training with the DEEP training facility service that supports the train-test-evaluation cycle for prototyping of AI models and applications.

As with the Federated Compute and Federated Data areas, these services are integrated or planned to be integrated within the first year of the EGI-ACE project with the Service Management Tools described in Section 2, thus for instance allowing Federated Identity access, monitoring and accounting.

6.1. Notebooks

The EGI Notebooks[59] is a web-based interactive development environment based on the JupyterHub[60] 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 that contain live code, equations, narrative text and rich media output. It is accessible to individually registered users as well as jointly enrolled virtual organisations. Computing resources are provisioned from participating providers integrated with the EGI Federated Cloud. Basic storage capacity is likewise provisioned from cloud sites in the EGI Cloud in the form of IaaS resource, but additional storage can be made available case-by-case from a variety of storage capacity providers, chief among them the Onedata-based EGI DataHub, object storage interfaces exposed by other providers members, or specialized, area-specific dataset providers such as the Sentinels Collaborative Ground Segment for Earth Observation data.

An essential set of notebook images is provided universally to all incoming users, and additional images can be made available upon request with the support of various programming languages like Python, Julia, R, Octave or MATLAB. The EGI notebooks storage is backed-up/restored from S3 provided by CESNET infrastructure. One-off use of alien images is facilitated through the Binder service, which shares the underlying infrastructure of EGI Notebooks. Furthermore, notebook's full integration with GitHub and Zenodo enables users to easily engage with the concept of Open Science.

[59] https://notebooks.egi.eu
[60] https://jupyter.org

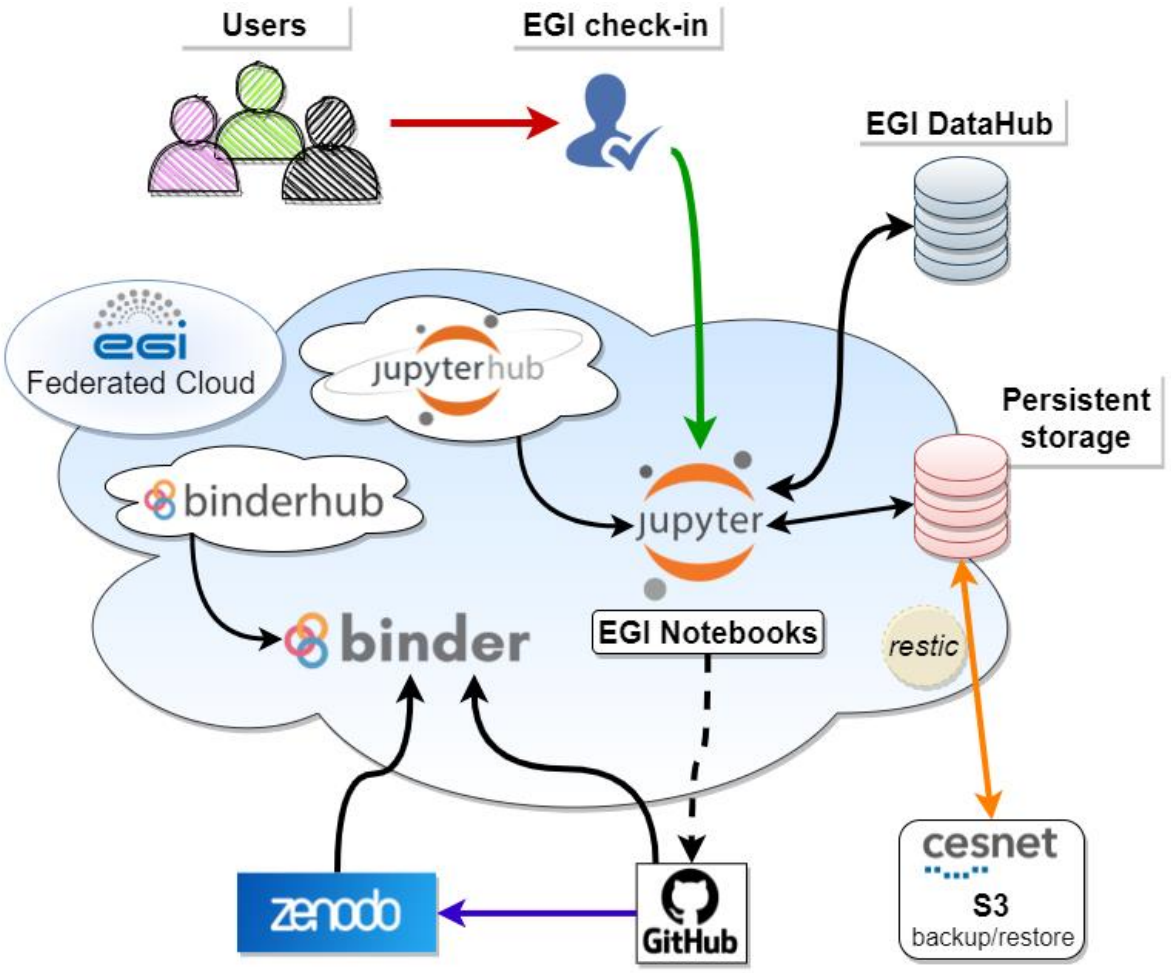

Figure 16 EGI Notebooks architecture

6.2. EC3

Elastic Cloud Computing Cluster (EC3)[61] is a tool to create 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 on-premises (such as OpenNebula and OpenStack). We offer recipes to deploy TORQUE (optionally with MAUI), SLURM, SGE, HTCondor, Mesos, OSCAR, ENES, Nomad and Kubernetes clusters that can be self-managed with CLUES: it starts with a single-node cluster and working nodes will be dynamically deployed and provisioned to fit increasing load (number of jobs at the LRMS). Working nodes will be undeployed when they are idle. This introduces a cost-efficient approach for Cluster-based computing.

[61] https://servproject.i3m.upv.es/ec3/

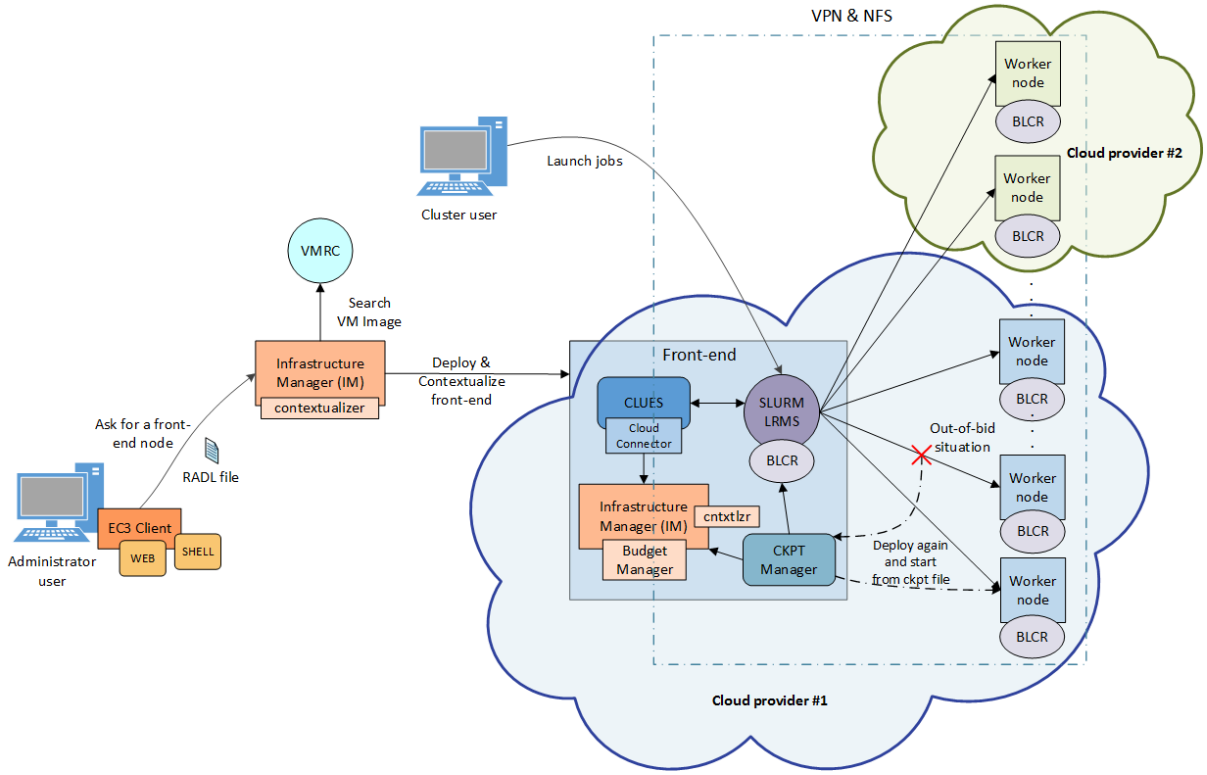

Figure 17 EC3 Architecture

EC3 proposes the combination of Green computing, Cloud computing and HPC techniques to create a tool that deploys elastic virtual clusters on top of IaaS Clouds. EC3 creates elastic cluster-like infrastructures that automatically scale out to a larger number of nodes on demand up to a maximum size specified by the user. Whenever idle resources are detected, the cluster dynamically and automatically scales in, according to some predefined policies, in order to cut down the costs in the case of using a public Cloud provider. This creates the illusion of a real cluster without requiring an investment beyond the actual usage. Therefore, this approach aims at delivering cost-effective elastic Cluster as a Service on top of an IaaS Cloud.

Figure 16 summarizes the main architecture of EC3. The deployment of the virtual elastic cluster consists of two phases. The first one involves starting a VM in the Cloud to act as the cluster front-end while the second one involves the automatic management of the cluster size, depending on the workload and the specified policies. For the first step, a launcher (EC3 Launcher) has been developed that deploys the front-end on the Cloud using the Infrastructure Manager (IM). Once the front-end and the elasticity manager (CLUES) have been deployed, the virtual cluster becomes totally autonomous and every 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 user will have the perception of a cluster with the number of nodes specified as maximum size. CLUES will monitor the working nodes and intercept the job submissions before they arrive at the LRMS, enabling the system to dynamically manage the cluster size transparently to the LRMS and the user, scaling in and out on demand.

Just like in the deployment of the front-end, CLUES internally uses an IM to submit the 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. Thus, the process to deploy the working nodes is similar to the one employed to deploy the front-end.

EC3 supports three deployment models:

- Basic structure (homogeneous cluster). An homogeneous cluster is composed of working nodes that have the same characteristics (hardware and software). This is the basic deployment model of EC3, where only one type of system for the working nodes is used.
- Heterogeneous cluster. This model allows the working nodes comprising the cluster to be of different characteristics (hardware and software). This is of special interest when you need nodes with different configuration or hardware specifications but all working together in the same cluster. It also allows you to configure several queues and specify from which queue the working node belongs to.
- Cloud Bursting (Hybrid clusters). It consists of launching nodes in two or more different Cloud providers. This is done to overcome user quotas or saturated resources. When a limit is reached and no more nodes can be deployed inside the first Cloud Provider, EC3 will launch new nodes in the second defined Cloud provider. This is also called a hybrid cluster. The nodes deployed in different Cloud providers can be different too, so that heterogeneous clusters with cloud bursting capabilities can be deployed and automatically managed with EC3. The nodes would be automatically interconnected by using VPN or SSH tunneling techniques.

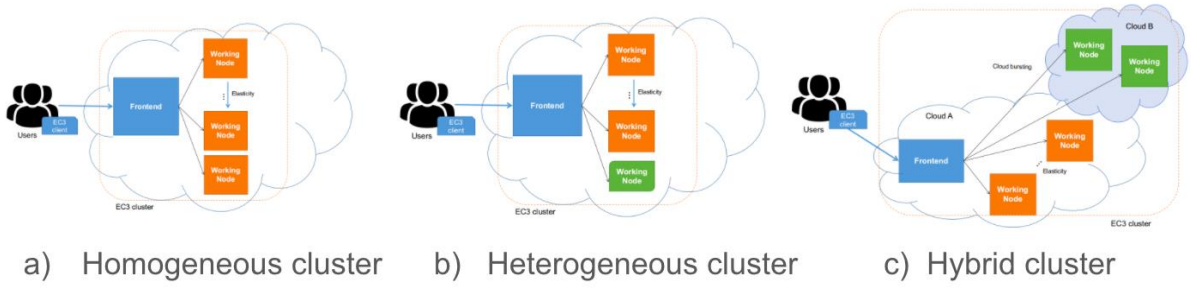

Figure 18 EC3 Deployment models

More information regarding EC3 can be found at https://servproject.i3m.upv.es/ec3-ltos/ or https://github.com/grycap/ec3.

6.3. PaaS Orchestrator

The PaaS Orchestrator is the core component of the INDIGO PaaS[62], an abstraction and federation layer on top of heterogeneous distributed computing environments: it allows to orchestrate and coordinate:

- provisioning of virtualized compute and storage resources on Cloud Management Frameworks, both private and public (like OpenStack, OpenNebula, AWS, etc.),

[62] https://link.springer.com/article/10.1007/s10723-018-9453-3, see also EGI webinar https://indico.egi.eu/event/5720/

- deployment of dockerized long-running services and batch jobs on Container Orchestration Platforms like Apache Mesos and Kubernetes,
- submission and monitoring of HPC jobs on HPC sites through a QCG gateway.

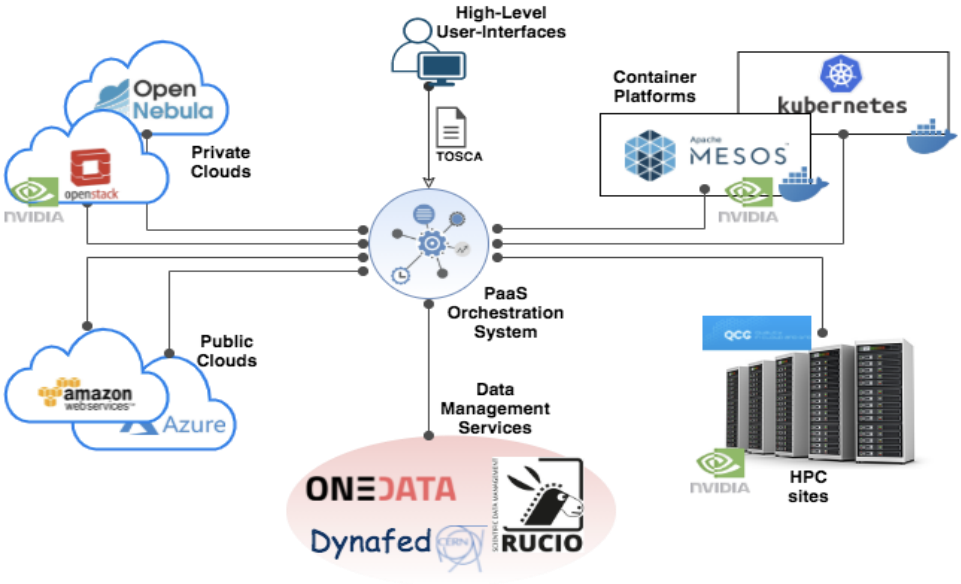

Figure 19 PaaS Orchestrator architecture

The Orchestrator receives the deployment requests, expressed through templates written in TOSCA (Simple Profile in YAML version 1.0), and orchestrates the deployments on the best available cloud sites.

In order to select the best site, the Orchestrator implements a complex workflow: it gathers information about the SLAs signed by the providers with the user, the monitoring data about the availability of the compute and storage services and the location of the data requested by the user (if any).

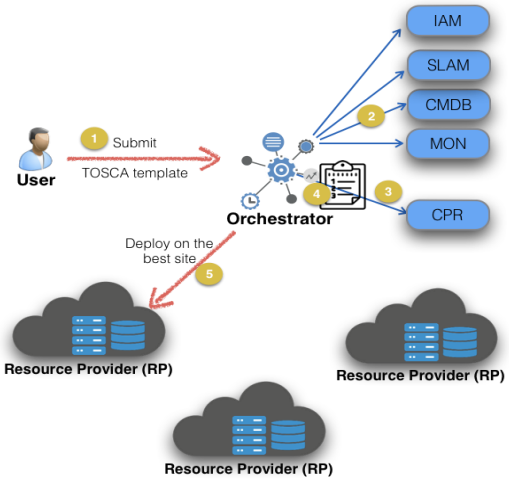

Figure 20 Orchestrator deployment workflow

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

- the Cloud/IaaS adapter implementing the interfaces with the relevant Cloud Management Frameworks through the Infrastructure Manager (external service);
- the Mesos connectors implementing the interfaces that manage the interactions with the relevant cluster framework, Marathon for managing long-running services and Chronos for managing batch-like jobs;
- the Kubernetes adapter implementing the interfaces for managing deployments on Kubernetes cluster;
- the HPC adapter implementing the interfaces for submitting jobs to HPC sites through a QCG Gateway.

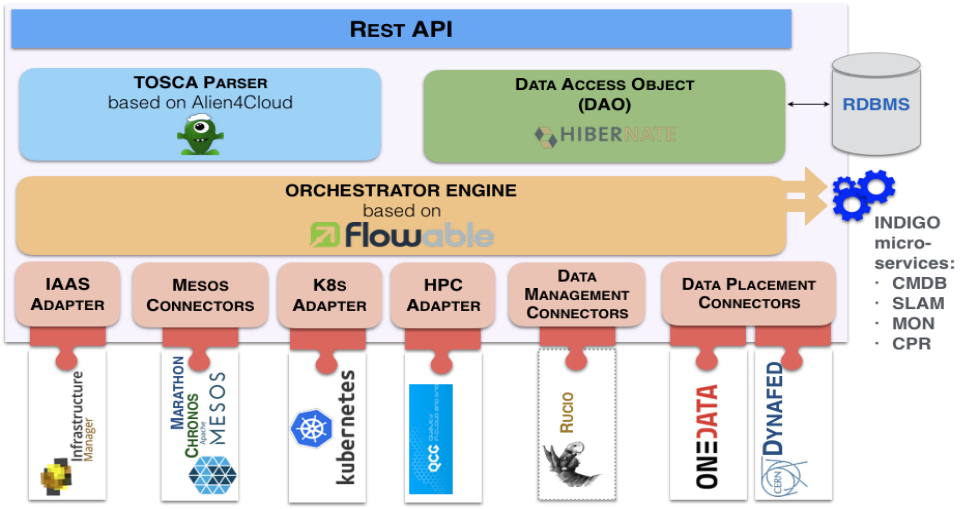

Figure 21 PaaS Orchestrator internal architecture

The PaaS Orchestration system provides the following key features:

- Support for deployments that need specialized hardware resources (e.g. GPUs and Infiniband).
- Support hybrid deployments and network orchestration.
- Automatic retry in case of deployment failure or timeout.
- Integration with Hashicorp Vault 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

DODAS (Dynamic On-Demand Analysis Software)[63] is a service allowing the execution of user analysis code based on batch jobs via command-line, as well as interactively via a Jupyter-based interface. DODAS, which is an open-source software deployed in production and demanding environments, is highly customizable, and is composed of several blocks that can be selected and

[63]See EGI webinar https://indico.egi.eu/event/5695/

combined together to create the best service composition for a given use case. Available blocks allow:

- to simply deploy and use an HTCondor cluster over heterogeneous Cloud resources,
- to combine Jupyter and HTCondor,
- to use the Jupyter interface for analysis,
- to deploy Big Data Pre-Post processing facilities,

without the user needing to take care of resource and software provisioning, set up and management.

Other DODAS blocks support data management through caches to optimize the processing of remote data, or via 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 containerized, and it can be easily customised and adapted to specific needs. Since DODAS natively interfaces with Kubernetes, its building blocks may also be composed via a web-based user interface through Kubeapps.

A DODAS key element is the integration of a flexible, standard-based and federated Authentication and Authorization system, based on INDIGO-IAM. From a technical point of view, this translates into a tight integration of an OIDC token-based mechanism at all the levels of the computing and data stacks. From the user 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 harmonization layer, so that Check-in grants access to the EGI cloud resources, while the IAM harmonised identity is used to manage DODAS domain-specific use cases.

Regarding the technical implementation, as shown in Figure 22, DODAS is highly modular and this allows for a high integration with services also available in the EGI-ACE portfolio of services: Identity and Access Management (INDIGO-IAM), PaaS Orchestrator and Infrastructure Manager (IM).

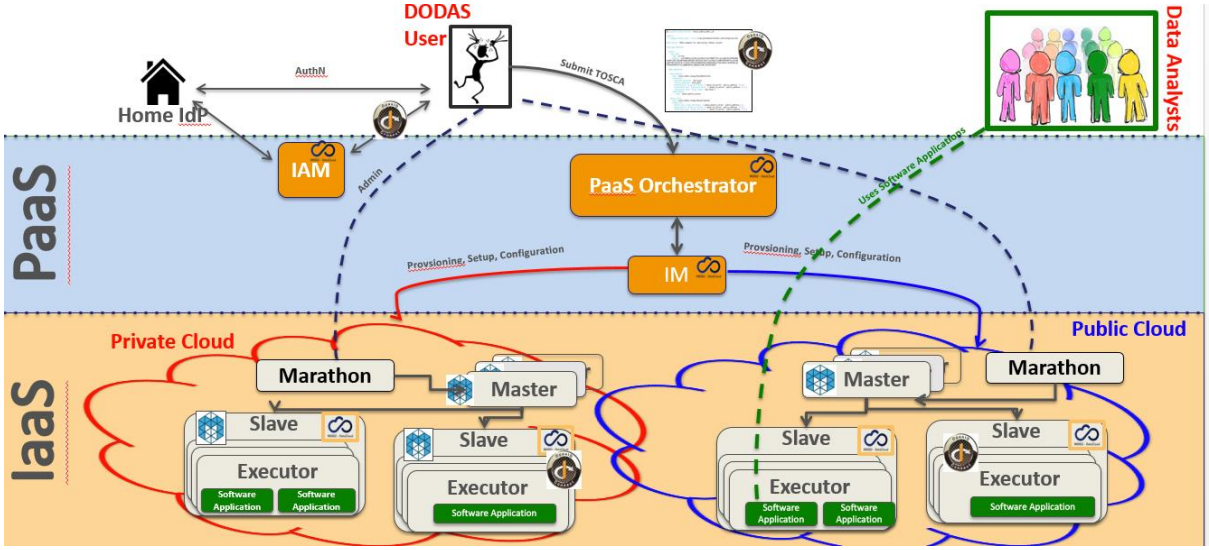

Figure 22 DODAS Architecture

6.5. DEEP training facility

The DEEP platform was developed in the DEEP-Hybrid-DataCloud project[64] and enhanced by services developed in the EOSC-Synergy project[65] and ongoing contributions from the DEEP partners. It offers a complete framework for users, practitioners, and developers of Artificial Intelligence (AI) with various levels of expertise. The framework allows transparent training, sharing, and serving of Machine Learning and Deep Learning (ML/DL) applications both locally and on hybrid cloud systems in the context of the European Open Science Cloud (EOSC). The provided set of tools and services covers the whole ML/DL development cycle, ranging from the models creation, data processing, training, validation and testing to the models serving as a service (through a serverless architecture), sharing and publication, with a DevOps approach. Developers of the services will be able to focus on domain-specific challenges, the rest of issues like AAI, resource management, marketplace, CI/CD software quality assurance will be managed by the DEEP platform.

The DEEP training facility[66] allows the 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. This facility therefore allows access to underlying Cloud, HTC and HPC resources exploiting accelerators, in a user transparent way through a user-friendly dashboard.

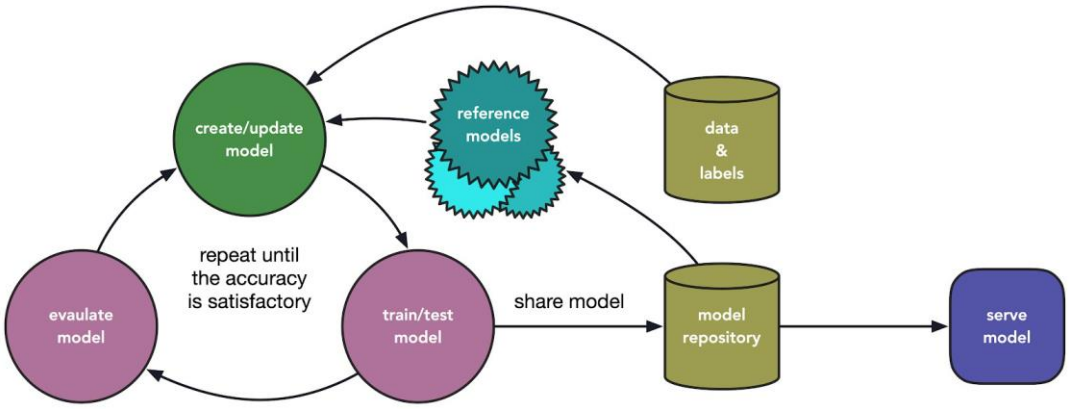

Figure 23 DEEP training model workflow

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

The Dashboard allows users to interact with the modules hosted at the DEEP Open Catalog, as well as deploying external Docker images hosted in Docker Hub. For all deployments, users are able to select the hardware (CPU or GPU), amount of memory and other relevant parameters. Another useful feature is the ability to store the history of all the performed training sessions. This allows

[64] https://deep-hybrid-datacloud.eu/
[65] https://www.eosc-synergy.eu/
[66] https://train.deep-hybrid-datacloud.eu/

monitoring the status of training directly from the training Dashboard keeping track of experiments' results.

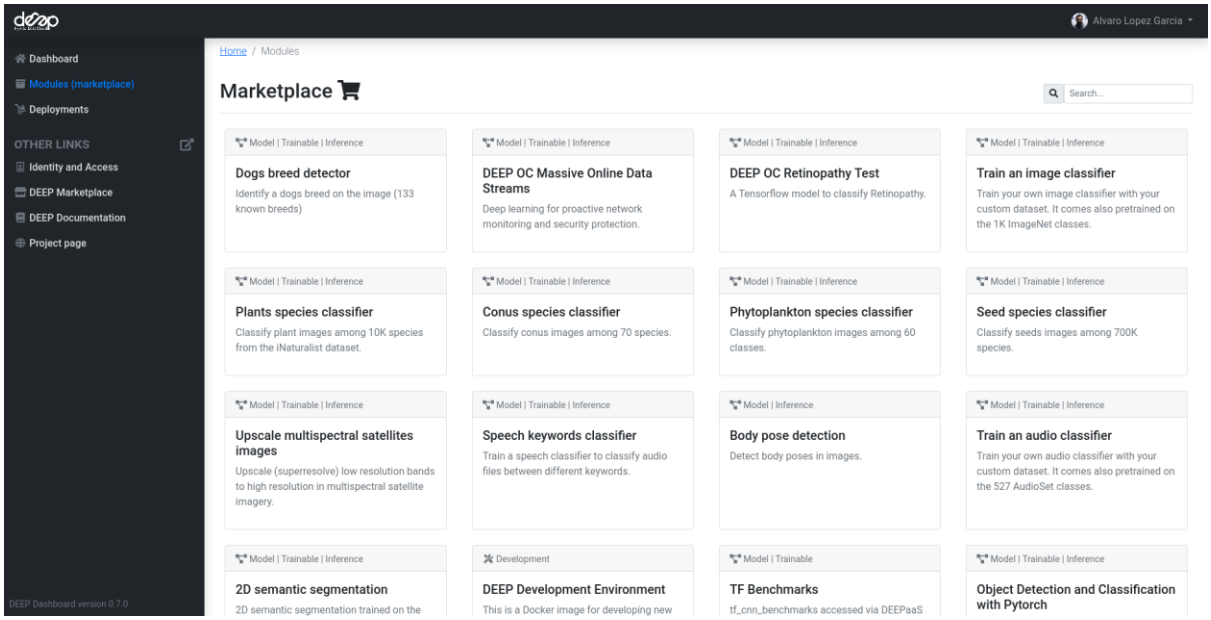

Figure 24 DEEP marketplace

The training installation relies on modules published on the DEEP marketplace, therefore it requires that a user application is containerized inside a Docker container and integrated with the DEEP API in order to expose its functionality. The training system will exploit the existing resources that are integrated with it (OpenStack, Kubernetes, Mesos or HPC systems), through the INDIGO PaaS Orchestrator.

7. Conclusions

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

A set of Service Management Tools provide federated operations (common Configuration Management, AAI, Monitoring, accounting and helpdesk and configuration management) for those providers that participate as a full member of the EGI federation and are an integral part of the services delivered by EGI to the EOSC marketplace. Providers not belonging to the federation can still be included in the EGI-ACE technical architecture and interact with the different services provided by the Compute Federation, Data Federation, Platforms and Thematic Service layers, although their usage will not be managed via EGI's SLA/OLA framework. Providers not fully integrated can rely on common AAI, shared applications, shared data, and hybrid orchestration of computing workloads. Each provider may choose the level of integration that better suits its needs and the communities it serves.

The Federated Compute and Federated Data services 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 needed.

A set of platform-level services provides further generic tools to exploit the compute and storage resources of EGI-ACE (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.

This document will have a second iteration to reflect adaptations needed to better integrate with the EOSC Core as they are defined and developed in EOSC Future project, the EOSC Task forces.