D4.4 Blue Cloud VRE Common Facilities (Release 2)

Work Package	WP4, Developing and operating the Blue Cloud VRE, its services and Virtual Labs
Lead Partner	CNR
Lead Authors (Org)	Massimiliano Assante (CNR), Leonardo Candela (CNR), Pasquale Pagano (CNR)
Contributing Author(s)	Roberto Cirillo(CNR), Andrea Dell'Amico (CNR), Luca Frosini (CNR), Lucio Lelii (CNR), Marco Lettere (Nubisware), Francesco Mangiacrapa (CNR), Giancarlo Panichi (CNR), Fabio Sinibaldi (CNR)
Reviewers	Marco Lettere (Nubisware), Dick M.A. Schaap (MARIS), Sara Pittonet Gaiarin (Trust-IT)
Due Date	31-12-2021, M27
Submission Date	28-12-2021
Version	1.0

Dissemination Level

X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

Blue-Clou Innovative services for Marine Research & the Blue Economy - has received funding from the European Union's Horizon p call BG-07-2019-2020, topic: [A] 2019 - Blue Cloud services, Grant Agreement n. 862409.

DISCLAIMER

"Blue-Cloud, Piloting Innovative services for Marine Research & the Blue Economy" has received funding from the European Union's Horizon programme call BG-07-2019-2020, topic: [A] 2019 - Blue Cloud services, Grant Agreement n.862409.

This document contains information on Blue-Cloud core activities. Any reference to content in this document should clearly indicate the authors, source, organisation, and publication date.

The document has been produced with the funding of the European Commission. The content of this publication is the sole responsibility of the Blue-Cloud Consortium, and it cannot be considered to reflect the views of the European Commission. The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

COPYRIGHT NOTICE

VERSIONING AND CONTRIBUTION HISTORY

Version	*Date*	*Authors*	*Notes*
0.1	16/09/2020	CNR	Definition of the deliverable Table of Contents and planning of contributions
0.2	22/10/2021	CNR	First version of planned contributions integrated into the overall document.
0.3	18/11/2021	CNR	Second version of planned contrib. integrated into the overall document.
0.4	09/12/2021	CNR	Complete version ready for review submitted.
0.5	20/12/2021	MARIS	Reviewed and edits suggested
0.6	20/12/2021	Nubisware	Reviewed and edits suggested
1.0	23/12/2021	CNR, Trust-IT	Final version

Table of Contents

List of Figures

List of Tables

Glossary

Achronym	Definition
AA	Authorization Service
ABAC	Attribute-based access control
API	Application Programming Interface
CMEMS	Copernicus Marine Environment Monitoring Service
CNR	Italian National Research Council
D4Science Infrastructure	Data Infrastructure promoting Open Science (managed by CNR)
DDAS	Data Discovery & Access Service
DIAS	Data and Information Access Service (funded by EC COPERNICUS programme)
EcoTaxa	Web application dedicated to the visual exploration and the taxonomic annotation of images focused on planktonic biodiversity
EOSC	European Open Science Cloud
HDA	Harmonised Data Access
IAM	Access Management Service
IdM	Identity Management Service
OIDC	OpenID Connect
SNL	Social Networking Library
UMA	User-Managed Access
VLab	Virtual Laboratory
VRE	Virtual Research Environment
WEkEO DIAS	WEkEO is one of the 5 Copernicus DIAS, bringing in the CMEMS, C3S and CAMS

Executive summary

The Blue-Cloud project is piloting a cyber platform bringing together and providing access to multidisciplinary data from observations and models, analytical tools, and computing facilities essential to support research to understand better and manage the many aspects of ocean sustainability.

This goal is realised by developing, deploying and operating the Blue-Cloud platform whose architecture consists of two major families of components: (a) the *Blue Cloud Data Discovery and Access* System to serve federated discovery and access to 'blue data' infrastructures; and (b) the *Blue Cloud Virtual Research Environment (VRE)* component to provide a Blue Cloud VRE as a federation of computing platforms and analytical services.

This Deliverable D4.4 "Blue Cloud VRE Common Facilities (Release 2)" is the revised version of the D4.2 "Blue Cloud VRE Common Facilities (Release 1)". This revised version of the document covers the second period of the project, from M13 up to M27, including the up-to-date information of the services reported on D4.2 and the new services that have been developed and added to the VRE common facilities in the reporting period to serve the needs of the Blue Cloud community.

The major changes and new services this deliverable introduces are: an **Orchestrator** (cf. Sec. 3.3), i.e. a software that allows for a declarative, technology agnostic definition of workflows to coordinate the execution of tasks across diverse services and systems; **enhancements to the Workspace service** to support tailored storage persistence and satisfy different application scenarios (cf. Sec. 4.1); enhancements in the **Publishing Framework** (cf. Sec. 6), namely the catalogue extension to deposit catalogue items to Zenodo and the facility to publish geospatial data from the workspace; the facility to **interface with the Data Discovery & Access System** (cf. Sec. 7.1) to transfer datasets of interest into the workspace for future uses; the notebook to facilitate the exploitation of the **WEkEO Harmonised Data Access** (HDA) **API** (cf. Sec. 7.2).

This deliverable also updates the Identity and Access Management (cf. Sec. 3.1) and the Analytics Framework (cf. Sec. 5.1 and Sec 5.2) with minor changes reflecting the activities performed in the reporting period. A description of all the services previously documented in D4.2, not modified in the period, is preserved for this document to be self-contained and provide the reader with an overall description of the whole VRE Common Facilities offering.

A total of 15 services and components are described in this deliverable by reporting their design principles, architecture and main features. These services and components contribute functionalities to the Blue Cloud VRE Enabling Framework (Identity and Access Management, VRE Management, Orchestrator), Collaborative framework (Workspace and Social Networking), Analytics Framework (Software and Algorithm Importer, Smart Executor), Publishing Framework (Catalogue Service) and improved support for RStudio, JupyterHub, ShinyProxy, and Docker Applications. Additionally, two new VRE services, aiming at bridging two VRE external systems such as the the WEkEO[1] catalogue from Copernicus and the Data Discovery and Access from Blue-Cloud with the VRE tools are described. Services and components discussed in this deliverable have contributed to 14 gCube releases, from gCube 4.26 (November 2020) to gCube 5.6.0 (November 2021). They have been used to develop and operate the Virtual Laboratories of the Blue Cloud gateway https://blue-cloud.d4science.org and its underlying infrastructure.

At the time of this deliverable the Blue-Cloud gateway and its services are serving more than 730 users with a total of 19000+ working sessions.

[1] https://www.wekeo.eu

1. Introduction

The Blue Cloud Virtual Research Environment components range from services to promote the collaboration among its users to services supporting the execution of analytics tasks embedded in a distributed computing infrastructure, and to services enabling the co-creation of entire Virtual Laboratories. This VRE component is built on the D4Science infrastructure and the gCube open-source technology (Assante et al. 2019a, 2019b) and deployed in the Blue Cloud gateway (accessible at https://blue-cloud.d4science.org) to make the services and Virtual Laboratories available.

This deliverable updates and complements *Deliverable D4.2 – Blue-Cloud VRE Common Facilities (Release 1)* and the more recent description of the Blue Cloud Virtual Research Environment as presented in *Deliverable D2.7 – Blue-Cloud Architecture (Release 2)* by focusing on both new services and revised versions of existing services that have been further developed in the reporting period (M13 to M27) to serve the needs of the Blue Cloud community.

This document describes selected services by highlighting their design principles, architectures and main features. These components contributed to 14 gCube releases, from gCube 4.26[2] (November 2020) to gCube 5.6.0[3] (November 2021). For each release, the following detailed information are made publicly accessible through the gCube site:

- Build Configuration: the configuration used to build the release;
- Build Report: released components with their version, location, and commit ID on the source control system;
- Tag Report: fingerprint of the release with the commit IDs and tags;
- Release Notes: the aggregated release notes reporting all the major new features and fixed issues for each released component.

The deliverable is organised as follows.

Section 2 briefly recalls the Blue Cloud VRE Architecture and highlights the components that either are new or have been significantly revised in the reporting period in comparison to what was earlier documented in D4.2. and more recently, in D2.7. *Section 3* describes the three services contributing to the Enabling Framework part, namely the Identity and Access Management solution, the VRE Management solution, and the Orchestrator solution. *Section 4* describes the two services contributing to the Collaborative Framework part, i.e., the Workspace and the Social Networking service. *Section 5* documents the six services contributing to the Analytics Framework, namely the Software and Algorithm Importer, the Smart Executor, RStudio, JupyterHub, ShinyProxy, and DataMiner for DockerApps. *Section 6* describes the service contributing to the Publishing Framework, i.e., the Data Catalogue. *Section 7* documents two new VRE services, aiming at bridging two VRE external systems such as the WEkEO[4] catalogue from Copernicus and the Data Discovery and Access from Blue-Cloud with the VRE tools. Finally, *Section 8* concludes the report.

[2] https://code-repo.d4science.org/gCubeCI/gCubeReleases/src/branch/master/2020.md
[3] https://code-repo.d4science.org/gCubeCI/gCubeReleases/src/branch/master#5-6-0
[4] https://www.wekeo.eu

2. Blue Cloud VRE Architecture and latest developments

The Blue Cloud Virtual Research Environment (VRE) components are described in the earlier Deliverable *D4.2 – Blue-Cloud VRE Common Facilities (Release 1)* and further updated in the more recent Deliverable *D2.7 – Blue-Cloud Architecture (Release 2).* To make it easier to follow the changes between the earlier descriptions and this second release, this section highlights the new VRE components, extensions and enhancements that have been implemented for existing VRE components in the period since the last publication (D2.7).

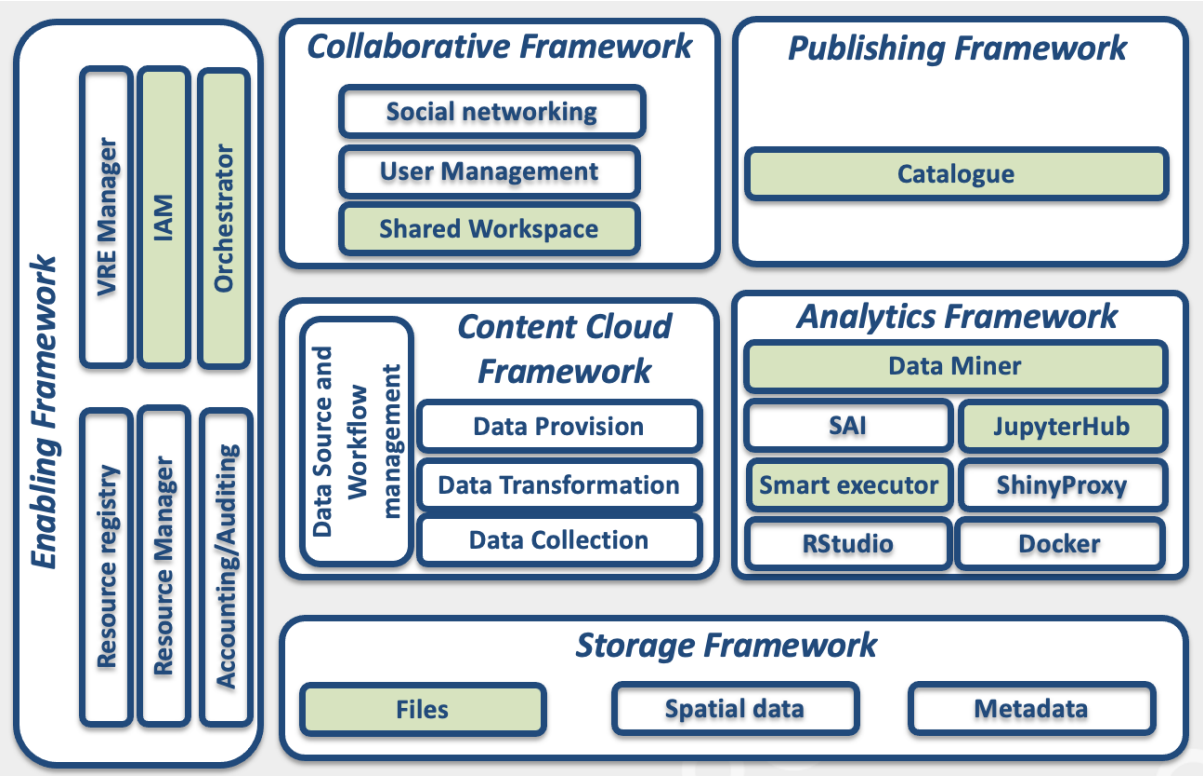

Figure 1. Blue Cloud VRE Architecture at M27 (components updated during the period are shaded in green)

Figure 1 depicts the revised version of the Blue Cloud VRE architecture resulting from the extension and enhancement of the architecture presented in Schaap et al. (2021) Deliverable D2.7 to serve the needs and requirements emerging in the Blue Cloud domain. The shaded boxes in Figure 1 indicate that these components have been updated during the period M13-M27. The architecture consists of services and components organised in 6 frameworks:

- ***Enabling Framework***: it includes services required to support the operation of all services, VREs and VLabs. In particular, it includes (a) **a resource registry service**, to which all e-infrastructure resources (data sources, services, computational nodes, etc.) can be dynamically (de)registered and discovered by users and other services; (b) **Identity and Access Management (IAM) services**, as well as accounting/auditing services, capable of granting and tracking access and usage actions from users; (c) **a VRE management framework** for deploying specialized VREs/VLabs based on a selected subset of "applications". This document complements the description of these components stemming from Schaap et al. (2021) Deliverable D2.7 by documenting the updated solution

for Identity and Access Management (cf. Sec. 3.1), the VRE Management (cf. Sec. 3.2), and the Orchestrator (cf. Sec. 3.3).

- ***Storage Framework***: includes services for efficient, advanced, and on-demand management of digital data represented by files in a distributed file system, collections of metadata records, and time series in spatially-enabled databases. It is used by most other services of the Blue Cloud VRE Architecture and has been extensively discussed in Schaap et al. (2020) Deliverable 2.6 and Schaap et al. (2021) Deliverable D2.7. This part was enhanced during the period to support the Blue-Cloud Dataspace (cf. Sec. 4.1).
- ***Content Cloud Framework***: includes services required to collect, transform, harmonize, and provide, via a variety of APIs, all metadata records published by the D4Science community and those provided by the organizations integrated by the D4Science consortium. It is not planned to be exploited by Blue Cloud and thus is not documented by this document.
- ***Collaborative framework***: supports all VREs and VLabs deployed and provides social networking services, user management services, shared workspace services, and Web-based User Interface access to the information cloud and to the analytics framework, via analytics laboratory services. This document describes the workspace (cf. Sec. 4.1) and the social networking components (cf. Sec. 4.2) of this framework.
- ***Analytics Framework***: includes the services required for executing analytics methods and processes provided by scientists. Compared to the services in Schaap et al. (2021) Deliverable D2.7, this framework has been extended by a revised version of the Smart Executor (cf. Sec. 5.2), the JupyterHub-based solution (cf. Sec. 5.4), and the DataMiner-based solution for DockerApps (cf. Sec. 5.6). While solutions for the Software and Algorithm Importer component (cf. Sec. 5.1) and the ShinyProxy-based solution for ShinyApps (cf. Sec. 5.5) were not subject to modifications during the period.
- ***Publishing framework***: includes services enabling users to document and make "public" any artifact, i.e., made available online. It primarily consists of a catalogue service (cf. Sec. 6.1) where diverse typologies of objects (e.g., datasets, processes, services) can be described, catalogued and made searchable and accessible using user/community defined metadata. To support the publishing of geospatial data, the BlueCloud VRE offers a Spatial Data Catalogue described in Section 6.2
- ***Bridging Systems services***: two new services aiming at bridging two VRE external systems such as the WEkEO[5] catalogue from Copernicus and the Data Discovery and Access from Blue-Cloud with the VRE Common Facilities offering.

[5] https://www.wekeo.eu

3. Enabling Framework Components

According to Section 2, the Blue Cloud VRE Enabling Framework is composed of five distinct macro components. In this chapter, we report on the ones that have been either re-implemented or have been re-designed to adapt to the evolving technologies. The VRE common facilities for the Enabling Framework part are represented by the following three macro components:

- Identity and Access Management (IAM);
- VRE Management;
- Orchestrator.

3.1 Identity and Access Management (IAM)

In the previous architecture, the Identity and Access Management Service (IAM) was "encapsulated" into two different services, namely the Identity Management Service (IdM) and the Authentication and Authorization Service (AA). The former was available into the Gateway service (Liferay web portal technology) and offered features such as login via a federated Identity Provider (e.g. Google, LinkedIn, EOSC Portal) and via the oAuth2 standard. The latter was based on the gCube Authorization framework[6], a token-based authorization system in a gCube-based infrastructure, such as D4Science, compliant with the Attribute-based access control (ABAC)[7] that defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes.

The IdM and the AA services together provided the IAM solution of D4Science for more than 8 years. However, we realised that the technology in which this solution was implemented could not fulfil the requirements, in terms of openness and international standards adoption, of the 'blue' research infrastructures and e-infrastructures Blue-Cloud builds upon. Therefore, we introduced an industry level Authorization Provider software, granting as much functionality as possible and respecting the philosophy of being open-source, extensible by nature and compliant with open and international standards for fine-grained authorization workflows.

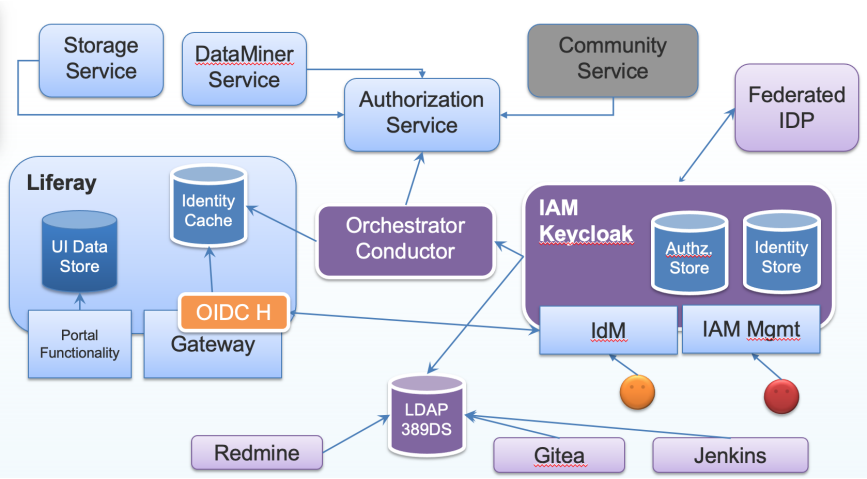

Figure 2. The Identity and Access Management Architecture

[6] https://wiki.gcube-system.org/gcube/Authorization_Framework

[7] https://en.wikipedia.org/wiki/Attribute-based_access_control

Figure 2 shows the architecture of the new IAM solution. It introduces a dedicated IAM Service that governs User Identity, Authorization management including the login process. The entire responsibility of governing Identity Management related workflows is offloaded to the above-mentioned industry quality IAM software component Keycloak[8], including the federation of the Identity Providers (e.g., Google, LinkedIn, EOSC Portal).

The Gateway service (the Liferay web portal technology) only keeps as much User information as necessary to fulfil its navigation and visualization requirements. It uses a user information cache which is created or updated every time a user logs into the Gateway. This makes it much less dependent on technology, version and custom code artefacts.

The Authentication and Authorization Service (Authorization Service in Fig. 2) processes, which will remain supported for backward compatibility, are paired with the one provided by the IAM architecture, that is interoperable by design as it adheres to open standards such as OAuth2, User-Managed Access (UMA), and OpenID Connect (OIDC) protocols.

While the oAuth2 protocol was already supported in the previous IAM solution, UMA and OIDC were not. Both protocols extend the oAuth2 protocol though UMA is more focused on the Authorization part while OIDC on the Authentication one; UMA is a lightweight access control protocol that defines a centralized workflow to allow an Entity to manage access to its resources. Specifically, it gives resource owners granular management of their protected resources by creating authorization policies on a centralized authorization server, this server then authorises who and what can get access to these resources and for how long. OpenID Connect (OIDC) is an identity layer over oAuth2 which uses JSON web tokens (JWT), allowing third party applications to verify the identity of the User based on the authentication performed by an Authorization Server, and to obtain basic user profile information. During the past years OIDC has become the leading standard for single sign-on and identity provision on the Internet.

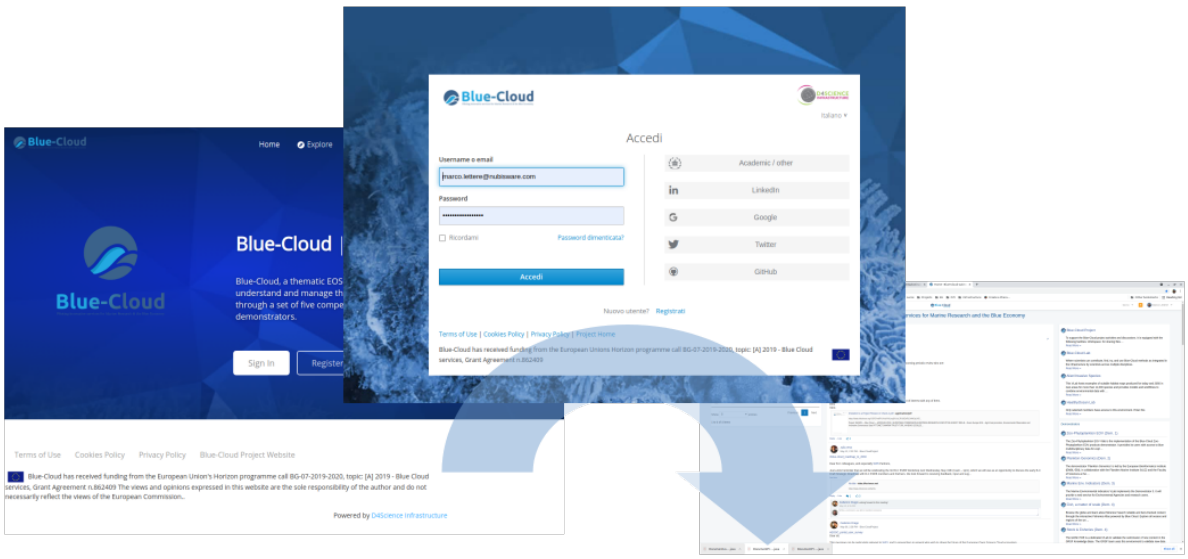

Figure 3.1: Login through authorization code grant flow on the Blue-Cloud Gateway

[8] https://www.keycloak.org

Starting from gCube Release 5.0[9] (Feb. 2021), user agents (i.e., Browsers) are redirected to proper login forms served by the IAM as shown in the figure 3.1. If they are not already signed (SSO), users are asked for their credentials which typically consist of usernames and passwords. At the end of the workflow users are redirected back to the Blue-Cloud Gateway and they will have their credentials exchanged with a JWT [10]Access token. This token instructs the gateway about the user's identity and roles and it could be used to make authorized calls to backend services which do not require other than the User's identity.

Since the deployment to production, the authentication and authorization model has undergone a continuous refinement procedure in order to support new functionality such as external authorized applications or to improve performance and reduce complexity. As an example, the introduction of composite roles has reduced the complexity of some management workflows by one order of magnitude in terms of REST calls to the IAM API.

A central part of the new IAM architecture is represented by the Orchestrator service (cf. Sec. 3.3), supporting the definition of workflows which are then executed by an engine. The orchestrator workflows aim at reducing the burden of services that act as Event sources which are not called to orchestrate and know the details on their own. User related events generated from these services (including the IAM) are sent out to the Orchestrator, which takes care of notifying the "interested" services directly by applying the required workflow. Section 3.3 describes the Orchestrator and its workflows implemented during the period to support these interactions in detail.

Lastly, the IAM service exports data onto a new LDAP (389DS based) server of the Blue-Cloud infrastructure, which is used by *(i)* the software code repository tool (based on Gitea), *(ii)* the software integration tool (based on Jenkins) and *(iii)* the issue tracker tool (based on Redmine), to authenticate project members and allow them access to these tools with their Blue-Cloud credentials.

3.2 VRE Management

VRE Management facilities comprise a set of services and applications offering functions for defining, creating and deploying VLabs. These services support VLab Designers and VRE Managers through graphical user interfaces to instruct the Blue-Cloud infrastructure on the expected features of the desired VLab as well as allowing to easily update the VLab once defined and operational.

The administration of these cooperation environments is a four-tasks activity envisaging:

- a definition phase in which a VLab Designer specifies the characteristics of a new VLab to serve an application scenario;
- an approval phase in which the VRE Manager decides whether the specified VLabs can be accepted or rejected. For the accepted VLab, the VRE Manager decides also how this VLab has to be deployed, e.g. which hosting nodes will be exploited;
- a verification phase in which the VRE Manager validates a VLab resulting from the approval phase;
- a management phase in which the VRE Manager operates on a deployed VLab in order to customise specific aspects (e.g. the layout of user interfaces constituents a.k.a. portlets) and monitors the operational state of the VRE as a whole.

[9] https://code-repo.d4science.org/gCubeCI/gCubeReleases/src/branch/master#5-0-0
[10] JWT: https://jwt.io

Further details and information can be found in the gCube Wiki page related to the service at the web address: https://wiki.gcube-system.org/gcube/VRE_Administration.

These services pre-date Blue-Cloud, during the period they have been re-designed to adapt to the evolving technologies.

3.2.2 VRE Enabling front-end applications

A set of interaction-oriented services and front-end applications providing functions to support VRE Manager in the above-mentioned phases.

Although these front-end applications (portlets) pre-date Blue-Cloud, during the project period they have been re-designed to adapt to the evolving technologies, in particular, they are now responsive (capable of displaying on different devices such as smartphones, tablets and desktop).

3.2.2.1 VRE Definition portlet

The VRE Definition portlet assists the *definition phase*, the procedure is performed by the VRE Designer and leads to the specification of a Virtual Lab Environment, i.e. the selection of the resources and the identification of other characteristics describing the desiderata for a VLab devised to serve the needs of a specific demonstrator.

This procedure is supported by a dedicated portlet, the VRE Definition Portlet, that implements a wizard-based approach.

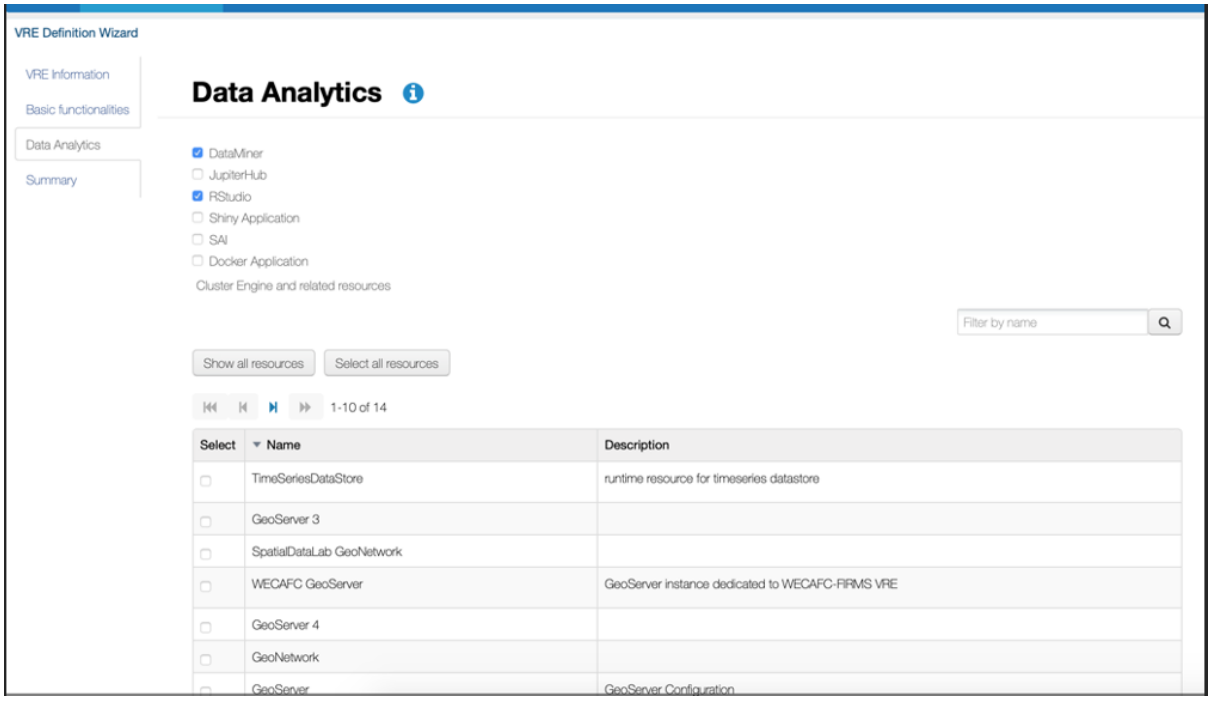

Figure 3.2. The VRE Definition portlet: the Data Analytics step

Figure 3.2 shows the selection of the Data Analytics resources; the user of the VRE Designer is provided with descriptive information to select the resources to add to the VLab, e.g. for Data Analytics, the DataMiner Service, the list of Server and Algorithms available for deployment.

3.2.2.2 VRE Manager portlet

The VRE manager portlet (vre-deploy) assists with the *approval and verification phase,* the action after the definition phase, aiming at actually deploying the VLab. To perform this action, the VRE Manager should use the VRE Manager portlet.

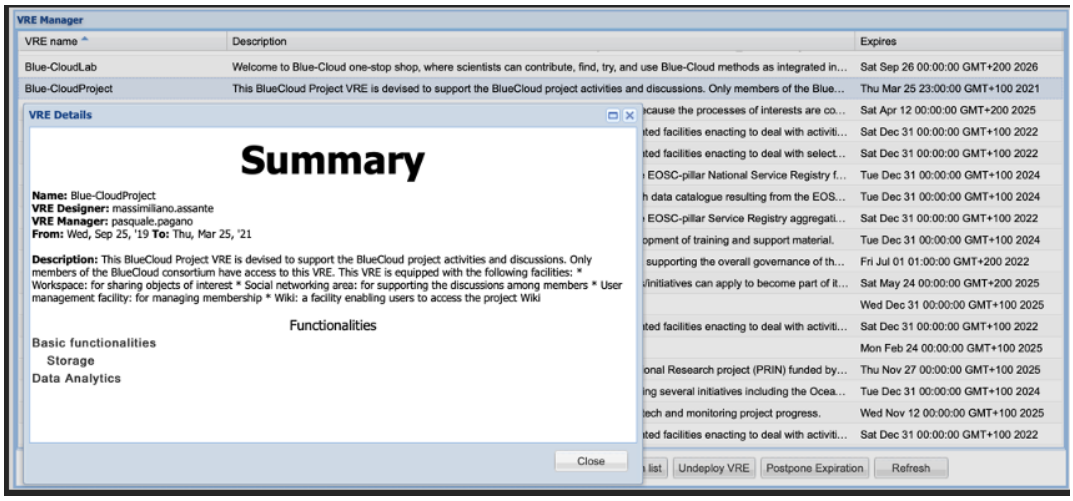

Figure 3.3. The VRE Manager Portlet: a screenshot

VLabs to be approved are in the "Pending" status. Using the "Action" menu, the VRE Manager can analyse the VRE definition ("View Definition"), edit a VRE definition made by a VRE Designer ("Edit"), start the approval phase ("Approve") or withdraw the VLab designed by a VRE Designer ("Withdraw").

3.3 Orchestrator (new service)

An Orchestrator is a software that allows for a declarative, technology agnostic definition of workflows which are then executed by an engine. Decoupling orchestration logic from the internals of single services enables a more scalable and manageable approach to complex procedures, facilitates tracking, monitoring and inspection of service interactions and finally provides a non-opinionated location for concentrating cross service logic.

The technology that has been chosen for the implementation of the orchestrator is the Conductor by Netflix[11]. Conductor is an operation oriented micro service orchestration software which favors configuration and code over design. It is designed as a distributed system from the ground up and supports several control structures such as fork-join, dynamic-fork-join, loop, termination, lambda, events.

Custom task definitions can be added by plugging worker processes which can poll for availability of tasks from queues. Definitions of workflows are written in JSON and all administrative operations are performed through a proper REST API.

[11] https://netflix.github.io/conductor/

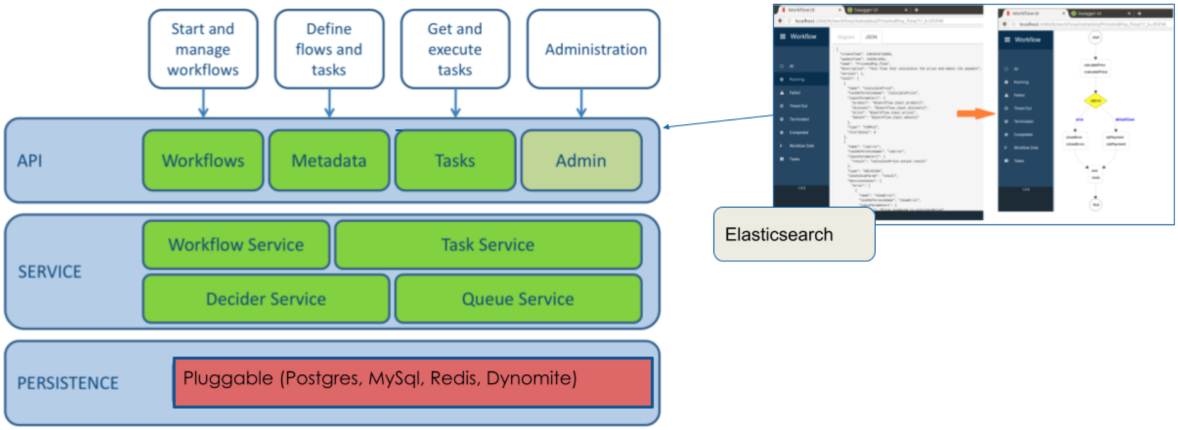

Figure 3.4: Orchestrator Architecture

Components such as the Portal can eventually start a workflow by sending the required input operation through a simple REST call.

Both REST API and Administration UI are protected by a proper authorisation control layer which requires clients and human operators to have a proper token or credential set to operate on the orchestrator.

Workflows, in a micro-service oriented architecture, are complex combinations of tasks (mostly HTTP calls) with the possibility to exploit common distributed computing patterns such as *retrial-on-error*, *conditional branching*, *fork-join* and *dynamic fork-join* for concurrent execution, *conditional-looping*.

Effective degree of parallelism by which operations are executed depends on the number of workers that are allocated to perform a particular task.

Table 1 shows the currently instantiated workers on the production environment for Blue-Cloud.

Task type	#Workers	Description
PyRest	6	Powerful worker written in Python for making HTTP calls.
PyRestBridge	6	Worker written in Python for directly transferring the download from one web URL to an upload to another URL.
PyMail	6	Worker written in Python for sending text and HTML mails with support for variables and interpolation.
PyEval	6	Execution of Python lambda expressions.

Table 1: Orchestrator Workers with their type and number on Blue-Cloud VRE

Table 2 shows the workflows that have been implemented in order to manage the Blue-Cloud Gateway and VRE (as reported in Schaap et al. (2021) Deliverable D2.7) plus a functional workflow for Data Discovery & Access Service integration with the VRE described in Section 7.1.

Name	Category	Description
group_created	IAM, VRE management	Handle creation of a new VLab
group_deleted	IAM, VRE management	Handle removal of VLab
role_created	IAM, VRE management	Handle creation of new role for VLab
role_deleted	IAM, VRE management	Handle removal of role from VLab
user-group_created	IAM, User management	Handle addition of a User to a VLab
user-group_deleted	IAM, User management	Handle removal of a User from VLab
invitation_accepted	IAM, User management	Handle event of User accepting email invitation to join a VLab
user-group-role_created	IAM, User management	Handle assignment of a VLab Role to a User
user-group-role_deleted	IAM, User management	Handle unassignment of a VLab role from a User
delete-user-account	IAM, User management	Handle cancellation of User account
create_system_service	IAM, Application management	Handle the creation of a Software application that needs to operate inside VLabs it is authorized for.
da_cache_to_shub	Application integration	Synchronize data artefacts fetched from the Data Discovery & Access service to a folder in a User's workspace.

Table 2: Orchestrator Workflows available on Blue-Cloud VRE

As an example, the following picture shows the execution of a user_group_deleted workflow where an error occurred. This demonstrates how important it is to have a clear tracking of errors and thus the possibility to decide what actions need to be taken to fix (including the possibility of replaying a workflow as a whole).

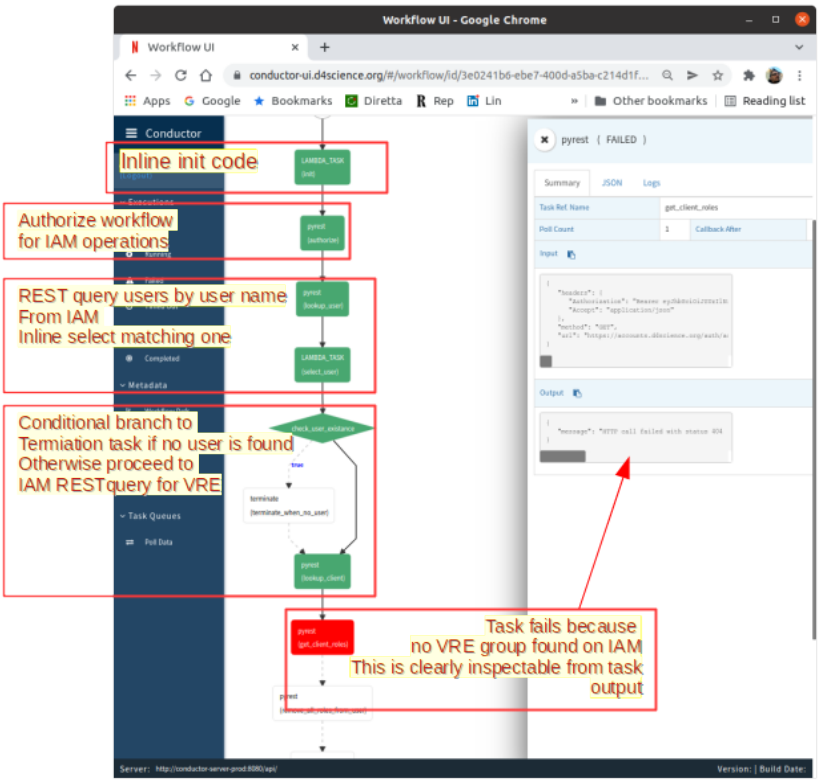

Figure 3.5: Orchestrator Workflow Example

4. Collaborative Framework Components

The Blue Cloud VRE Collaborative Framework includes a set of services and components enabling their users to collaborate on an activity or a project by sharing material and communicating in smart and flexible ways. In this section, two constituents of this framework are detailed:

- The workspace component, to organise and share digital material using an interface resembling a standard file system with items organised in folders.
- Social networking, to have discussions and information exchanges using social networking approaches and practices (e.g. post, hashtags, mentions, likes).

4.1 Workspace (new service)

The workspace service is a key component of every D4Science-based VRE and VLab. It resembles a typical file system with files and other items organised in folders. Internally, it supports an open-ended set of items that (i) contain rich and extensible metadata and (ii) rely on an array of storage solutions. The workspace is fully integrated with the rest of services of a VRE / VLab (Figure 5) to facilitate access to content and to store new content. It is integrated with most of the analytics components, i.e. Data Miner, the Software and Algorithm Importer, RStudio and JupyterHub (cf. Sec. 5). It is integrated with the publishing framework meaning that contents of the workspace can be published via the Data Catalogue (cf. Sec. 6). Moreover, the workspace allows sharing of sections of it thus facilitating collaborative practices (Assante et al. 2019b).

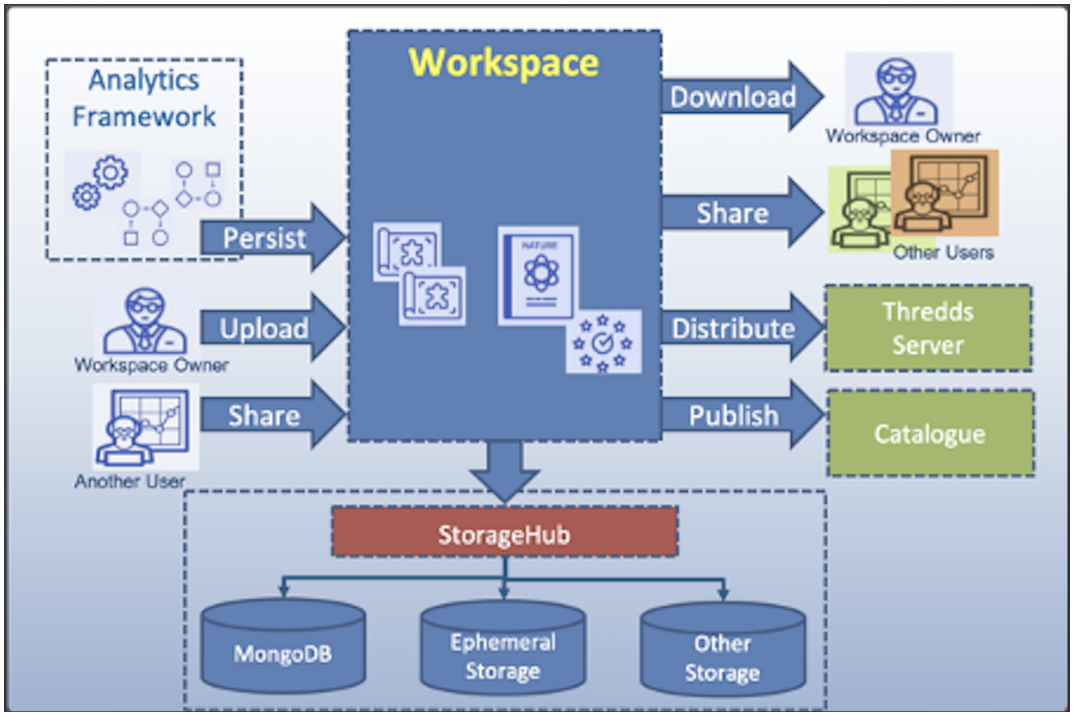

Figure 5. Workspace interactions diagram

The graphical user interface of the workspace is reported in Figure 6.

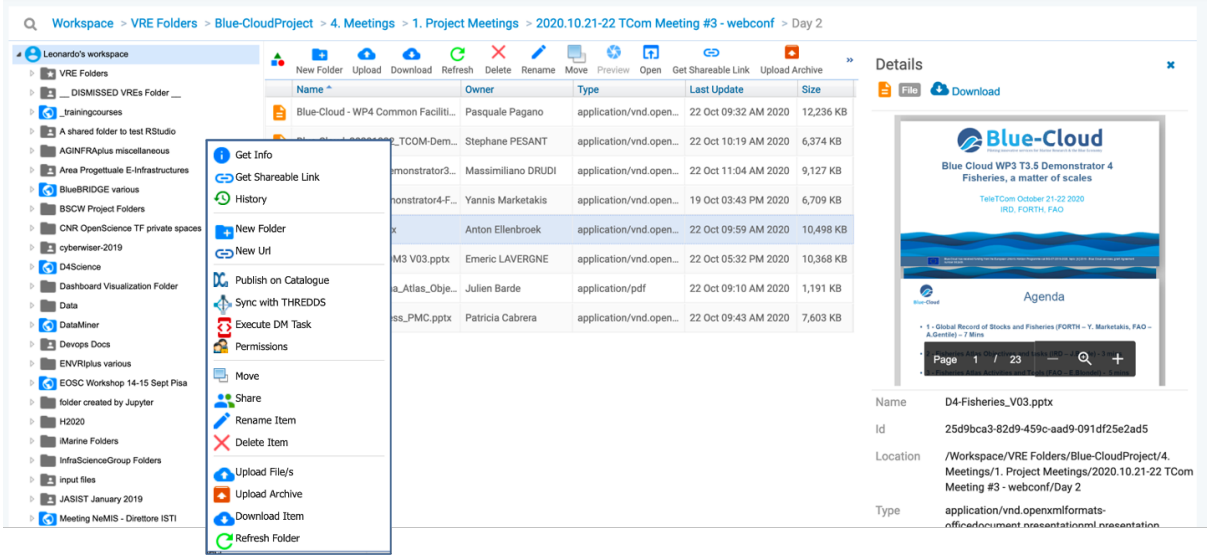

Figure 6. The Workspace graphical user interface

The workspace content is organised in folders. The types of a folder supported by the Blue-Cloud VRE have been largely extended in this period to support different scenarios requiring tailored storage solutions.

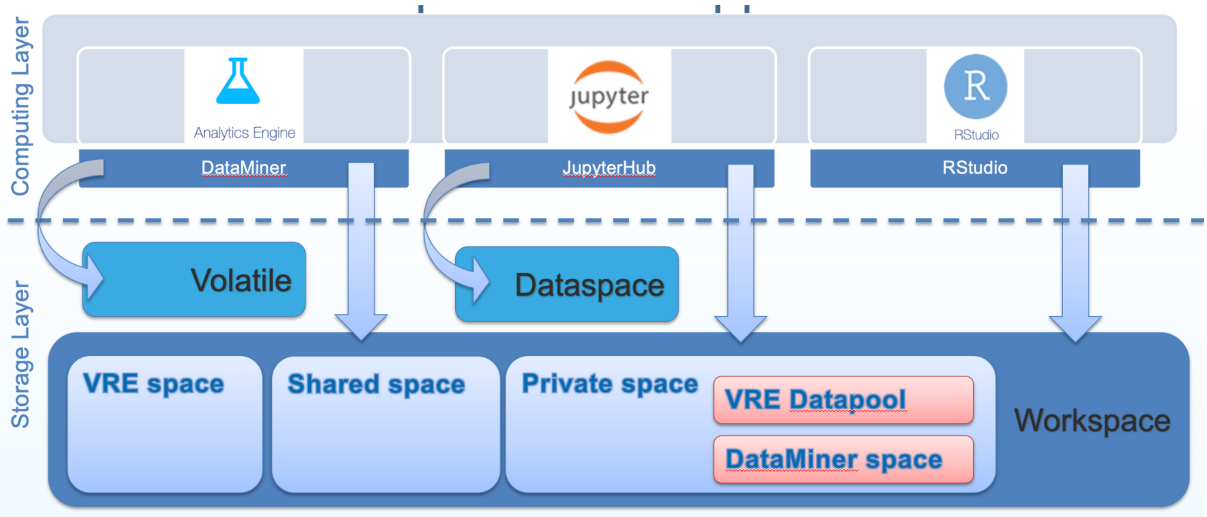

Figure 6.1. The Workspace tailored spaces and the overall storage volumes

The Volatile space is a temporary and fast space where to store transient files. It is currently exploited by the Analytics Engine.

The Dataspace is a very large (hundreds of Terabytes) space connected to the JupyterHub cluster that is local to the notebook computational environment.

The Workspace is a large, fault-tolerant and secure storage volume hosting several spaces:

- VRE space: a storage area paired with a VLab and available to all VLab members;
- Shared space: a storage area created by a user and shared with other users;
- Private space: a storage area reserved to any user. It includes the DataMiner space where all provenance information about each job executed through the DataMiner engine are automatically stored and the VRE Datapool where all datasets are accessed via the Data Discovery & Access service.

For every workspace element, be it a folder or an item, it is possible to see some metadata (e.g. a name, a description, creation and update dates) and possibly a preview of its content. Moreover, it is possible to create links (as URI's) pointing to it. For folders, it is possible to create links enabling those who receive it to access the folder in "guest mode", and no login will be requested. The contextual menu offers shortcuts enabling users to perform some actions on it, e.g. to publish the item into the catalogue, to execute a data miner process passing the item as input.

Figure 6.2. presents a comparison of the different and tailored storage options accessible through the Blue-Cloud VRE. It is a work in progress activity the extension of the Workspace to include the access and management to the Dataspace and the Volatile storage. This extension will preserve the main characteristics of those spaces while simplifying their exploitation and the overall user experience.

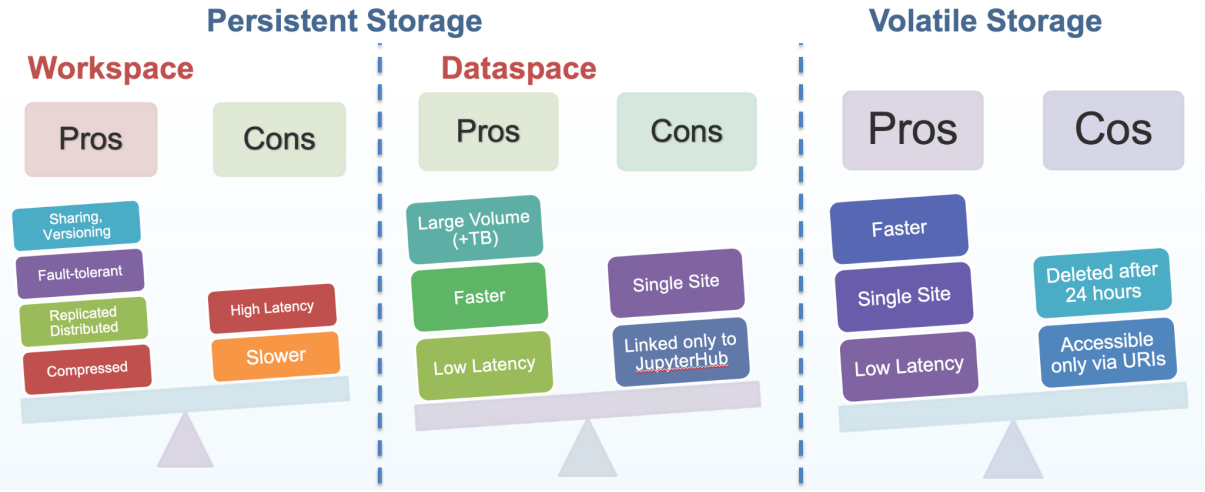

Figure 6.2. Comparison of the different and tailored Storage volumes accessible via the Blue-Cloud VRE

4.2 Social Networking

The social networking component provides its users with facilities for communicating and cooperating by exploiting social networking practices (e.g. rich posts, likes, hashtags, and mentions).

Figure 7 shows the software architecture of the social networking collaborative platform. The Social Networking collaborative platform relies on the Social Networking Engine, an Apache Cassandra[12] database for storing social networking related data and on Elasticsearch[13] for the retrieval of social networking data. The Engine exposes its facilities through an HTTP REST Interface and comprises two services: *(i)* the Social Networking Service that efficiently stores and accesses social networking data (Posts, Comments, Notifications, etc.) in the underlying Cassandra Cluster, and *(ii)* the Social Networking Indexer Service that builds Elasticsearch indices to perform search operations over the social networking data.

[12] https://cassandra.apache.org

[13] https://www.elastic.co/elasticsearch

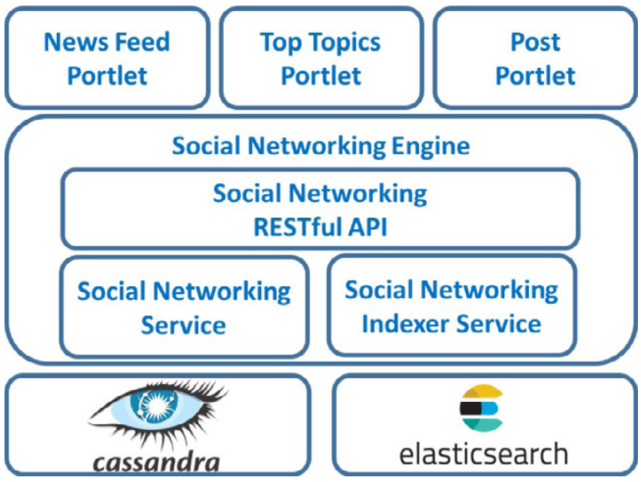

Figure 7. The architecture of the social networking collaborative platform

4.2.1 Service

As mentioned above, the Social Networking Service stores and accesses social networking data (Posts, Comments, Notifications, etc.) in the Cassandra database. The service relies on a core Java library called gCube Social Networking Library (SNL). SNL offers methods for posts creation, retrieval and removal as well as comments creation, retrieval and removal, and manages notifications etc. The Social Networking Service exposes a subset of the functionalities over SSL protocol in a standard, reliable and secure way.

The Social Networking service and the Social Networking Library pre-date Blue-Cloud, during this first reporting period, they have been re-designed to adapt to evolving technologies. Detailed information about the service and the library can be found at the URLs below:

- https://wiki.gcube-system.org/gcube/Social_Networking_Library
- https://wiki.gcube-system.org/gcube/Social_Networking_Service

4.2.2 Indexer

The Social Networking Indexer Service (also Social Networking Data Discovery service) offers full-text search capabilities over the social networking data in the Blue-Cloud. The Social Networking Indexer Service pre-dates Blue-Cloud, during the first reporting period it has been re-designed to adapt to evolving technologies

The full-text search is enabled by Elasticsearch, a distributed, RESTful search and analytics engine capable of addressing a growing number of use cases. It is based on the Apache Lucene software library. It runs over one or more cluster nodes and is reachable over http(s) protocol. ElasticSearch allows organising documents in one or more indices/types according to their schema, which can be defined in JSON format.

The main goal of the Social Networking Indexer Service is to let users quickly search over Blue Cloud's (potentially huge) amount of data, taking into account their profile: a user is allowed to search only the data of the VLabs in which she is registered. In order to do that, a Java client library, the social-data-indexing-common, receives the query submitted from the users and returns the list of posts belonging to the user's VLabs, if any, sorted according to a score; and a Java gCube Smart Executor Plugin, which triggers the indexing of data on the service when necessary, social-data-indexer-se-plugin, which has been re-designed to adapt to evolving technologies.

The Social Networking Indexer Service pre-dates Blue-Cloud, during this first period, it has been re-designed to adapt to the evolving technologies. Detailed information about the service and the library can be found at the URLs below:

- https://gcube.wiki.gcube-system.org/gcube/Social_Networking_Data_Discovery

4.2.3 User Interfaces

Figure 8 shows the user interface of the social networking area. Starting from the top right we can see the "Share Updates" Portlet (Post Portlet in the Architecture Figure) allowing members to post, just below, the News Feed portlet which collects the posts and shows them in reverse chronological order allowing members to comment on them. On the left, starting from the top we find some statistics on the usage of the platform, the workspace shared folder content of the VLab and the Top trending topics, showing them using the hashtag prefix. This is indeed the area VLab users rely on to communicate with their VLab co-workers and be informed on others achievements, discussions and opinions. It resembles a social networking environment with posts, tags, mentions, comments and reactions, yet its integration with the rest makes it a powerful and flexible communication channel for researchers.

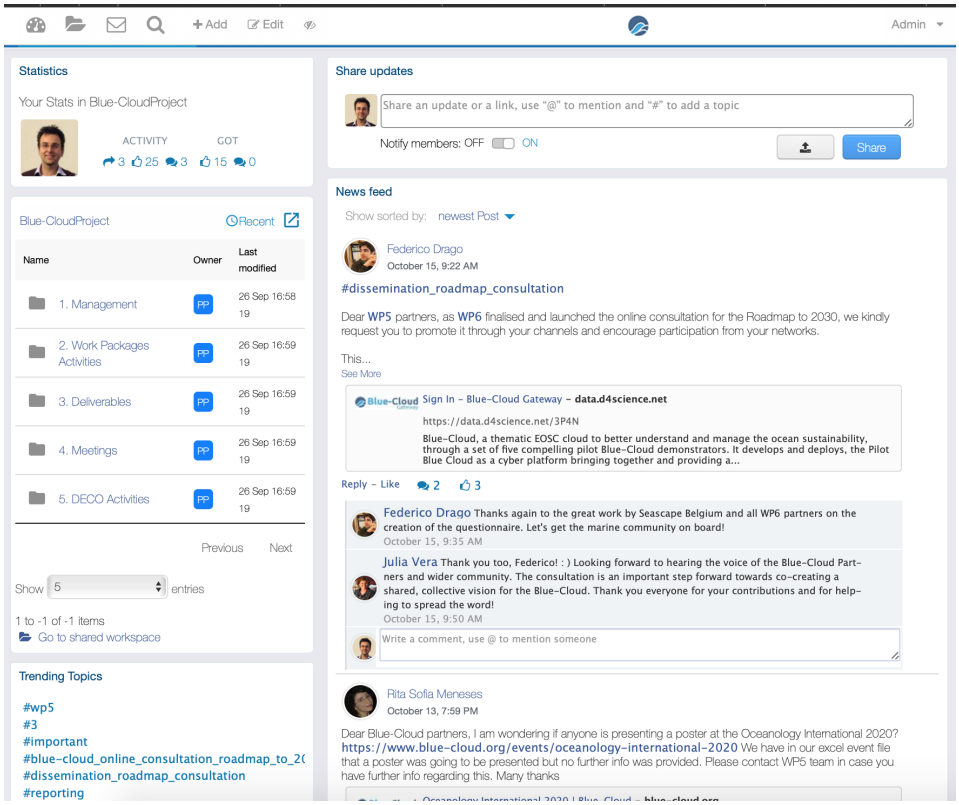

Figure 8. The social networking user interface: a screenshot

5. Analytics Framework Components

The Analytics Framework includes a set of services and components for performing data processing and mining on information sets.

As presented in Schaap et al. (2020) Deliverable D2.6, the Blue Cloud VRE Analytics Framework was designed to include the DataMiner System, the Software and Algorithms Importer, and the Smart Executor System. This set of services is complemented by integrating RStudio and by supporting the integration of RShiny applications. To meet the Blue community requirements of the Blue Cloud demonstrators, the Blue Cloud VRE extended this set of systems, components, and tools by adding support for dynamic, interactive notebooks, via JupyterHub, and the support for community-specific applications delivered as a Docker container.

These enhanced capabilities of the Analytics Framework allow users to do science by selecting the most appropriate and familiar tool to implement the analytical methods and data processing. Users' analytical methods can use any programming language (R, Java, Python, Fortran, Octave, etc.) and then they can select from several integration patterns of the Blue Cloud VRE.

The Blue-Cloud Lab is set-up as a typical VLab that provides access to the overall components of the Analytics Framework in a user-friendly and easy-to-use way.

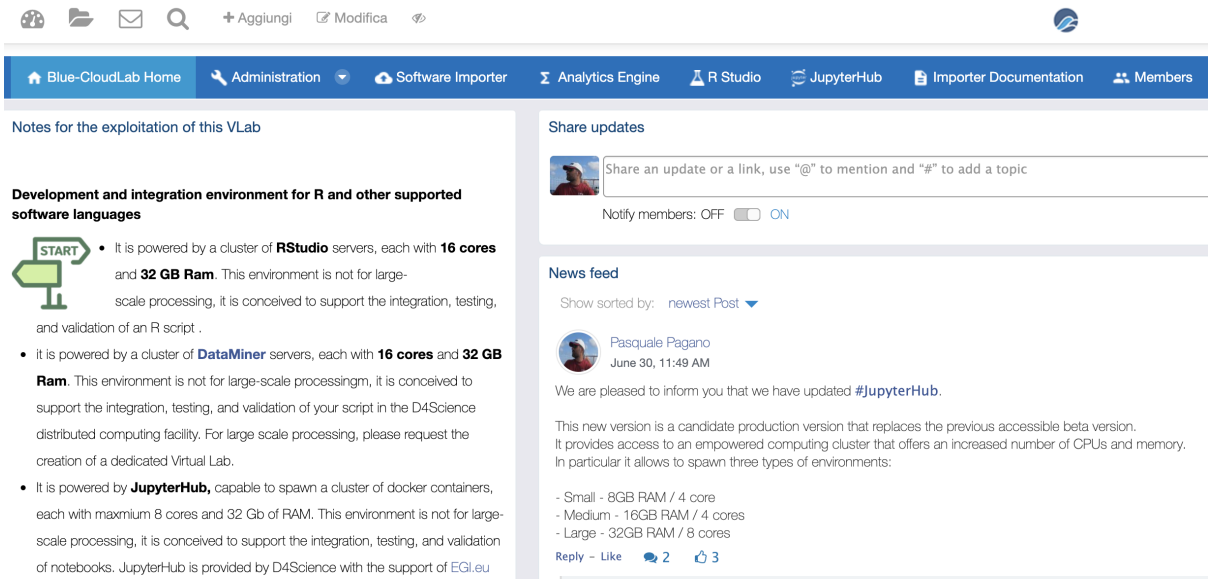

Figure 9. Virtual Laboratory typical graphical user interface

DataMiner, RStudio, and JupyterHub are all integrated with the Blue Cloud storage allowing to download, upload, remove, add and list files, define access rights to files, and allowing private, public, or shared (group-based) access.

For entire applications, Docker containers (with for instance a RShiny application) can be deployed in different ways:

- If a public container is already available in Docker Hub or any other public container registry, it is sufficient to report the coordinates of that container;
- If a public container is not yet available
    - A build of a public image can be requested. It must be accessible from the D4Science Jenkins instance so that the process can be automated. The result container image will be uploaded into Docker Hub and deployed into the cluster.

- A build of a private image can also be requested. Also, in this case, it must be accessible from the D4Science Jenkins instance so that the process can be automated. The resulting container image will be uploaded into the D4Science's private registry and deployed into the cluster.

The Blue Cloud VRE Analytics Framework provides

- Support for the execution of analytical methods on multi-core computational nodes (DataMiner, Smart Executor, RStudio, JupyterHub, and Docker);
- Support for multi-tenancy and concurrent access (DataMiner, Smart Executor, RStudio, JupyterHub, and Docker);
- Support for Auditing (DataMiner)
- Automatic distribution of the execution of analytical methods on sets of computing nodes (DataMiner, Docker Swarm);
- Automatically transfers control to a duplicate computational node when faults or failures are detected (DataMiner, Docker Swarm);
- Support for scheduled and repeated execution (Smart Executor);
- Automatic generation of provenance information to enable reproducibility and repeatability of the computed results (DataMiner);
- Support for the standard Web Processing Service (WPS) protocol (DataMiner).

5.1 Software Importer and DataMiner

The Software and Algorithms Importer (SAI) is an interface allowing any user to easily and quickly import scripts into DataMiner. DataMiner, in turn, publishes these scripts as-a-Service and manages multi-tenancy and concurrency. Additionally, it allows scientists to update their scripts without following long software re-deploying procedures each time. In summary, SAI produces processes that run on the Blue-Cloud VRE Cloud computing platform and are accessible via the WPS standard.

In order to import a script, three main passages are required:

1. Indicate Input, Output and types of the main script orchestrating the process;
2. Create the Software: this operation packages the script and prepares it for the execution on the computing platform. It has to be used each time either the interface (I/O) or the required dependencies have been changed;
3. Publish the Software: this operation enables the execution of the script on the computing platform in the context of the virtual laboratory where it has been imported.

Additionally, the Repackage function can be used to upgrade a published script that has been evolved without changing neither the I/O nor its dependencies.

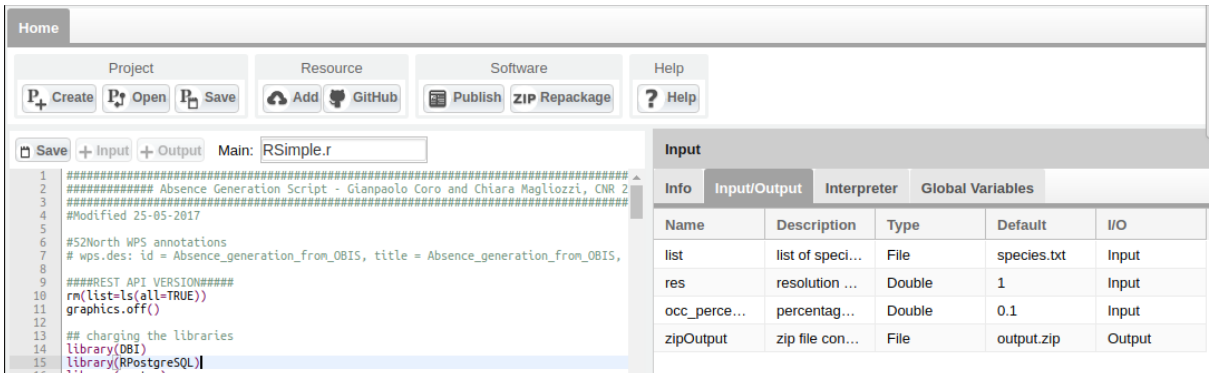

Figure 10. Blue-Cloud VRE Software and Algorithms Importer Interface

Once imported, scripts become exploitable through the DataMiner component.

It offers a Web GUI organised in three main areas: Data Space, Execution Space, Computations Space.

The DataMiner "Data Space" (See Figure 11) allows accessing, reusing, and downloading the set of input and output datasets respectively used and generated in one or more execution and enriched with business metadata reporting the computational method used for its generation, its execution environment (including the input parameters), the virtual laboratory where it was generated, and the date of generation.

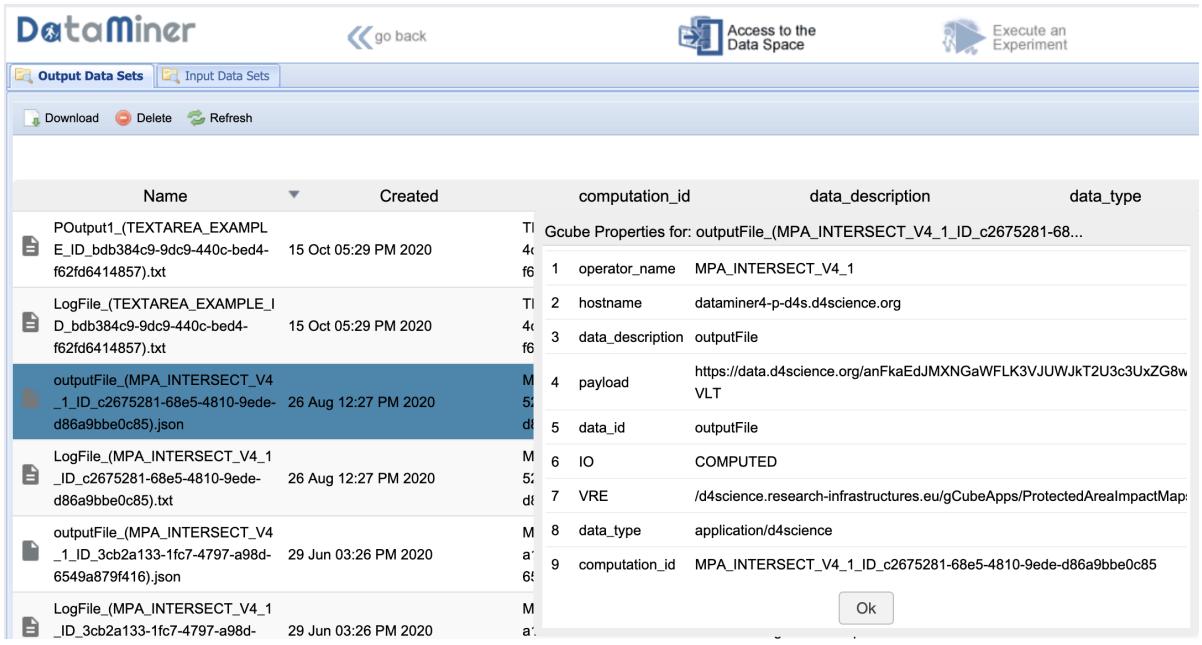

Figure 11. Blue-Cloud VRE Analytics Data Space Interface

The Execution Space presents two panels:

- On the left panel, the GUI presents the list of computational methods available in the virtual laboratory, which are semantically categorised (the category is indicated through SAI). For each method, the interface calls the WPS DescribeProcess operation to get the descriptions of the inputs and outputs.
- On the right panel, the GUI presents a form allowing to specify the input parameters of the selected computational method. Input data can be selected from the Workspace clearly.

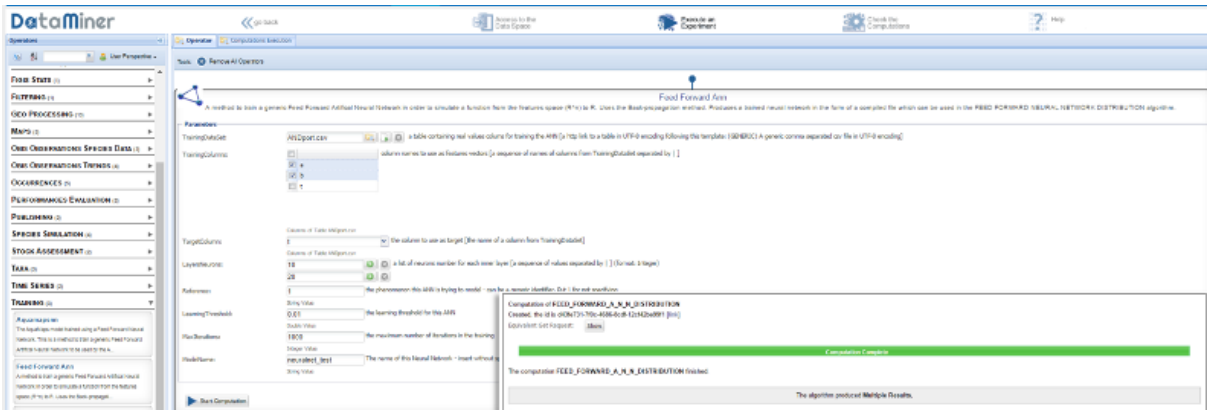

Figure 12. Blue-Cloud VRE Execution Space Interface

The Computations Space represents an important added-value since it reports a summary sheet of the provenance of the execution either performed by the user or shared with him. From this same space, the computation can be also re-submitted. In this case, the Prov-O XML information associated with the computation is used to rebuild the computation request with the same parameters and the execution can be re-submitted.

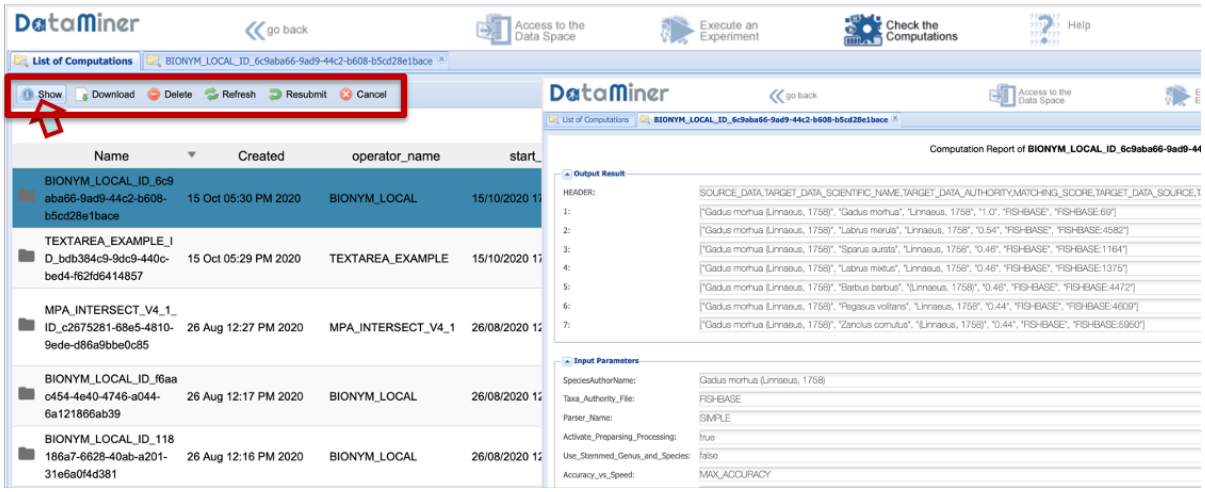

Figure 13. Blue-Cloud VRE Computation Space Interface

The DataMiner component can rely on two distinct distributed computing clusters: the master and the worker clusters as reported in Schaap et al. (2021) Deliverable D2.7. The working cluster has been configured and now it is operated to serve a different application scenario.

As presented in Figure 13.1, the Worker cluster has been configured to assign all the resources to a single job in exclusive mode and to exploit exclusively the Volatile space. This configuration guarantees increased performance and predictable job execution time. A job can be either executed directly in the worker cluster or it can be spawned by another job, playing the role of an orchestrator, running in the master cluster.

This configuration better supports the analysis of large collections of datasets and it better fits the needs of the Blue-Cloud community.

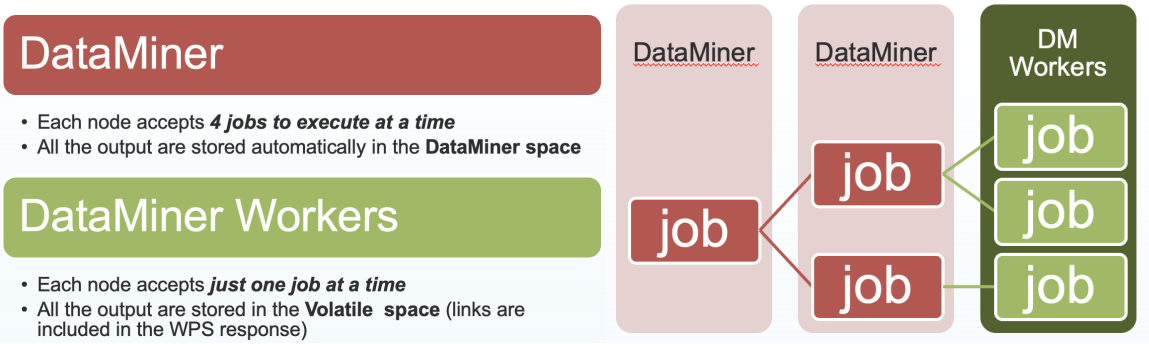

Figure 13.1. DataMiner master and worker clusters characteristics and typical exploitation scenarios

5.2 Smart Executor (new service)

The SmartExecutor service allows users to execute tasks and monitor their execution status. Any task can be either scheduled, or repeated periodically, or activated upon request. A task has to be implemented as a plugin of the service while its usage and exploitation can be performed using the SmartExecutor REST API.

It is typically used to periodically invoke a computational method imported into the DataMiner. A common case is the monthly aggregation of raw data that is gathered daily. In this example, the data aggregation is implemented via a computational method imported in DataMiner; the monthly execution of it is instead realized by a task of the Smart Executor service. The combination of the two services will allow either a single user or all the virtual laboratory users to access the collection of aggregated data through the workspace, and is a useful pattern to manage confidential data.

5.3 RStudio

The RStudio allows performing online statistical analyses with R.

The Blue-Cloud VRE makes the RStudio Application accessible on the D4Science infrastructure ensuring its operation and orchestration in a cluster. The cluster is composed of multiple hosts, each of which is assigned in exclusive mode to a user for an entire online session. At the end of the session, all the content stored in that host will be removed by the D4Science Infrastructure. All data, scripts, and other resources are thus secure, and not accessible to others. The user can persist the R-session into the private Workspace that is accessible through the RStudio Application.

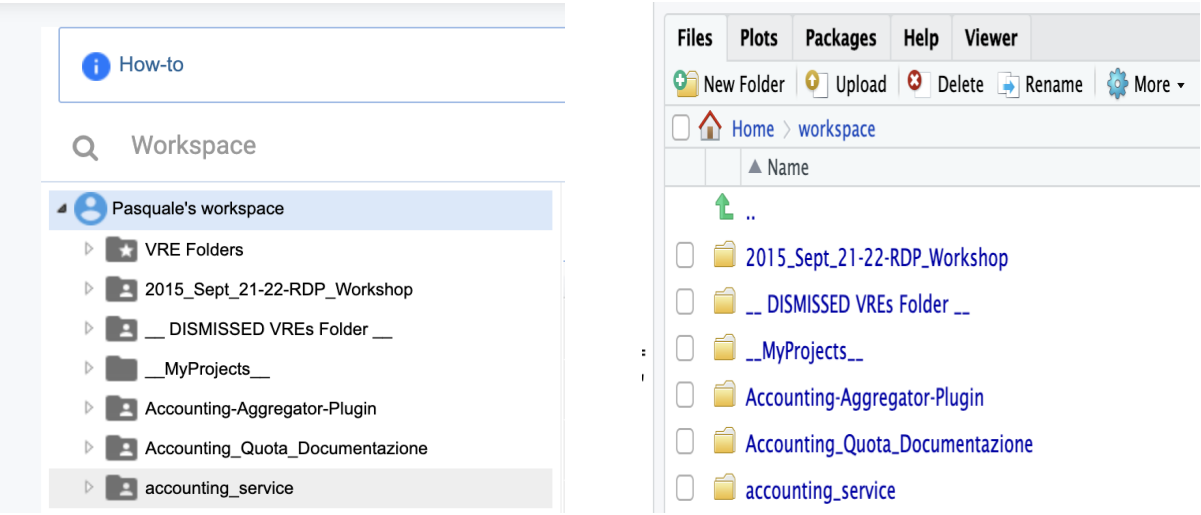

Figure 14. Blue-Cloud VRE Workspace and RStudio Workspace

5.4 JupyterHub

JupyterHub is a web-based interactive development environment for Jupyter notebooks, code, and data. It allows users to configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

The Blue-Cloud VRE makes the JupyterHub accessible on the D4Science infrastructure ensuring its operation and orchestration in a cluster. The cluster is composed of multiple hosts, each of which is assigned in exclusive mode to a user for an entire online session. At present, the cluster supports 80 concurrent users (at 8 GB RAM; Cores per user: 4/8; Memory per user (GB): 8/16/32; Storage per user (GB): 10GB; Allocation type: pledged).

JupyterHub uses ephemeral storage and all the content stored in that host is removed at the end of the session. All the data, scripts and other resources that the user needs to persist have to be stored into the Workspace that is accessible through JupyterHub.

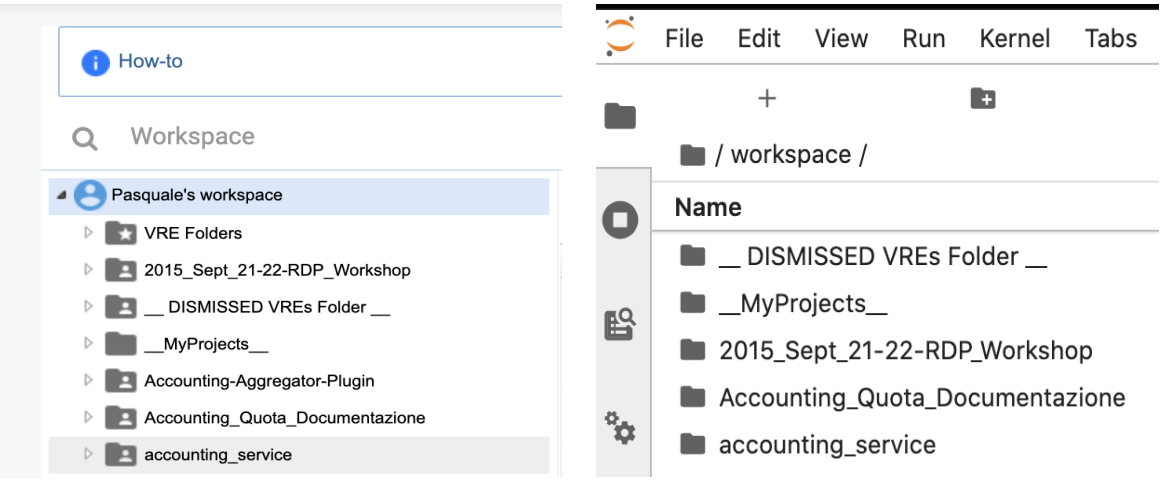

Figure 15. Blue-Cloud VRE Workspace and JupyterHub Workspace

5.5 ShinyProxy and Docker

ShinyProxy allows deploying Shiny apps in an enterprise context with no limits on the concurrent usage of them. Shiny is an R package that makes it easy to build interactive web apps straight from R. A Shiny app can be embedded in R Markdown documents or used to build dashboards by combining the computational power of R with the interactivity of the modern web.

When deploying a Shiny app with ShinyProxy, the application is simply bundled as an R package and installed into a Docker image. Every time a user runs an application, a container spins up and serves the application. This has numerous advantages:

- fully isolated environment per session;
- plug and play different docker images (even with different R versions or different Shiny versions);
- control on memory and CPU usage via the Docker API; and
- monitoring and debugging using standard Docker tooling.

Blue-Cloud VRE uses Docker Swarm to run multiple Docker containers across a cluster of virtual machines. Docker Swarm defines a manager container that runs on a virtual machine that manages the environment, deploys containers to the various agents, and reports the container status and deployment information for the cluster. The manager is the primary interface into Docker. Agents are "docker machines" running on virtual machines that register themselves with the manager and run Docker containers. When the client sends a request to the manager to start a container, the manager finds an available agent to run it. It uses a least-utilized algorithm to ensure that the agent running the least number of containers will run the newly requested container.

Routing external traffic into the cluster, load balancing across replicas, and DNS service discovery are a few capabilities that require an additional layer. The Blue-Cloud VRE exploits the HAProxy load balancer to accomplish those capabilities.

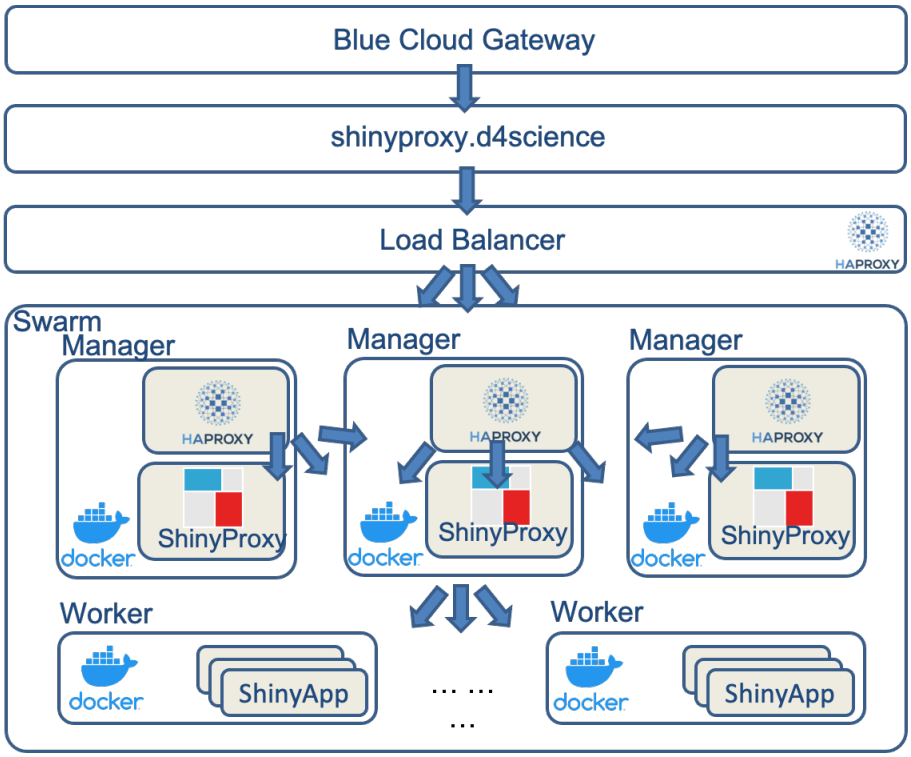

Figure 16. Blue-Cloud VRE cluster supporting Shiny and any other Docker app

There are three ways to exploit HAProxy:

- one HAProxy container: Swarm's ingress routing mesh forward clients' requests to it;
- one HAProxy container: HAProxy receives clients' requests directly without using the ingress routing mesh;
- Create a replica of HAProxy on each node: each will receive clients' requests directly.

Blue-Cloud VRE adopted the third approach guaranteeing the capacity to handle more requests because there are more running instances of HAProxy. In order to make this deployment mode efficient, the VRE uses an external L4 load balancer, still using HAProxy in TCP mode, in front of the Swarm cluster. This allows spreading the load across the different HAProxy containers.

This architecture is also used to deploy any other application delivered as a Docker container.

5.6 Docker and DataMiner

A Docker image represents an easy-way to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files and thus they may contribute to simplifying the configuration and operation of the Blue-Cloud VRE infrastructure.

The Blue-Cloud VRE delivers an additional solution allowing to exploit Docker while preserving the main features of the VRE: replicability, reusability, sharing, accounting of the execution will all be preserved by following and exploiting the Docker Image Executor algorithm.

The Docker Image Executor algorithm allows its users to retrieve and run an image in the presented Swarm cluster, from a public Docker repository, e.g. DockerHub. Each execution will be accounted

for and presented in the Blue-Cloud VRE Computation Space Interface. A screenshot of the Docker Image Executor method is in Figure 17.

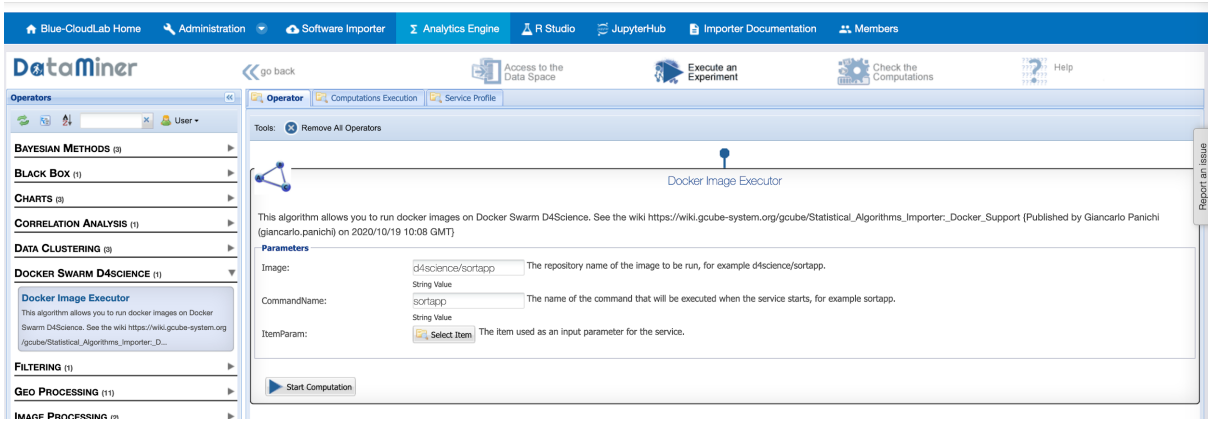

Figure 17. The DataMiner Docker Image Executor Algorithm

6. Publishing Framework Components (new services)

The Publishing Framework includes a set of services and components enabling their users to document and make "public" any artifact worth being published, i.e. made available online. It comprises two major services:

- The VRE Data Catalogue Service;
- The Spatial Data Catalogue Service;

6.1 The VRE Data Catalogue Service

The VRE Data Catalogue service is a catalogue service built on an open-source technology for data catalogues (CKAN ckan.org) but extended to (a) be integrated with D4Science services and (b) support a rich, community-defined and extensible set of catalogue item typologies.

The architecture of the service is depicted in Figure 18.

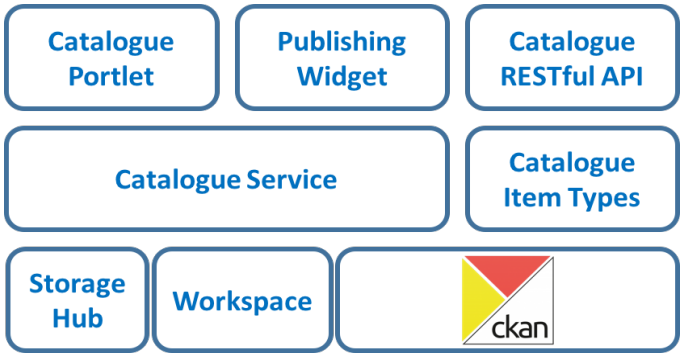

Figure 18. Catalogue Service Architecture

The Catalogue Service is the core component called to implement the business logic of the overall service. The catalogue service interacts with a component called to support the creation of catalogue item typologies, i.e. specifications characterising items in terms of attributes, controlled vocabulary, etc. The data model supported by the catalogue is in Figure 19.

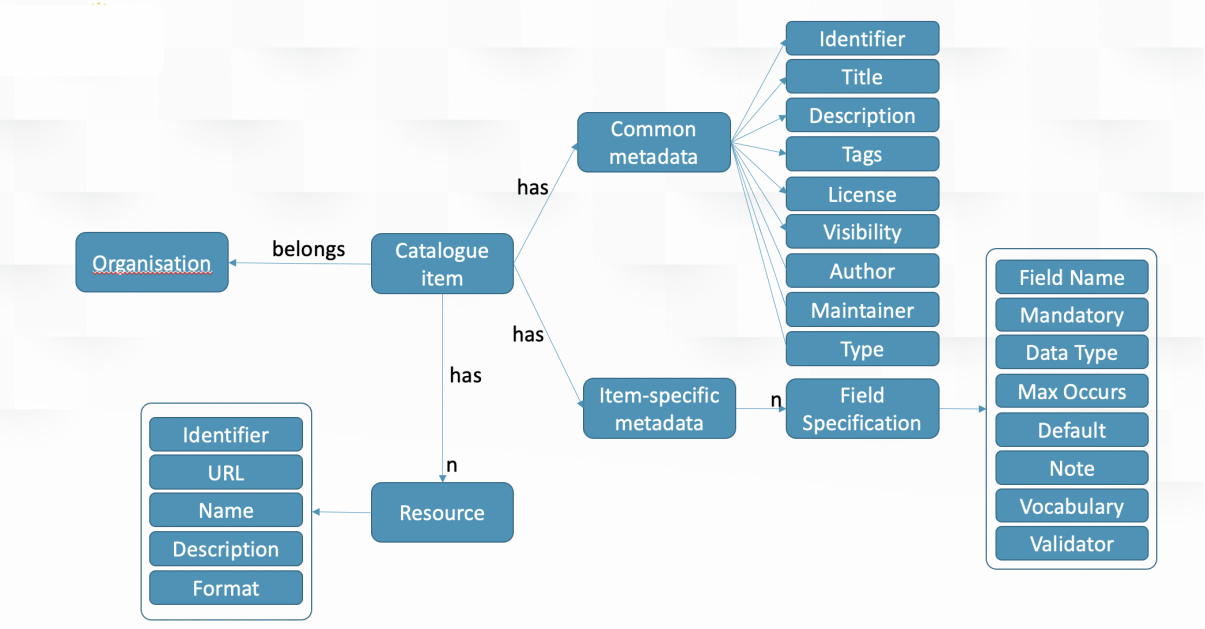

Figure 19. Catalogue Service data model

According to this data model:

- every catalogue item belongs to one (and only one) ***organisation*** representing the context / authority responsible for the publishing of the item. Organisations are usually paired with VLabs for the items stemming from them or with other existing contexts where the items pre-exist VLabs;
- every catalogue item is characterised by a set of ***common metadata*** including a unique identifier, a title, a description, a list of tags (e.g. keywords, subjects), a licence, visibility (whether the item is publicly available or visible only to the members of a VLab), an author, a maintainer, and a type;
- every catalogue item may have a list of ***item-specific metadata*** depending on the type. In practice, this is a list of fields each having a name, a mandatory directive (whether the field is mandatory or optional), a type (e.g. string, number, spatial extent), a max occur directive (whether the field can be instantiated one time only or many times), a default value, a descriptive note helping to understand the intended meaning of the field, a controlled vocabulary (if any) of allowed values to use to compile the field, and a validator (if any) to check the inserted value adherence to specific validation rules.
- every catalogue item can be equipped with a set of ***resources***, i.e. catalogue item constituents representing the real item payload or part of it. Resources are characterised by their identifier, a name, a description, a format and, most importantly, the URL pointing to the content.

There are two main routes to use to populate the catalogue with contents: by harvesters from existing data sources and by explicit publishing via the Web GUI or the Web API (a.k.a. the gCat REST Service). Figure 20 clarifies that for what regards harvesters the catalogue builds upon standards including DCAT, CSW, OAI-PMH as well as CKAN standard APIs.

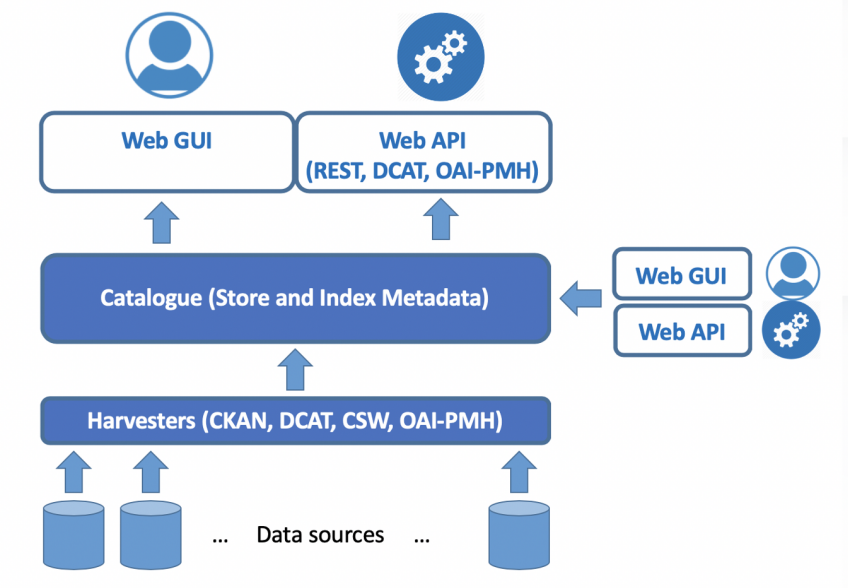

Figure 20. Catalogue Service: Feeding and Consumption options

For what regards the consumption options, the catalogue offers both (i) programmatic access for "machines" to access its contents via a proprietary REST API (a.k.a. the gCat REST Service) and standards like DCAT and OAI-PMH and (ii) human-oriented access via a dedicated Web-based GUI to search the catalogue contents via queries, to browse via a set of faceted options and to access and visualize every item.

6.1.1 Zenodo

The catalogue is equipped with a facility to transfer items to Zenodo and establish a mutual link between the two items, i.e. to document the fact that the Zenodo item results from the Catalogue item (by instantiating a related identifier with "Compiled by" containing the URL of the catalogue item) and that a Zenodo item associated with the catalogue entry exists (by annotation the catalogue entry with a related Identifier pointing to the Zenodo item DOI).

This facility is made available by the Catalogue GUI of a single item via the Upload to Zenodo menu option (see Figure 21 below).

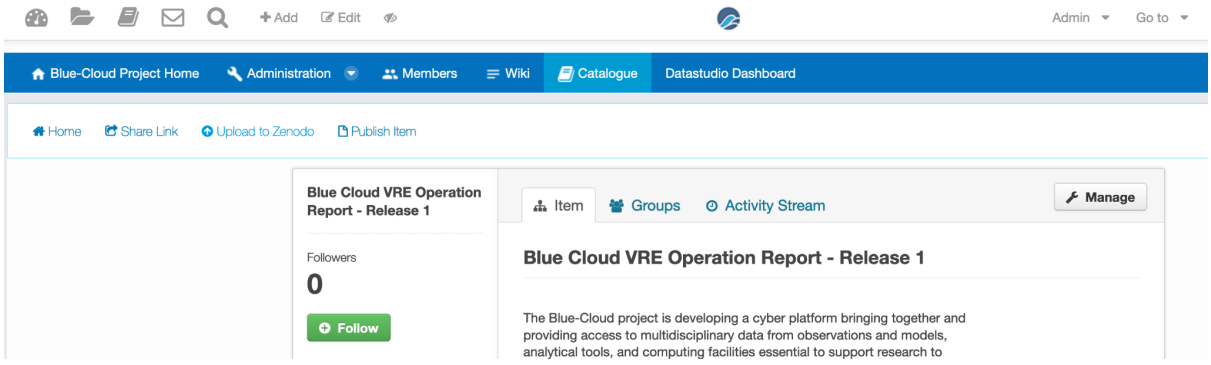

Figure 21. Upload to Zenodo Data Repository

When an authorized user clicks on the Upload to Zenodo option, he/she is provided with a form allowing to revise and complement the metadata and to carefully select the files to be transferred to Zenodo.

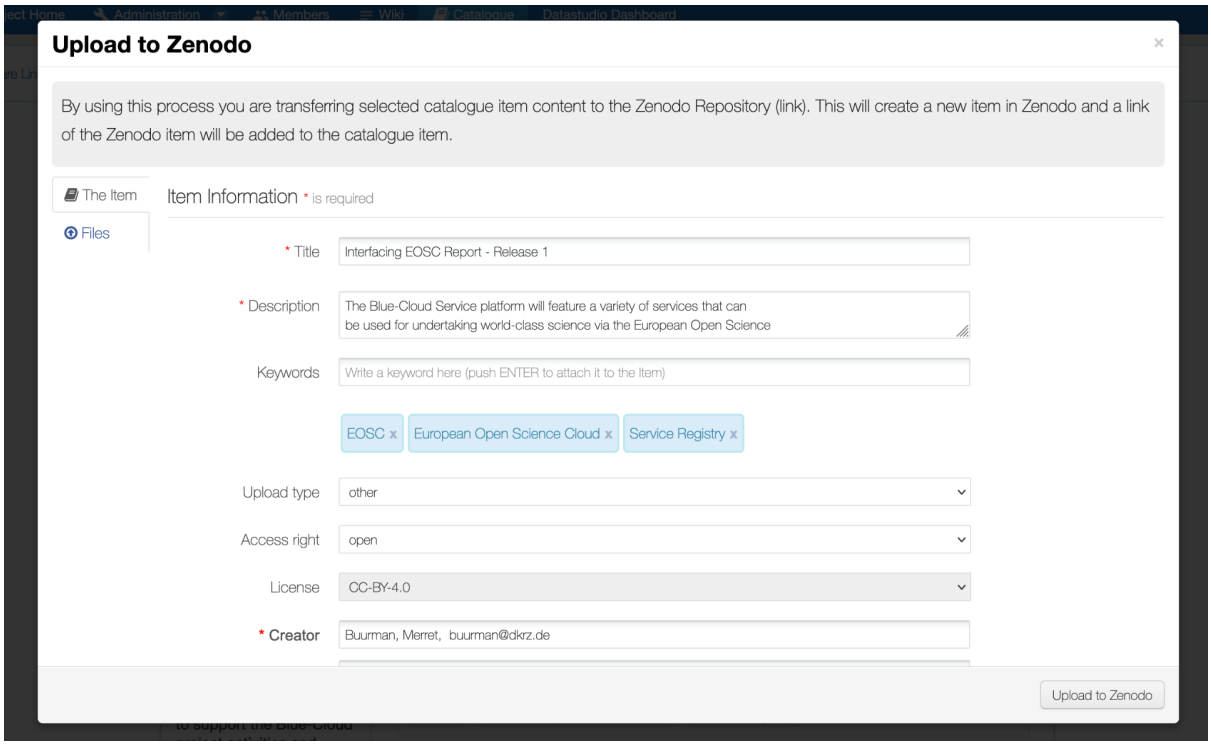

Figure 22. Upload to Zenodo: Form

6.2 The Spatial Data Catalogue

To support the publishing of geospatial data, the BlueCloud VRE offers an organised set of technologies realising a Spatial Data Infrastructure including a set of repositories and an unifying catalogue (see Fig. 23).

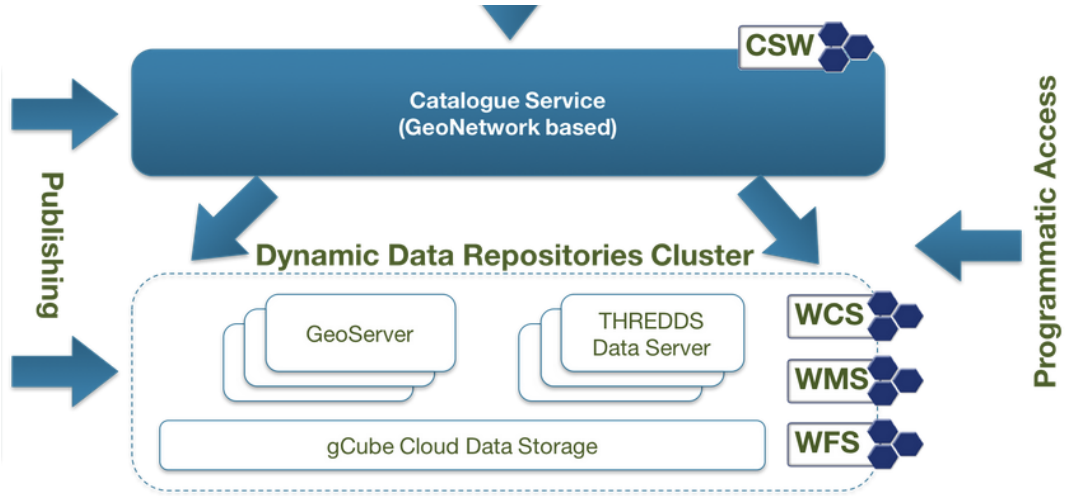

Figure 23. Spatial Data Catalogue

Thanks to the Data Catalogue harvesting facilities, the Spatial Data Catalogue content can be collected and made seamlessly searchable with the rest of items.

Various approaches and strategies could be exploited to populate the repositories and make them searchable via the unifying catalogue varying with respect to programming skills needed and

technological expertise. To simplify this, a user-friendly approach allowing a user to deposit geospatial data into a repository based on THREDDS Data Server (TDS) directly from the workspace has been developed (see Figure 24).

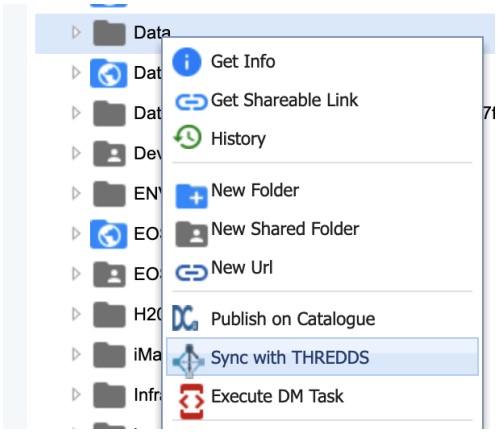

Figure 24. Sync with THREDDS workspace menu option

This facility allows an authorized user to create a catalogue on TDS out of a workspace folder containing the data. The user is requested to specify the directives driving catalogue creation, e.g. specifying the context where the catalogue is going to be hosted, whether an existing catalogue is going to be reused or a new one should be created, etc. This associates a configuration directive to the folder driving the tasks of synchronizing the catalogue with workspace content and monitoring the overall process.

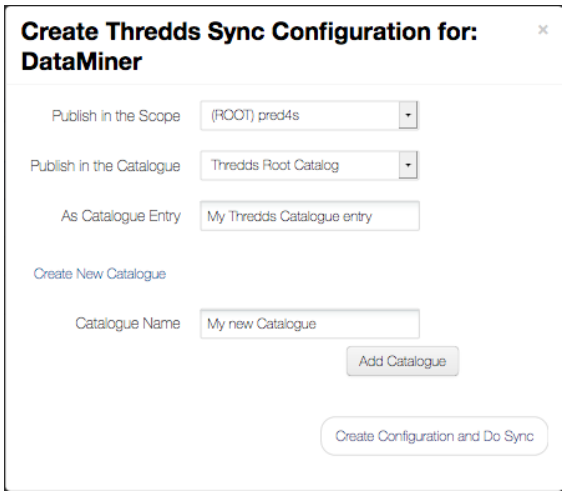

Figure 25. Sync with THREDDS: Configuration Creation

7. Bridging Systems

Some VRE Common Facilities are realised by interfacing with external systems. In particular, this section describes the solution enacting users to transferring datasets of interest discovered by the Data Discovery & Access System (Schaap et al 2021 Deliverable D2.7) directly into the workspace for future uses (cf. Sec. 7.1) and the solution developed to facilitate the exploitation of the WEkEO Harmonised Data Access (HDA) API (cf. Sec. 7.2).

7.1 VRE Integration with Data Discovery and Access System

As depicted in Figure 26, a new VRE Common Facility available for Blue-Cloud users, that was developed during the period, is the VRE integration solution (framed in red below) complementing the Data Discovery & Access Service (DDAS). It allows users to transfer the data coming from DDAS transparently into their private VRE Workspace (cf. sec. 4.1).

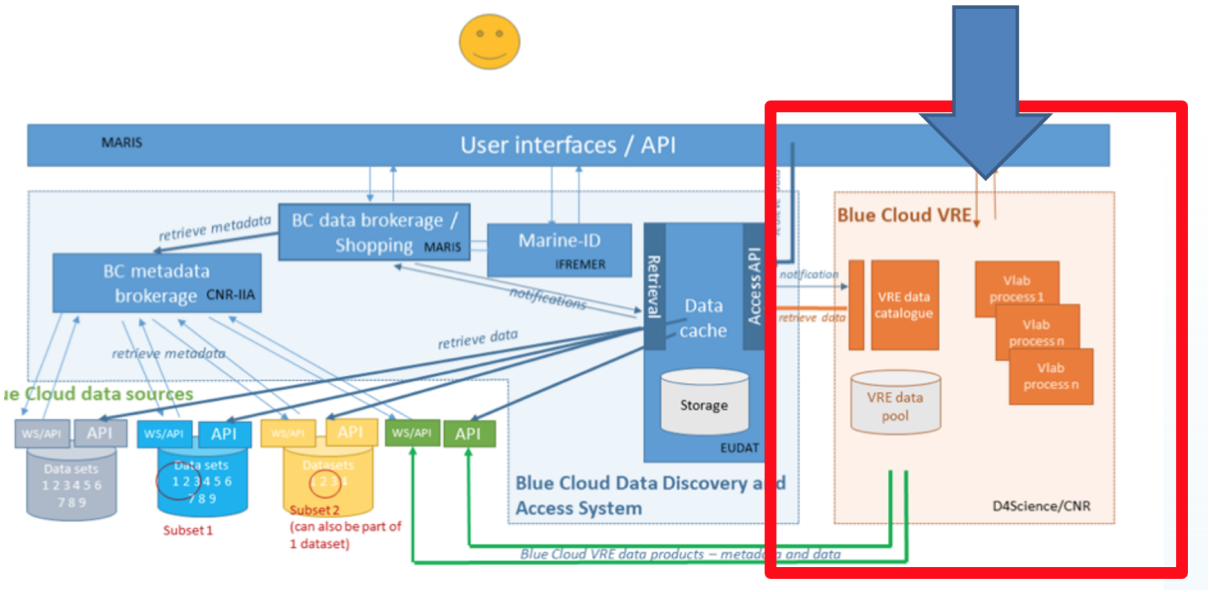

Figure 26. Architecture of Data Discovery & Access System and the role of Blue-Cloud VRE

From an end user point of view transferring the selected data from any DDAS standing order to the VRE workspace can be achieved by simply pushing a button (Push order to VRE) directly on the DDAS Web Interface (see Figure 27) and waiting for the transfer to be completed, the user is informed with a dedicated email as soon as the data transfer operation concludes.

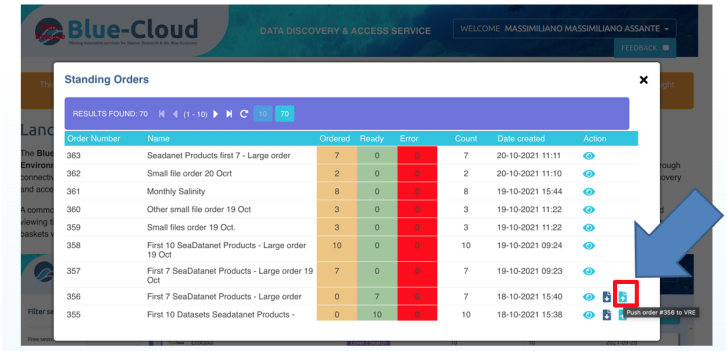

Figure 27. Data Discovery & Access System standing orders

The implementation of this facility required different services to coordinate and inter-operate though different standards and protocols, a complex combination of different activities which involved authorisation, calling of VRE Workspace APIs, parsing the JSON of a transfer request sent from DDAS and starting the listed download operations in parallel. In addition, the majority of these activities needed to be handled via HTTP calls and, as a consequence, were subject to failures which needed to be handled.

We decided to use the Orchestrator (cf. Sec. 3.3) and its workflows feature to implement the DDAS to VRE data transfer because it represented a well suited solution to all of the challenges described. The Orchestrator receives an authorised data transfer request with all the information in a descriptor, required for sending email notification to the user and outcome reports back to DDAS. This descriptor, besides the order metadata, contains a list of download links, including size for each download, that are the transfer operations to be performed in parallel. Hence, the Orchestrator engine executes a dedicated workflow, called *da_cache_to_shub* (cf. Table 2), to perform the transfer tasks and save the downloads on the user private workspace area.

It is important to note that the transfer operations performed by *da_cache_to_shub* workflow do not use any temporary intermediate storage by design. This has been achieved by developing a new worker called PyRestBridge (cf. Table 1), that exploits the possibilities offered by modern HTTP client libraries in terms of streaming and optimization. The implementation of PyRestBridge is based on an extremely optimised usage of the *requests* library[14].

Figure 28.1, 28.2 and 28.3 show a visual description of the three subsequent phases of *da_cache_to_shub* workflow execution, respectively. A first phase where initialisation and preliminary checks on the user's workspace is performed (Figure 28.1 and Figure 28.2) and a third phase with the transfer task (Figure 28.3), that include the reporting tasks (email notification to the user and outcome reports back to DDAS).

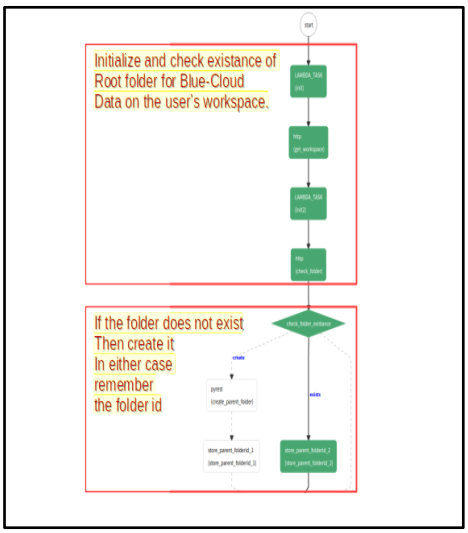

Figure 28.1: da_cache_to_shub workflow - Initialization and global folder check or creation.

[14] https://docs.python-requests.org/

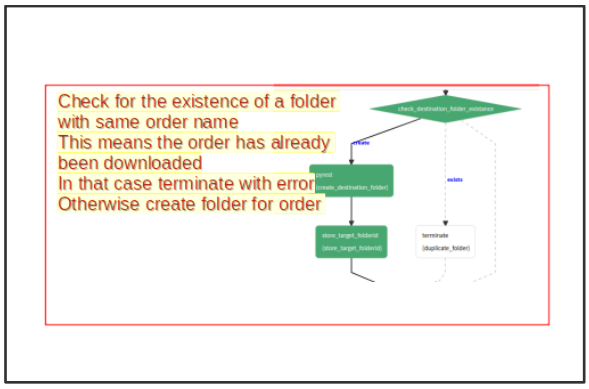

Figure 28.2: da_cache_to_shub workflow - Order Folder check.

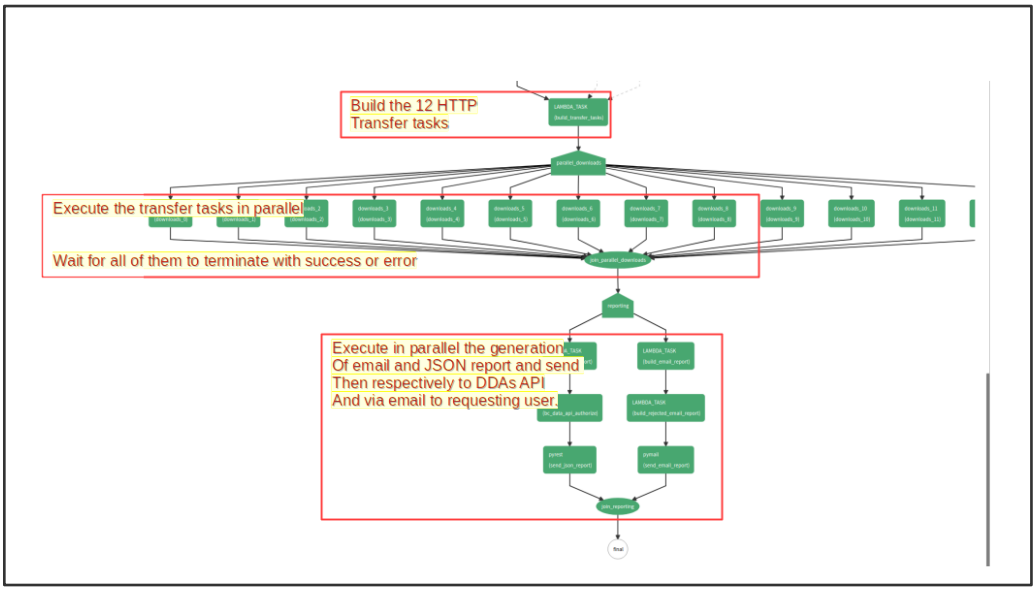

Figure 28.3: da_cache_to_shub workflow - Parallel transfer and reporting.

7.2 WEkEO - Harmonised Data Access API (HDA)

The WEkEO Data and Information Access Services (DIAS) service provides users with a single distributed tool for accessing, visualising and analysing all Copernicus[15] satellite data, model products, and data products. It offers a comprehensive graphical user interface for end-users that helps in selecting large datasets and then sub-setting them to the portion of the dataset required to satisfy a specific need. The WEkEO Harmonized Data Access (HDA) is a unique Application Programming Interface (API) that implements a REST-based single protocol enabling users to issue requests for the data needed.

To simplify the exploitation of the HDA API, a tailored notebook has been implemented and delivered in Blue-Cloud to bridge the Blue-Cloud VRE and the WEkEO DIAS service. This notebook exploits the HDA API and allows users to search for datasets, to select the one that is needed, to see the properties of it, and to issue a subsetting request that is required to access the data the user

[15] https://www.copernicus.eu/

needs to work on. Subsetting is the phase where the user can express the temporal and spatial coverage, the product type, the resolution and all other parameters that are supported by the specific dataset.

The HDA API allows to perform six steps from the authentication to the download as shown in Figure 29.

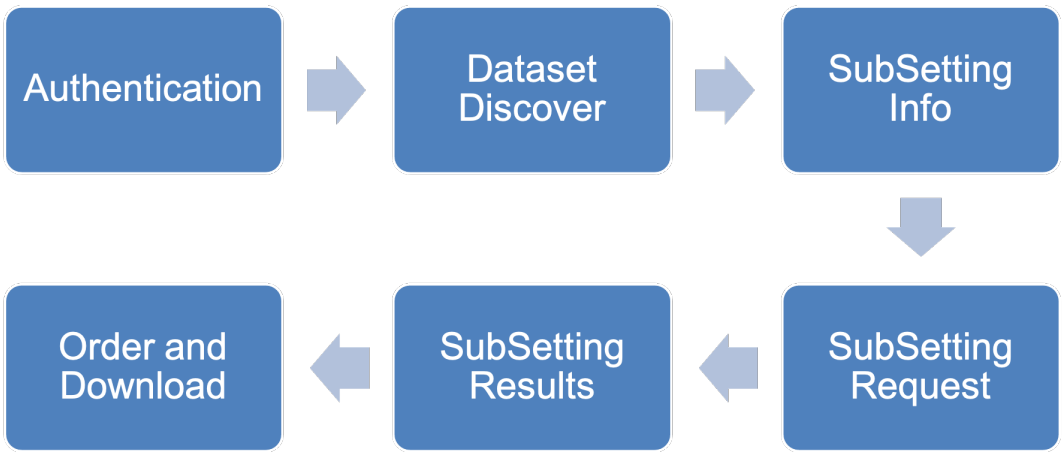

Figure 29: HDA API steps.

For each step, the notebook guides the user through the required actions.

The first step is performed thanks to the single sign-on. It is sufficient to get the token of the session by using the specific button (see Figure 30).

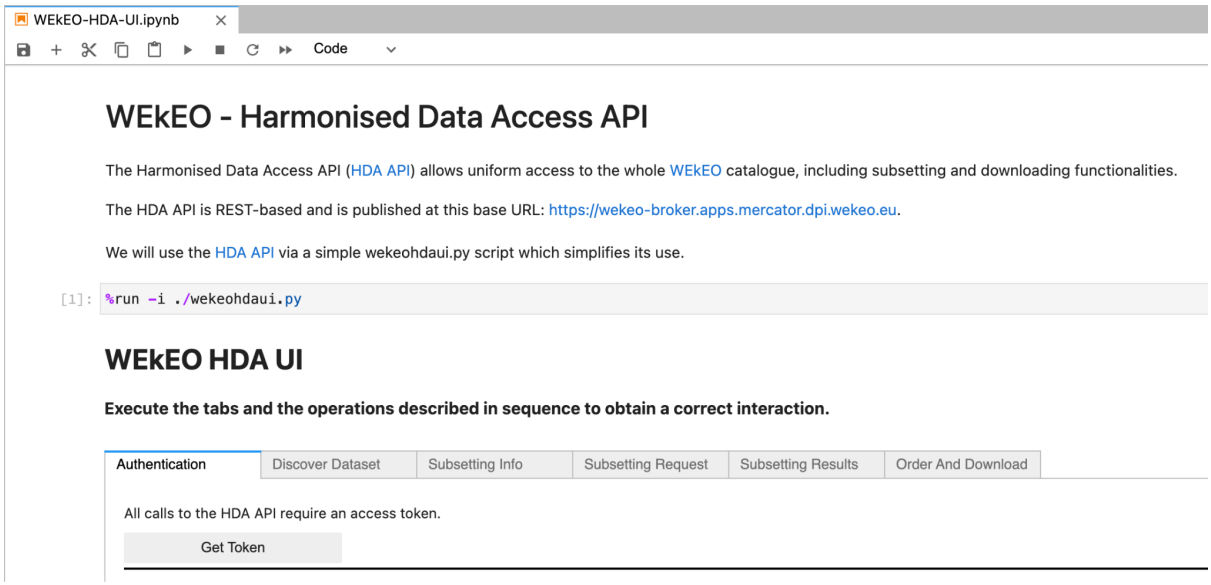

Figure 30: WEkEO authentication performed via the Blue-Cloud VRE.

At the second step, it is possible either to discover specific datasets by typing a query or to access the list of all datasets. For each dataset, the identifier of the dataset, its title, and a description are visualized (See Figure 31).

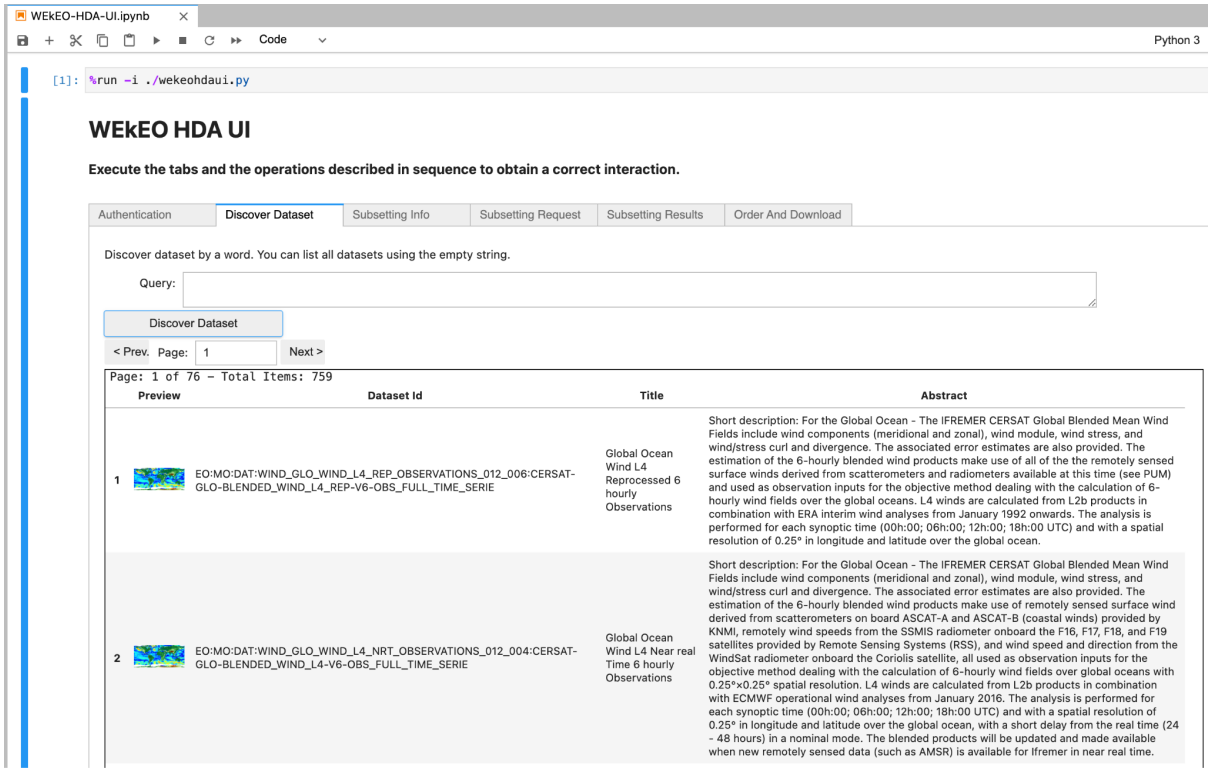

Figure 31: Discover of the WEkEO datasets performed via the Blue-Cloud VRE.

The third step is fundamental to access the subsetting info. Those information are required to perform a valid subsetting request and change according to the selected dataset (See Figure 32).

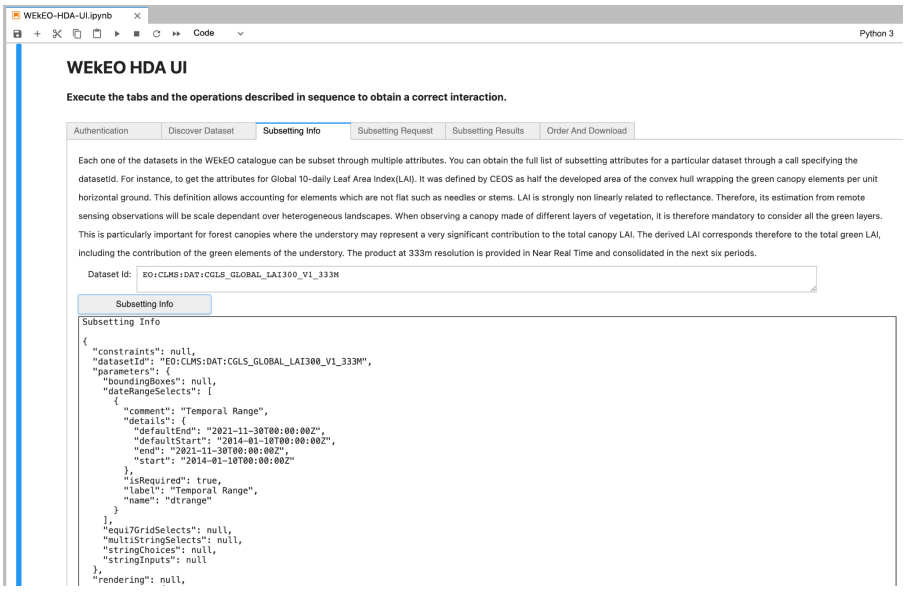

Figure 32: Accessing the subsetting information via the Blue-Cloud VRE.

By exploiting the information collected at the previous step, it becomes possible to issue a proper subsetting request. The request is monitored until its completion (See Figure 33).

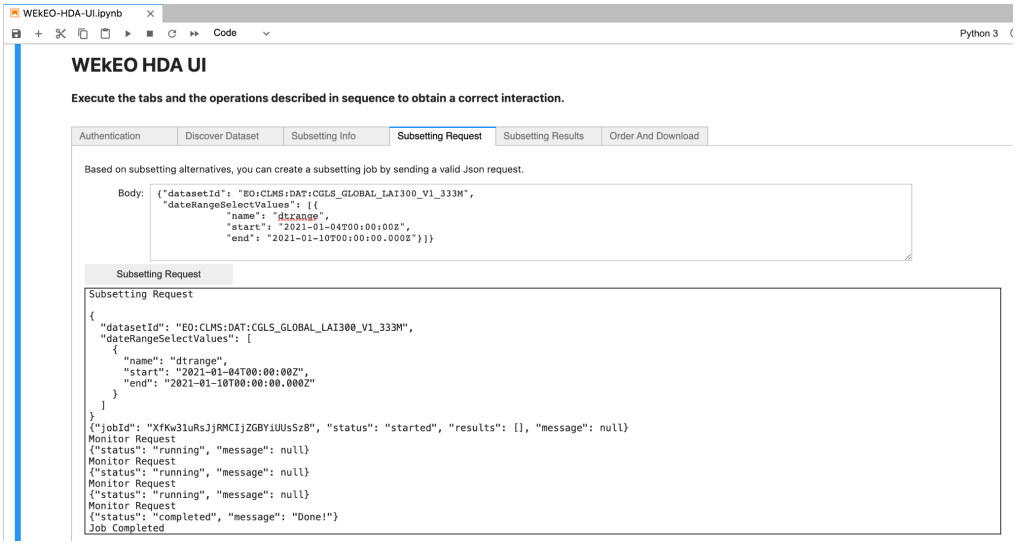

Figure 33: Issue a subsetting request via the Blue-Cloud VRE.

At this point it is possible to retrieve the list of results. Each result contains the key information required to perform the download operation (See Figure 34).

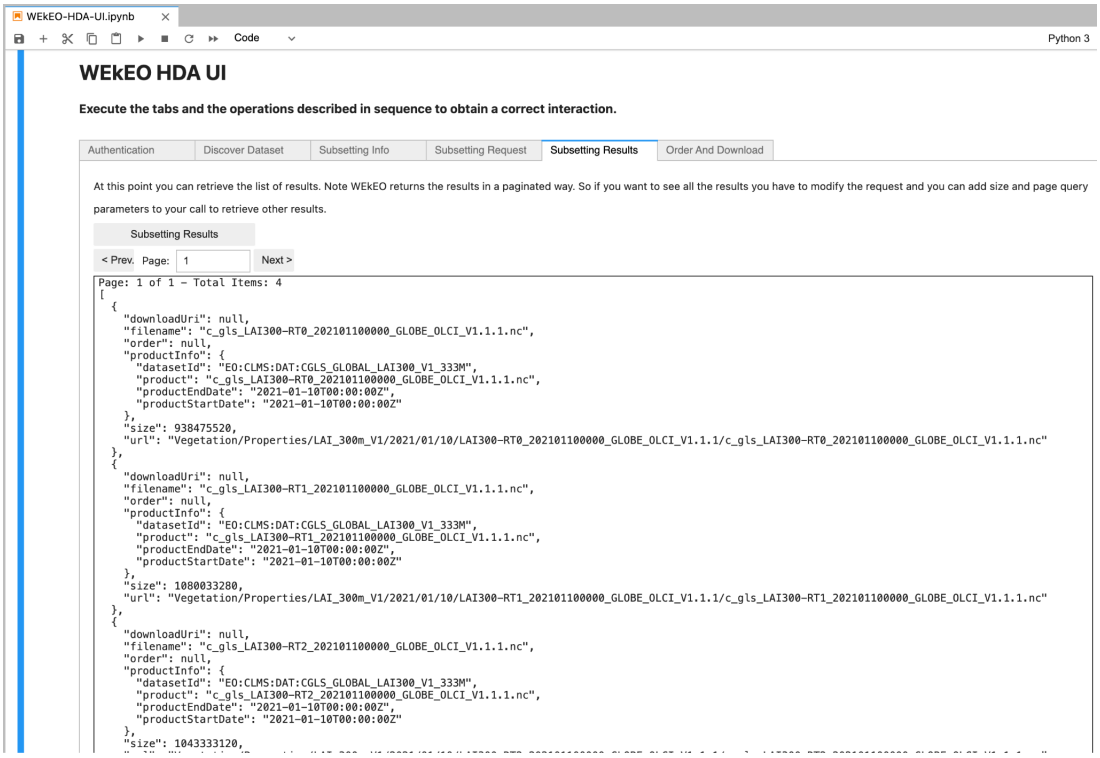

Figure 34: Dataset download instruction via the Blue-Cloud VRE.

Order and Download save the dataset in the folder specified in the request contained into the WEkEO_datasets space of the Dataspace (See Figure 35).

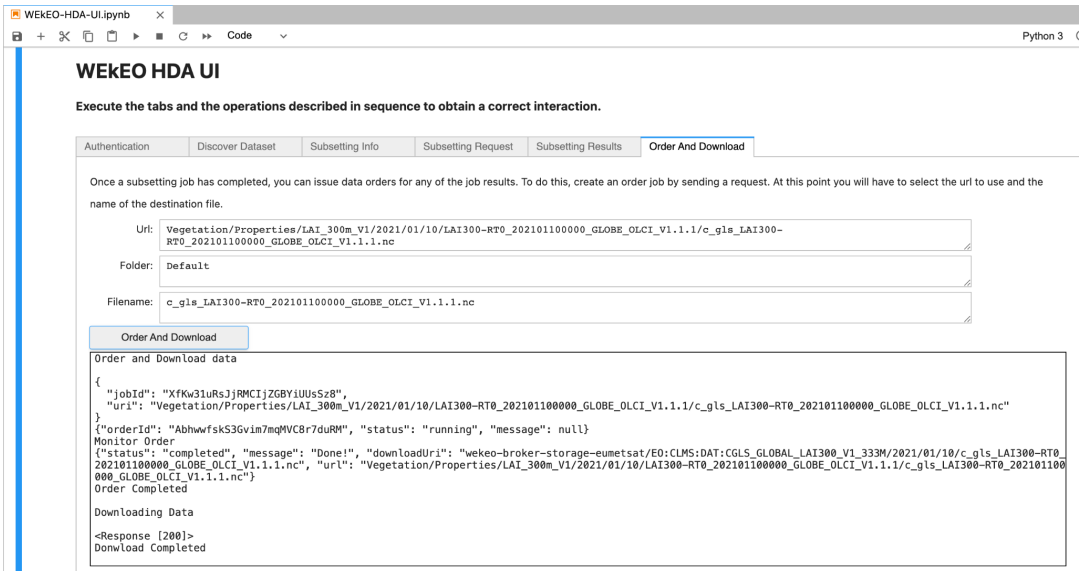

Figure 35: Dataset download via the Blue-Cloud VRE.

For each step, instructions are also provided and in case of error it is sufficient to restart from that step without reinitiating the entire workflow.

8. Conclusion

The deliverable documents the revised and extended Blue Cloud VRE Architecture and its constituents. In particular, it describes the revised version of the overall architecture (cf. Sec 2) and then detailed (a) the services contributing to the Enabling Framework part, namely the Identity and Access Management, the VRE Management, and the Orchestrator; (b) the services contributing to the Collaborative Framework part, i.e. the Workspace and the Social Networking service; (c) the services contributing to the Analytics Framework, namely the Software and Algorithm Importer, the Smart Executor, the RStudio-based solution, the JupyterHub-based solution, the ShinyProxy-based solution for ShinyApps, and the DataMiner-based solution for DockerApps; (d) the service contributing to the Publishing Framework, namely the Data Catalogue and the spatial data catalogue; and (e) the services interfacing with external systems, namely the service interfacing with the Data Discovery and Access Service and the service interfacing with the WEkEO.

This deliverable is implemented as a revised and extended version of *D4.2 ``Blue-Cloud VRE Common Facilities (Release 1)*". Specifically, in order to be self-contained and give an overall description of the VRE Common Facilities it includes a description of all the services previously documented in D4.2 by complementing this with major changes and new services. The major changes and new services this deliverable introduces are: an Orchestrator (cf. Sec. 3.3), i.e. a software that allows for a declarative, technology agnostic definition of workflows to coordinate the execution of tasks across diverse services and systems; the Workspace, i.e. the service providing access to tailored storage persistence (cf. Sec. 4.1); the Smart Executor (cf. Sec. 5.2), namely a component to schedule and execute tasks in batch; enhancements in the Publishing Framework (cf. Sec. 6), namely the catalogue extension to deposit catalogue items to Zenodo and the facility to publish geospatial data from the workspace; the facility to interface with the Data Discovery & Access System (cf. Sec. 7.1) to transfer datasets of interest into the workspace for future uses; the notebook to facilitate the exploitation of the WEkEO Harmonised Data Access (HDA) API (cf. Sec. 7.2).

The components documented by the deliverable have been included and released by the following 15 gCube open source software releases: 4.26 (Nov. 2020), 4.27 (Dec. 2020), 4.28 (Feb. 2021), 5.0 (Feb. 2021), 5.0.1 (Mar. 2021), 5.1 (Mar. 2021), 5.2 (May 2021), 5.3 (Jun. 2021), 5.3.1 (Jun. 2021), 5.4 (Aug. 2021), 5.4.1 (Oct. 2021), 5.4.2 (Oct. 2021), 5.5 (Oct. 2021), 5.6 (Nov. 2021), and 5.6.1 (Dec. 2021). Moreover, they are in the pipelines producing the forthcoming releases. They are exploited to update, develop and operate the Blue Cloud gateway (https://blue-cloud.d4science.org/home) and the underlying infrastructure and services. At the time of this deliverable (December 2021), the gateway hosts a total of 12 VLabs including 6 specifically conceived to support the co-development of some the Blue Cloud demonstrators (namely, the Aquaculture Atlas Generation for Demonstrator #5, the Fisheries Atlas for Demonstrator #4, the GRSF pre for Demonstrator #4, the Marine Environmental Indicators for Demonstrator #3, the Plankton Genomics for Demonstrator #2, and the Zoo-Phytoplankton EOV for Demonstrator #1). This gateway and its services are serving more than 730 users that (since January 2020) performed a total of more than 19000 working sessions, more than 4900 accesses to the workspace, and a series of analytics tasks including 3500+ analytics tasks, 1800+ JupyterLab working sessions, and 800+ RStudio working sessions. These exploitation and uptake indicators are destined to grow in the coming months thanks to data updates and continued use, and the further development of existing VLabs and the creation of new ones.

References

M. Assante, L. Candela, D. Castelli, R. Cirillo, G. Coro, L. Frosini, L. Lelii, F. Mangiacrapa, V. Marioli, P. Pagano, G. Panichi, C. Perciante, F. Sinibaldi (2019a) ***The gCube system: Delivering Virtual Research Environments as-a-Service***. Future Gener. Comput. Syst. 95: 445-453 10.1016/j.future.2018.10.035

M. Assante, L. Candela, D. Castelli, R. Cirillo, G. Coro, L. Frosini, L. Lelii, F. Mangiacrapa, P. Pagano, G. Panichi, F. Sinibaldi (2019b) ***Enacting open science by D4Science***. Future Gener. Comput. Syst. 101: 555-563 10.1016/j.future.2019.05.063

M. Assante, L. Candela, P. Pagano (2020) ***Blue Cloud VRE Common Facilities (Release 1)***. D4.2 Blue-Cloud Deliverable, November 2020

D. M. A. Schaap, P. Thijsse, P. Pagano, M. Assante, E. Boldrini, M. Buurman, M. D'Antonio, C. Ariyo, G. Maudire, C. Nys (2020) ***Blue Cloud Architecture (Release 1)***. D2.6 Blue-Cloud Deliverable, July 2020

D. M. A. Schaap, E. Boldrini, G. Maudire, M. D'Antonio (2021a) ***Blue Cloud Data Discovery and Access service (Release 2)***. D2.4 Blue-Cloud Deliverable, May 2021

D. M. A. Schaap, P. Thijsse, P. Pagano, M. Assante, E. Boldrini, M. Buurman, M. D'Antonio, C. Ariyo, G. Maudire, C. Nys (2021b) ***Blue Cloud Architecture (Release 2)***. D2.7 Blue-Cloud Deliverable, May 2021