JERICO-S3 DELIVERABLE Joint European Research Infrastructure for Coastal Observatories **Science, Services, Sustainability**	
DELIVERABLE #, WP# and full title	JERICO-S3 D.7.6 - WP7 - "Documentation of JERICO-RI e-infrastructure and capabilities"
5 Key words	CORE, e-Infrastructure, VRE, VA, FAIR, Digital ecosystem
Lead beneficiary	SOCIB
Lead Authors	Miguel Charcos Llorens, Miguel Ángel Alcalde, Juan Gabriel Fernandez, Jay Pearlman, Francoise Pearlman, Joaquin Tintore and Juan Miguel Villoria.
Co-authors	Marco Alba, Leo Bruvry-Lagadec, Thierry Carval, Lorenzo Corgnati, Eric Delory, Jerome Detoc, Patrick Gorringe, Keith Jeffery, Simon Keeble, Julien Mader, Carlo Mantovani, Simone Marini, Gilbert Maudire, Antonio Novellino, Antoine Queric, Damia Rita, Pauline Simpson, Peter Thijsse.
Contributors	
Final version date/ Submission date	28/02/2023 / 23/04/2023

Nature: **R**

(R = Report, P = Prototype, D = Demonstrator, O = Other)

Dissemination level : **PU**

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

GRANT N°: 871153
PROJECT ACRONYME : JERICO-S3
PROJECT NAME : Joint European Research Infrastructure for Coastal Observatories - Science, services, sustainability
COORDINATOR : Laurent DELAUNEY - Ifremer, France - jerico@ifremer.fr

DOCUMENT TECHNICAL DESCRIPTION

Document ID	JERICO-S3-WP7-D7.6-28.02.2023-V1.0

REVISION HISTORY

Revision	Date	Modification	Author
V1.0	28-02-2023	First Submission	Miguel Charcos Llorens

APPROVALS

	Name	Organisation	Date	Visa
Coordinator	Delauney Laurent	Ifremer	13/03/2023	LD
WP Leaders	Eric Delory	PLOCAN	28/02/2023	ED

Diffusion list

Consortium beneficiaries	Third parties	Associated Partners	other
X	X	X	

TABLE OF CONTENT

1. EXECUTIVE SUMMARY

In previous projects, the JERICO network has recognized the need to develop a digital ecosystem supporting the activities of coastal communities. JERICO-S3 aims to explore the options and challenges of creating an e-infrastructure. Task 7.5 designed and developed a digital ecosystem prototype in close collaboration with other activities of the project. This e-infrastructure was named in the proposal e-JERICO and re-branded along the process as the JERICO Coastal Ocean Resource Environment (JERICO-CORE).

This deliverable (D7.6) explains the actions undertaken to implement the JERICO-CORE. The process of this task included the collection of the initial requirements of this infrastructure as well as an assessment of the state of the art technologies and infrastructures. We report in this document the methodology used to set the ground of the design of JERICO-CORE. Furthermore, we describe the design resulting from this initial assessment as well as the final implementation. From a planning perspective, we highlight the strategic aspects and decisions of the process. The description of the system is also provided along the document. This includes the details of the system and the usage procedures as well as the procedures for deployment and development. Finally, we explain the outcomes of the demonstration that was planned as milestone MS38.

2. INTRODUCTION

Our planet is experiencing an ongoing change that impacts every aspect of our society [10]. Understanding ocean and coastal areas is critical to evaluate these changes and keep coastal communities, economies and ecosystems healthy. In fact, half of the world population lives within 200 kilometres of the coastal areas because they are the most biologically diverse regions. Coastal oceans connect physical, economic and biological systems of the continents to oceans and seas [11] . Therefore, they are the link for transporting organic and inorganic, natural and anthropogenic material between land, sea and atmosphere and thus, they are subject to be strongly affected by human activities.

The Joint European Research Infrastructure of Coastal Observatories (JERICO) aims to provide an integrated solution in Europe to face the challenges of the coastal marine systems due to natural and anthropogenic stressors. The complexity of physical and biological mechanisms within the coastal areas and their interaction with ocean and land requires an integrated multidisciplinary and multi-observation approach that provides a holistic appraisal of the coastal marine system. In this context, JERICO-RI (https://www.jerico-ri.eu/) is working toward contributing to fill a key gap in the landscape of the European STRategy Forum on Research Infrastructures (ESFRI).

The process of collecting multiplatform observations and integrating data of different nature is complex and requires a high level of harmonisation and integration of technologies and data. JERICO started this process in 2011 in order to increase cooperation and improve synergies between European partners of coastal marine Research Infrastructures (RI). After JERICO (2011-2015, FP7) and JERICO-NEXT (2015-2019, H2020) projects, the JERICO network reached a significant level of maturity of infrastructures, technologies and platforms such as moorings, drifters, ferrybox and gliders as well as the interconnection between physics, biochemistry and biology. The strengthening of the JERICO network channelled the creation of a high number of assets, products and services that are used for the observation and analysis of coastal areas. The explosion of resources in the JERICO community requires the creation of a digital ecosystem that supports their discoverability and usability. This e-infrastructure must include virtual access services facilitating open-access of FAIR and reliable information to industry and society and catalysing international science collaborations.

For this purpose, JERICO-S3 and JERICO-DS are building the **JERICO Coastal Ocean Resource Environment** (**JERICO-CORE** or **J-CORE**), formerly known as **e-JERICO**, as the unified central hub of JERICO to discover, access, manage and interact with JERICO resources including services, datasets, software, tools, best practices, manuals, publications, organisations, projects, observatories, equipment, data servers, e-libraries, training, and similar assets. Increasing discoverability of assets is also critical to create services that respond to specific needs of users. JERICO-CORE also provides a framework to facilitate collaboration and creation of services. While JERICO-DS will perform a study of the available technologies for a e-infrastructure that supports the work of JERICO-RI in the long term for the ESFRI roadmap, JERICO-S3 created a prototype that will help the JERICO community to mould, carve and shape their ideas into a final e-infrastructure that responds to the needs of coastal oceanography. This process is critical to scope the needs of the community and to identify the main challenges in the creation of an integrated solution. The JERICO-CORE e-infrastructure prototype was developed under task 7.5 as a scalable framework that allows the visibility and access of JERICO-S3 resources and a connected Virtual Research Environment where services can be developed to increase the scientific and societal impact in a long-term sustained RI.

In this document, we explain the methodology used to design and implement the JERICO-CORE prototype as well as the results of the work. This work was framed under task 7.5 and it was focused on co-design and co-development with various internal and external stakeholders. Task 7.5 aims to *"define and develop a Virtual Access (VA) scalable framework that allows the visibility and access of the JERICO-S3 resources with the aim of increasing the scientific and societal impact in a long-term sustained RI."* In particular, the development of the VA portal under task 7.5.1 is described as follows: *"Operational requirements will be derived with JERICO-RI partners, modellers, product developers and other experts in collaboration with WP11. Requirements will be used for detailed design of the VA portal. This development will include a User Interface (UI), an IT infrastructure, connectivity to the JERICO data and services catalogues, access to the best practices systems and an e-library for tools and similar resources. In addition, the VA may provide access to aggregators like Regional Operational Oceanographic Systems (ROOS), Copernicus Marine Service (NRT), SeaDataNet (validated archives), EMODNet Physics and Biology portals. Access to priority/mature tools from partners will be incorporated into the VA and will help to test the e-infrastructure performance. This activity will set up the first elements of the JERICO e-infrastructure, e-JERICO, that will be operated in WP11 VA to support users."*

Firstly, the methodology section explains the interfaces when co-designing with stakeholders and the results of the conversations. The interaction with key partners and the commitments of the project resulted in a list of requirements and the conceptual design of the prototype responding to them. Section 3.2 (Requirements) explains the discussed and agreed requirements established with stakeholders during the process. The section 3.3 (Concept) provides an overview of the conceptual framework and how each component of the concept responds to the needs of the stakeholders. Among the interaction with stakeholders we highlight the discussions with experts from various e-infrastructures that we used as reference and potential baseline for JERICO-CORE developments. This approach was necessary to respond to the challenge of creating a complex infrastructure in the short deadline that the project committed to. The study of these infrastructures is presented in section 3.4 (Study of Technological Solutions). In this analysis, we explore various e-infrastructures and we examine how they respond to the concept that was agreed with stakeholders. The study of existing solutions includes an analysis of available technologies that we examined to integrate and fill the gaps of the studied e-infrastructures. Technological solutions were analysed with small proof of concepts before deciding the architecture for the JERICO-CORE prototype. The architecture design and the implementation of the solution is presented in section 3.5 (Architecture and Design). The design and implementation plans were adapted along the advancement of the project in relation to the decisions with stakeholders along the process and the availability of the selected solutions. Section 3.6 (Planning) explains the timelines of the different phases of the project and the response to the various circumstances along the way. Then, section 3.7 (Implementation) explains the technical details of the final implementation. A description of the resulting solutions is provided for each component of the architecture and compared with the initial planning. This section is followed with a description of the procedures that should be followed to develop and maintain expanded features of JERICO-CORE. Section 3.8 (Development and deployment procedures) also includes the description of the methods followed to deploy JERICO-CORE system. Finally, section 3.9 (Demonstration and future plans) explains the results of the demonstration of JERICO-CORE to the JERICO partners. We included in this last section the list of lessons learned from the demonstration and the experience developing and testing JERICO-CORE.

3. MAIN REPORT

3.1. Methodology

The work to reach operation of JERICO-CORE under WP11 was performed in five phases (see figure 1). The initial phase (study of the current situation) required understanding the current status of JERICO and the needs among partners and the international community. This phase was critical in order to make sure that the design responds to the requirements of JERICO-RI and takes advantage of all available resources. It required a large amount of interaction and co-designing with stakeholders. The resulting requirements and concept from this first phase (study current situation) are shown in sections 3.2 (Requirements) and 3.3 (Concept).

The study of the current situation considered four aspects:

- **Understanding the current status** of JERICO-RI resulting from previous projects and the aspects that JERICO-CORE should address to support JERICO-RI.

- **Collecting needs from partners** and co-designing the concept and architecture.
- **Understanding the role** of JERICO-CORE in the international context and co-designing the infrastructure to guarantee the integration of the e-infrastructure in the European and broader landscapes.
- **Studying the available solutions** (technological and other e-infrastructures)

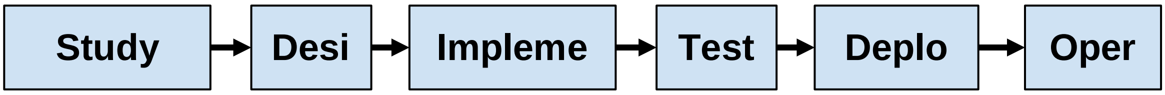

Figure 1: Process to implement and test JERICO-CORE in the context of T7.5 of JERICO-S3 toward its operation under WP11

These four aspects of the methodology used to address the current situation are explained below.

3.1.1. Understanding Current Status

The need of a virtual infrastructure to support JERICO activities was identified during the JERICO-NEXT project. In particular, JERICO-CORE should support the discoverability of all resources created during the JERICO and JERICO-NEXT projects. We started the process by collecting the information from documents and deliverables from JERICO and JERICO-NEXT projects. The list of materials is shown in table 1.

We related each of these documents to the potential different capabilities for JERICO-CORE (see appendix A):

- **Core and interfaces** (see table A-1): main components of JERICO-CORE including the resource catalogue design and the interfaces with the JERICO infrastructures.
- **Integration of data** (see table A-2): components related to the operability of distributed data systems.
- **Integration of services** (see table A-3): components related to the operability of distributed service systems.

The information collected from these reports were the baseline of the conversations with partners explained in the following section.

Material	Reference	Project
Integrated pan European atlas	deliverable 2.3 , deliverable 2.5	FP7
Demonstration of the feasibility of joint trans-regional product production	deliverable 2.4	FP7

Data management reports	deliverable 5.3, deliverable 5.7	FP7
Delayed mode handbook	deliverable 5.6	FP7
RT mode handbook	deliverable 5.8	FP7
Service Access Provision	deliverable 7.1	FP7
Catalog of Virtual Access services	deliverables 5.2, 8.2 & 8.3	NEXT
JERICO-NEXT website	deliverable 8.7, deliverable 8.8	NEXT
Requirements, high level design and software architecture of an observatory operator console	deliverable 5.6	NEXT
Table summarizing the JERICO-NEXT infrastructures in WP6, Virtual access and used by Task 5.8 D5.16	JERICO-NEXT_Deliverable_5.1_v1.3.pdf (page 8)	NEXT
Linking JERICO-NEXT activities to a Virtual Access infrastructure	deliverable 5.16	NEXT
Virtual Access posters		NEXT

Table 1: Existing material from JERICO and JERICO-NEXT that provide outcomes related to the JERICO-S3 e-Infrastructure.

3.1.2. Collecting needs from JERICO-S3 partners

The aim of WP7 of JERICO-S3 is to develop and integrate innovative technologies and methodologies, in tight collaboration with other work packages of JERICO-S3. Thus, involving all partners of JERICO-S3 in the process of designing and implementing JERICO-CORE was a natural path and critical to collect the appropriate requirements and solutions. Those were assembled, completed and consolidated with a series of meetings and interactions (see appendix B). JERICO-CORE infrastructure integrates various technological components including a User Interface (UI), an IT infrastructure, connectivity to the JERICO data and services catalogues, access to the best practices systems and an e-library for tools and a variety of resources. Meetings with partners were critical to define the interfaces and the needs from other tasks that affect the design of these components. Therefore, the design and development of JERICO-CORE required a high level of collaboration, co-design and co-development. In general, the co-design process had a positive effect in the JERICO partners relation since it helped the project building capacities in the elaboration of the JERICO virtual framework.

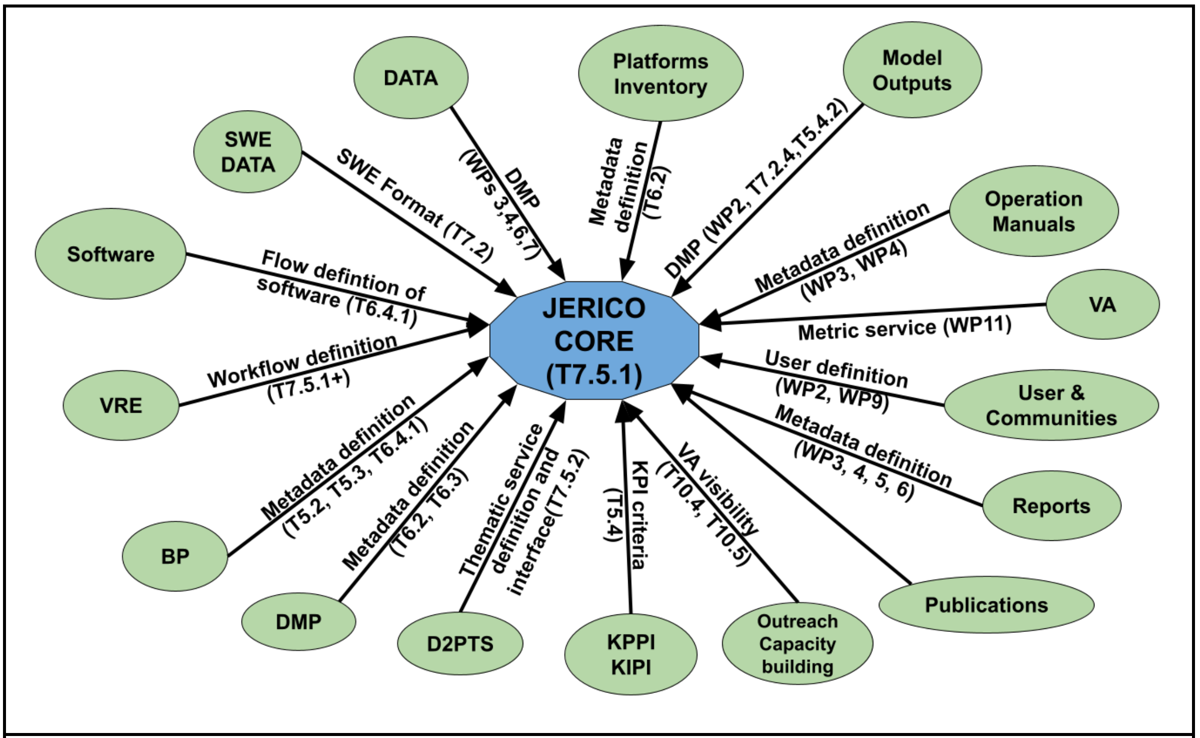

Figure 2: Roles and interfaces of work packages, tasks and teams in the design of the JERICO-CORE prototype (T7.5).

Components of JERICO-CORE were defined during the co-design process and are described in more detail in the conceptual design section of this document. Defining, designing and implementing these components required interfacing with various JERICO-RI teams. We defined these interfaces in the JERICO-S3 Grant Agreement and during the initial discussions, especially during the JERICO-S3 kick-off meeting. Figure 2 shows the activities that were carried out to design JERICO-CORE inside WP7 (in particular under task 7.5.1) and also all the links with other WP's. Bubbles represent the different aspects of JERICO-CORE that were addressed. Arrows represent the interaction with the groups that were required to address these aspects. We explain below in detail these aspects and interactions:

- **Data Management Plan (DMP) - Metadata definition (T6.2 and T6.3)**: Access to DMP requires a clear definition of the collection methods of these documents.
- **Best Practice (BP) - Metadata definition (T5.2, T5.3 and T6.4.1)**: Access to best practices requires a clear definition on ways to identify the ones that belong to the JERICO community and the flow defining the methods to collect them through the OBPS. It is important to define the tools and agree on ways to link the BPs to other resources. The definition of the tools to support the interactive implementation will impact the integration of these tools.
- **Virtual Research Environment (VRE) - Workflow definition (T7.5.1+)**: The European Open Science Cloud (EOSC) is an initiative launched by the European Commission in 2016, as part of the European Cloud Initiative. EOSC aims to provide a virtual environment with open and seamless services for storage, management, analysis and re-use of research data, across borders and scientific disciplines, leveraging and federating the existing e-infrastructures and thematic infrastructures.

The JERICO-CORE VRE should align with EOSC plans in the context of a marine thematic service such as Blue-Cloud.

- **Software - Flow definition of software (T6.4.1)**: The integration of software requires knowing the status of the current tools among the community and the methods to share them.

- **Sensor Web Enablement (SWE) Data - SWE format (T7.2)**: The sensor web server is a critical piece to integrate SWE sensors in the data flow

- **Data - DMP (WPs 3, 4, 6 and 7)**: The definition of the data management plan determines how the data is collected in the JERICO community and who are the key players in this flow. The interface with aggregators (e.g. EMODnet, SeaDataNet and Copernicus Marine Service) defines the possible systems that can be in place to collect data as well as the type of metadata that can be gathered. It was decided that citizen science is very diverse to be included in the data flow. SWE sensors are an important piece and including them to the data flow is an important added value to JERICO and the ocean community in general.

- **Platform inventory - Metadata definition (T6.2)**: The diversity of platforms used in the JERICO community is an important strength of the RI. Integration of platforms required understanding the status of platform inventories such as Sextant (https://sextant.ifremer.fr/).

- **Model Outputs - DMP (WP2, T7.2.4, T5.4.2)**: Models are an important element of coastal oceanography. Although modelling was not included in the JERICO-S3 proposal, we interacted with partners that required interaction with the modelling community.

- **Operation manuals - Metadata definition (WP3 and WP4)**: The support of operation requires discovering and accessing manuals, calibration sheets and other important operational documents. We interacted with the regional partners to learn from their expertise in the field of operation.

- **Virtual Access - Metric service (WP11)**: The discovery of the VA requires the definition of the information that needs to be collected to represent these infrastructures and the ways the metrics are collected and presented to various types of users.

- **User and communities - User definition (WP2 and WP9)**: Creating a useful tool for the community requires understanding the users and stakeholders of the e-infrastructure.

- **Reports - Metadata definition (WPs 3, 4, 5 and 6)** and **Publications:** Reports and publications are products resulting from the assets of JERICO. They can have an important impact on society and decision making.

- **Outreach capacity building - VA visibility**: Community outreach programs are avenues to bring growth to society and the people around us, raise social awareness on sensitive matters. Supporting these programs require understanding the link with the JERICO activities and resources.

- **KPPI and KIPI - KPI criteria (T5.4)**: Defining the ways of how the parameters defining these Key Performance Indicators (KPI) will be collected and presented to the users are important to integrate methods for inputting indicators and defining the ways to view dashboards in the JERICO-CORE portal. The e-infrastructure will provide support to the visualisation of Key Platform Performance Indicators (KPPI) and Key Integration Performance Integration (KIPI) and ultimately to the JERICO label during the ESFRI roadmap.
- **D2PTS - Thematic service definition and interface (T7.5.2)**: The integration of data to product services required a close definition of the way they can be accessed from outside the data centre providing these services.

Additionally, we involved stakeholders outside the JERICO projects that could share their experiences working with similar infrastructures and could provide feedback on how JERICO-CORE fits in the international landscape. In the following section, we explain the details of the co-design and co-development process with these external stakeholders.

3.1.3. Understanding our role in the international context

It is envisioned that JERICO-CORE will provide access to and integrate with European and international data aggregators and information systems. The European landscape for marine data sets is a complex environment involving various "blue data infrastructures" (BDI) [1]. In particular, three European marine data infrastructures play a major role in managing and collecting marine in-situ and remote sensing data: the Copernicus Marine Service, SeaDataNet and the European Marine Observation and Data network (EMODnet). Figure 3 summarises the relation between these three European aggregators and national actors.

Real time or near real time data from organisations are collected by the Copernicus Marine Insitu TAC service with their EuroGOOS/ROOS partners. SeaDataNet (https://www.seadatanet.org) ingest, validates and distributes delayed mode data sets. EMODnet (https://www.emodnet.eu) collects data from Copernicus In Situ TAC and SeaDataNet as well as organisations around Europe and offers ocean data in the form of thematic products. JERICO-S3 data management plan defines the role of these three pan-European marine infrastructures in sharing and distributing data. In this complex framework of pan-European infrastructures we do not have to forget the role of the various nodes and National Oceanographic Data Centers (NODC) that play a specific role in the flow of the data. Some examples of these nodes are Coriolis for glider data and the EU HFR node for HF radar data. The Global Telecommunication System (GTS) also plays a key role in the data cycle by collecting ocean and meteorological data on a global scale. There are other important data infrastructures that are not shown in figure 3 but that should be included in the horizon of JERICO-CORE such as the Integrated Carbon Observation System Research Infrastructure (ICOS-RI), Euro-Argo (ocean physics and marine biogeochemistry), EuroBIS (marine biodiversity), ELIXIR-ENA (biogenomics). We met with key representatives of some of these infrastructures. The results of these conversations are explained in appendix B which also includes discussions with experts of Sensor Web Enablement (SWE) technologies such as 52ºNorth (https://52north.org). In this context the Sensor Observations Service (SOS) defines a set of interfaces for managing data (metadata requests, filters, retrieve, ...) servicing real-time sensors via SWE servers. SWE standards are gaining ground in the JERICO plans mainly due to the increasing use of smart sensors and platforms in the ocean community. The developments under WP7 of new sensors and the expansion of parameters and platforms is in line with the ongoing SWE development.

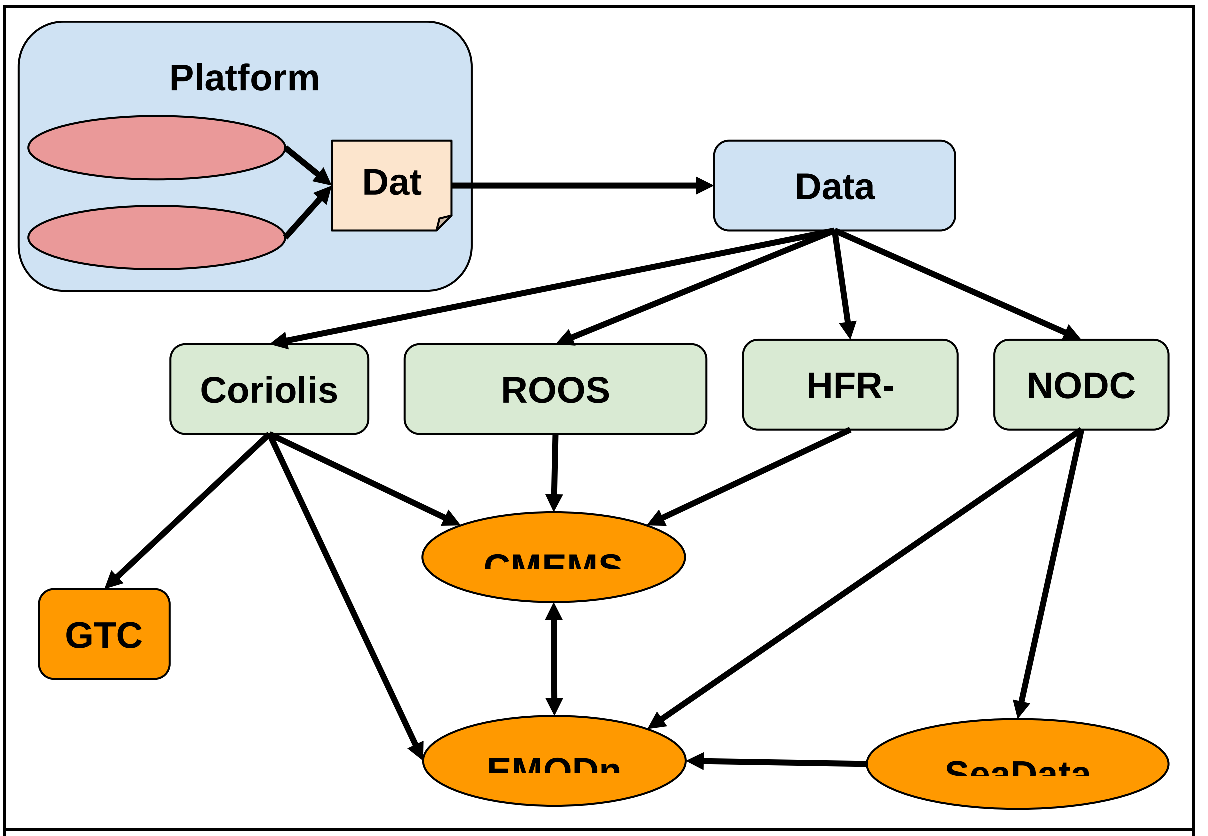

Figure 3: JERICO-RI data flow diagram in the European landscape

Federated resources related to ocean observations should be made available not only concerning datasets but also related to best practices, manuals, processing software, virtual access and operational equipment and documents.

The long-term plans for the JERICO community should consider the existing European and international initiatives such as the European Open Science Cloud (EOSC) and the IOC Ocean Data and Information System (ODIS). In the context of JERICO-S3, it was decided that the representation of resources should adhere when possible to the approach of these catalogues and be represented as a knowledge graph. These knowledge graphs should not only represent available services of the JERICO-RI community but also describe a complete picture of the assets defined within the conversations with JERICO partners (see section Collecting needs from JERICO-S3 partners). It was necessary to consider providers of other types of resources as follow:

- **Virtual Research Environment**: Blue-Cloud will encompass the leading European blue data infrastructures and integrate with the existing Common Data Index (CDI) based services of SeaDataNet. It was expected that JERICO-CORE would build synergies with the Blue-Cloud project.
- **Documents**: The Ocean Best Practice System (OBPS) is a recognized technological solution and community to manage methods and best practices. AquaDocs

(https://aquadocs.org/) is a hub for collecting, preserving and discovering research and technical documents from the marine and aquatic community. This repository replaced OceanDocs (https://oceandocs.org) which is now discontinued.

- **Software**: There are a number of infrastructures that serve as a repository for software in a broad view outside the ocean research. Github and Docker Hub are the largest communities and repositories for various types of software tools and images.

JERICO-S3 made a special effort to define data management from acquisition to processing and dissemination. FAIRness is extensively described in deliverable 6.1 [9] together with interoperability and the re-use of the collected data. The access to information of JERICO resources and their link with data makes JERICO-CORE an ideal candidate for increasing FAIRness of the JERICO data and other resources. We strived to implement the whole concept of FAIRness within the infrastructure including making all data and services Findable, Accessible, Interoperable, and Re-usable, both for machines and for people. Once again, we see the importance of adapting the JERICO-CORE approach to cloud initiatives and therefore to the principles defined by the ENVRI-FAIR project aiming to connect the ENVRI cluster to EOSC.

The interaction with stakeholders resulted in a clear definition of the flow of information of data and other digital assets (see figure 4). Understanding the role of JERICO-CORE as an e-infrastructure and the flow of information of digital resources was necessary before investigating the technological solutions that are available as explained in the next section.

3.1.4. Studying available solutions

The design and development of such a complex e-infrastructure in just twenty-four months required strong synergies with experts of other digital ecosystems. Not only at the initial phases of the project but we reviewed the design and development regularly with the teams that were most impacted by technical or operational decisions. We performed the initial study of existing similar e-infrastructures in three phases. In the first phase, we studied the existing documentation about these infrastructures. This study was done mostly from public documents. These documents provided details of these infrastructures that were necessary to compile a list of important criteria for JERICO. These criteria are the base for deciding which infrastructure fits best the requirements of the JERICO community and the technical requirements for the development of the e-Infrastructure. During the second phase we contacted these infrastructure partners via email followed by remote calls to discuss the important technical details of their infrastructures. In the third phase, these documents (see table 4) were shared among the partners of these infrastructures for the purpose of completing the details and guaranteeing the accuracy of the information.

The details and conclusions of this analysis are explained in section Analysis of existing e-infrastrucrures. After the discussions and analysis of these infrastructures, we studied the components that complement the gaps of these solutions. These required studying the existing technological solutions mostly in the area of knowledge representation and storage (see section Analysis of existing technologies). The study of these solutions was necessary for the elaboration of the architecture design that responds to the requirements that were collected among the internal and external stakeholders. We present these requirements in the following section.

3.2.Requirements

In the context of JERICO-S3, requirements were collected among stakeholders for the purpose of designing and implementing the JERICO-CORE prototype that is planned under task 7.5.1. Further evolution of the JERICO-CORE design are carried out in the JERICO-DS project under WP3. Task 3.1 of JERICO-DS is dedicated to a more thoughtful collection of requirements for the long term operation of JERICO-CORE for JERICO-RI. In this section we explain the requirements collected inside task 7.5.1.

There was a general consensus among the stakeholders that JERICO-CORE must use the existing infrastructures and fit in the context of the coastal ocean European and international landscape. Moreover, not only should it not have to replicate any of the work of these infrastructures but it should highlight their role. The JERICO-CORE infrastructure niche lives in cataloguing and making available the large diversity and heterogeneity of the distributed resources in the JERICO community, basically a single access to a very broad range of resources and tools. The main spotlights of these conversations about JERICO-CORE are:

- It should not be a data repository, but integrate with the existing European aggregators as data references.

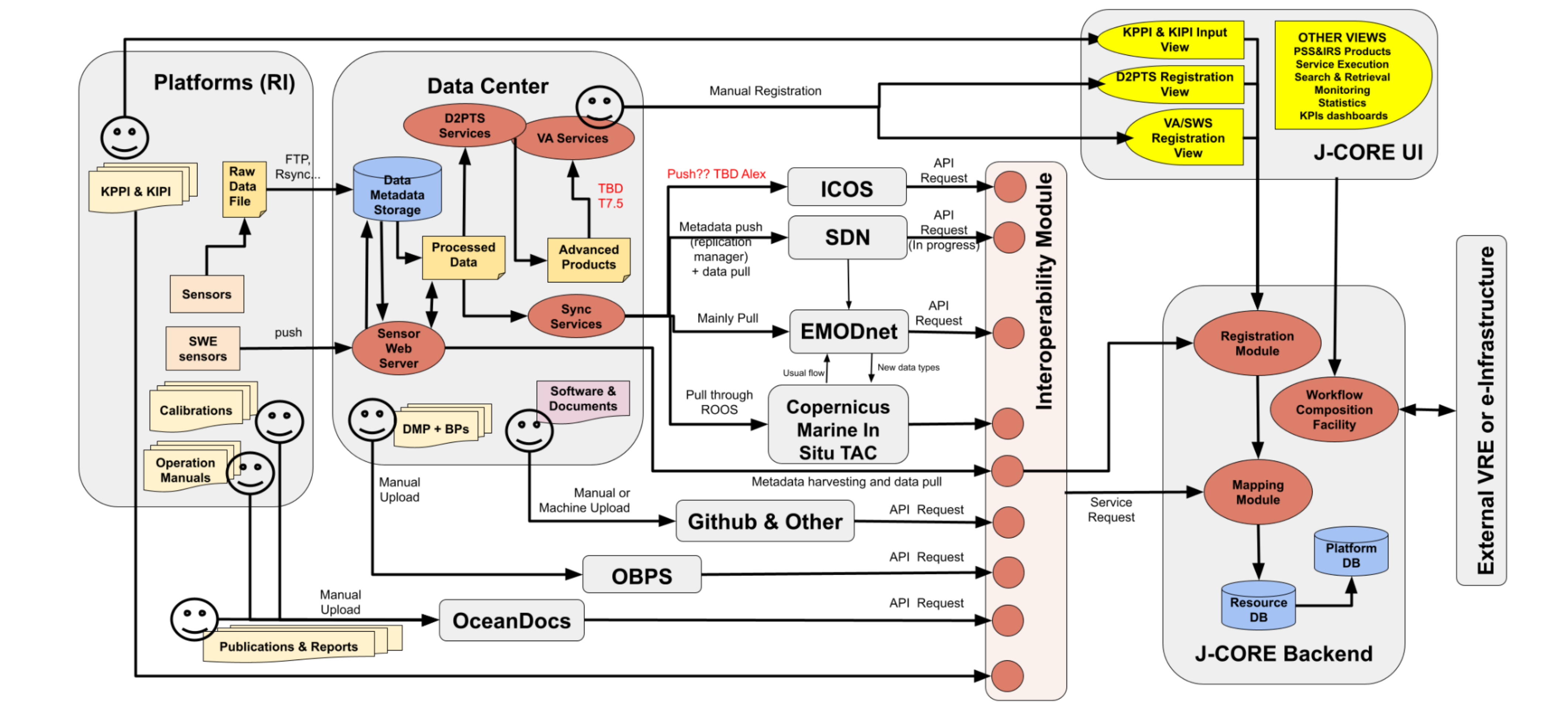

Figure 4: Detailed view of the flow of information in the process of collecting the data and knowledge from resource providers. These flows are the result of the interaction with WP6, WP5, 52N, OBPS, EMODnet, SeaDatanet and Copernicus Marine Service Insitu TAC. Other partners were also contacted for clarification of some aspects.

- It should emphasise the distributed services that exist in the JERICO organisations. Exposing Virtual Access should be one of the functionalities but there should be integrated services that run in the distributed infrastructures but are exposed seamlessly. The D2PTS pilots should be used as pilots for that purpose.
- There is a need to integrate in the resource flow the necessary information to stand out the Key Platform Performance Integrator and the Key Integration Performance Indicators (KIPI) that will be defined in the context of T5.4.
- It will add an important value to the JERICO community and in general the ocean community to integrate sensors with SWE capabilities. These sensors are also developed in the context of JERICO-S3 by T7.2.
- It should host a catalogue of platforms to support the use of SWE technologies.

There were suggestions about possible technologies that we will consider for the design of the architecture in subsequent phases:

- ERDDAP: data server that gives a simple, consistent way to download subsets of gridded and tabular scientific datasets in common file formats and make graphs and maps.
- Helgoland viewer: open-code software client for visual exploration and analysis of sensor web data developed by 52°North.
- Data proxies and data caches to avoid duplication of data and avoid problems of synchronisation of the data with aggregators.

The complete list of requirements and needs from stakeholders was broad. In order to keep the goal realistic, we limited the scope of the prototype to the main requirements and the commitments of the JERICO-S3 project. The other requirements will be considered in the context of JERICO-DS for a long-term design of JERICO-CORE. Table 2 lists the requirements as described in the Description of Work (DoW) of JERICO-S3 and the relation to the discussions with the JERICO partners.

These requirements were the foundation for the work done under task 7.5. In particular, the conceptual design described in the next section was co-designed in order to respond to the needs identified during the interaction with other JERICO partners.

Reference	P.	Description
Summary	3	JERICO-S3 is mainly targeting a more science integrative approach to better observe the coastal ecosystem, raising up the scientific excellence, with consideration of the regional and loca **preliminary development of an e-infrastructure in support to scientists and users by offering access to dedicated services**; progress on the design of the RI and its strategy for su
Deliverables	10	**Documentation of JERICO-RI e-infrastructure and capabilities**
Deliverables	10	Intelligent services and data science methodologies for the JIIM and the **VA einfrastructure**
WP5 objectives	36	A JERICO-RI dashboard will be designed for **its implementation in the JERICO e-infrastructure** (WP8) as a hub for a coordinated management of the harmonized observing platforms a Key Performance Indicators.
T5.2.2	37	Tools will be **made available through JERICO e-infrastructure** (WP7) with direct impact for the harmonization in IRS (WP3) and PSS (WP4).
T5.4.3	37	A dashboard will be designed for an integrated management of the harmonisation in each PSS/IRS as a tool for the JERICO Label Committee (WP13). It will allow a continuous monitorin Level of each technology in each PSS/IRS. A first version will be launched M12 and a completed version will include the developed Key Performance Indicators from the previous subtask. **implementation will be delivered to JERICO e-infrastructure (WP11).**
T6.2.2	42	For the following platform types there will be activities supporting data management as defined under T6.4 in close relation to the **Virtual Research Environment in WP7**:
T6.2.2	42	use of the **VRE** by the partners engaged in WP3 and WP4 (D6.10). This will be done in cooperation with WP3 and WP4, and is an **important condition for data to be accessible for use in the Virtual Environment in T7.4.**
WP7 objectives	49	**The JERICO research e-infrastructure will be developed (task 7.5), where a portal geared with a metadata-based registry system and search engine will ease discoverability a resources, including observation data and data product services, open software and best practices.**
T7.4	50	The proposed methodologies will be designed **to be included in the JERICO e-infrastructure (Task 7.5)**.
T7.4.4	50	Data science methodologies and Intelligent services **integrated into the e-infrastructure** (M21 - M33)
T7.5	50	**define and develop a Virtual Access (VA) scalable framework that allows the visibility and access of the JERICO-S3 resources with the aim of increasing the scientific and so long- term sustained RI.**
T7.5.1	50	**VA Portal development (M1-M30): Operational requirements will be derived with JERICO-RI partners, modelers, product developers and other experts in collaboration with WP11.**
T7.5.1	50	**Requirements will be used for detailed design of the VA portal. This development will include a User Interface (UI), an IT infrastructure, connectivity to the JERICO data and s access to the best practices systems and an e-library for tools and similar resources.**
T7.5.1	50	**In addition, the VA may provide access to aggregators like ROOSes/Copernicus Marine Service (NRT), SeaDataNet (validated archives), EMODNet Physics and Biology portal priority/mature tools from partners will be incorporated into the VA and will help to test the e-infrastructure performance.**

T7.5.1	50	**This activity will set up the first elements of the JERICO e-infrastructure, e-JERICO, that will be operated in WP11 VA to support users.**
T7.5.2	51	Data-to-Products Thematic Services (D2PTS) (M1-M25) : This subtask will create four pilot-focused regional/thematic services from JERICO-S3 data to demonstrate the benefits of the JERICO RI information life cycle. The work will be done in the areas of physical, biogeochemical and biological oceanography to be exemplars on "how to" for larger scale creation of products and services.
Deliverables	53	**D7.6: Documentation of JERICO RI e-infrastructure and capabilities.**
Milestones	54	**MS38 Demonstration of VA infrastructure**
T9.2.1	66	The JERICO-User Committee will be consulted to give feedback on services delivered by JERICO-RI at the hardware level that will partially be based on JERICO-RI TNA experiences (sin 2011), and at the virtual level **that includes services access in the e-JERICO-RI** initiated in the prototype phase in JERICO-S3
T9.3	67	**WP11 will operate services within the preliminary e-infrastructure of JERICO-RI (including a Virtual Research Environment, VRE).**
WP10	72	to promote the benefits for the scientific community: internal strengths and capabilities of the RI, including development of capacity building, sharing of expertises, **training to access and e-infrastructure**,
T10.4	73	A particular emphasis will be given to promote synergies with the Virtual Research Environment that will be developed in WP7.
T10.5.2	73	In particular, the **enhanced JERICO-RI e-Infrastructure** and its contents, including JERICO-RI resources catalogs (platforms, data, documentation, e-libraries and e-training modules) an Thematic Services.
		General Objective: **Provide smooth and efficient access to the e-JERICO**, the virtual part of the JERICO-RI and Resources for researchers or research teams from academy and indus TA and VA instruments, assuring integration of TA activities in VA framework and enhancing the access to bio-geo-chemical and biological observing systems, thus reinforcing the ocean c access opportunities will build long-term, collaborations between users and JERICO-S3 RIs.
WP11 objectives	77	Provide and facilitate centralized access to JERICO-S3 RI resources (data, products, services, best practices, VRE tools, platforms, e-libraries and e-training modules).
WP11 VA description	77	This VA framework implies the **development of an JERICO RI e-infrastructure (e-JERICO)** supported by two main components in specific tasks that are to be included in WP7 (VA Porta central hub where users can individually discover and access the JERICO-RI Resources and a set of Pilot Data-to-Products Thematic Services for integrating and analysing multidisciplina
WP11 VA obligations	77	update (when required from WP11 leaders) into the **online resources catalog the VA activities** to be publicized and accessed through the e-JERICO,
WP11 support description	78	The offered support will be achieved through the **maintenance and operation of the e-JERICO during the projec**t, including the VA portal interface and backend, resources catalogues, functionalities and both tools and services developed in TA, NAs and JRAs. Support for extraction of large data volumes of selected in situ data from PSSs and IRSs will be offered. Active WP3, WP4, WP6, WP7, WP9, WP10 and WP13 will be facilitated.
WP11-ID1.1	78	**host the JERICO e-Infrastructure** during the project, including Virtual Research Environment and the associated data, when High Performance Computing and/or powerful access to dat
WP11-ID33.2	80	VA framework that provides access to the most important JERICO Resources (Catalogued) and Pilot Data-to-Products Thematic Services (D2PTS). Services currently offered by the infras **and operate the JERICO RI e-Infrastructure during the JERICO-S3 project**, including the VA portal interface and backend, resources catalogues, access to new data, functionalities ar

		services developed in TA, NAs and JRAs. Support for extraction of large data volumes of selected in situ data from PSSs and IRSs will be offered.
T13.3	86	JERICO-User Committee (JUC, ref. WP9) will be consulted to give feedback on services delivered by JERICO-RI at the hardware level thanks to the return on JERICO-RI TNA experience FP7, 2011), at the virtual level: services access in the e-JERICO-RI initiated prototype phase in JERICO-S3. It will interact with the User panel dedicated to VA.

Table 2: Requirements extracted from the JERICO-S3 Description of Work that are related to JERICO-CORE.

3.3. JERICO-CORE Concept

The general concept of JERICO-CORE has two main data flows (see figure 5):

1) On one side the system collects the information from the JERICO partners through the various European and international infrastructures. Figure 4 shows the flow of information from the various providers through the interoperability module.

2) On the other hand, the metadata information that are recorded in the resource catalogue are used to make available the information that the various types of users require for their own purpose.

There are three main components in this concept:

- **Interoperability layer**: This interface is critical to harvest the information and to provide access to resources that are federated among providers. It is the interface between JERICO-CORE and resource providers.
- **System manager**: After the interoperability module (right side of the diagram in figure 5), the system manager is the orchestrator that uses the information in the catalogue to manage the entire system. This includes user access to the system and to the resources from the distributed partners.
- **Resource catalogue**: The resource catalogue drives the requests and the access to the various resources through the interoperability module.

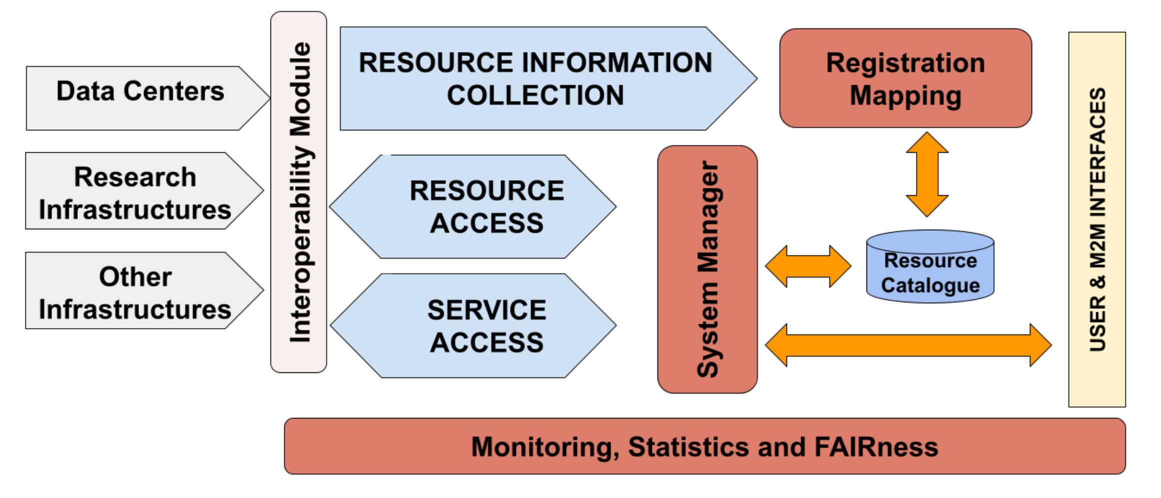

Figure 5: Initial conceptual design overview of the JERICO-CORE prototype.

During the co-design process with stakeholders, we evolved the conceptual design (see appendix D) until the final concept was agreed.

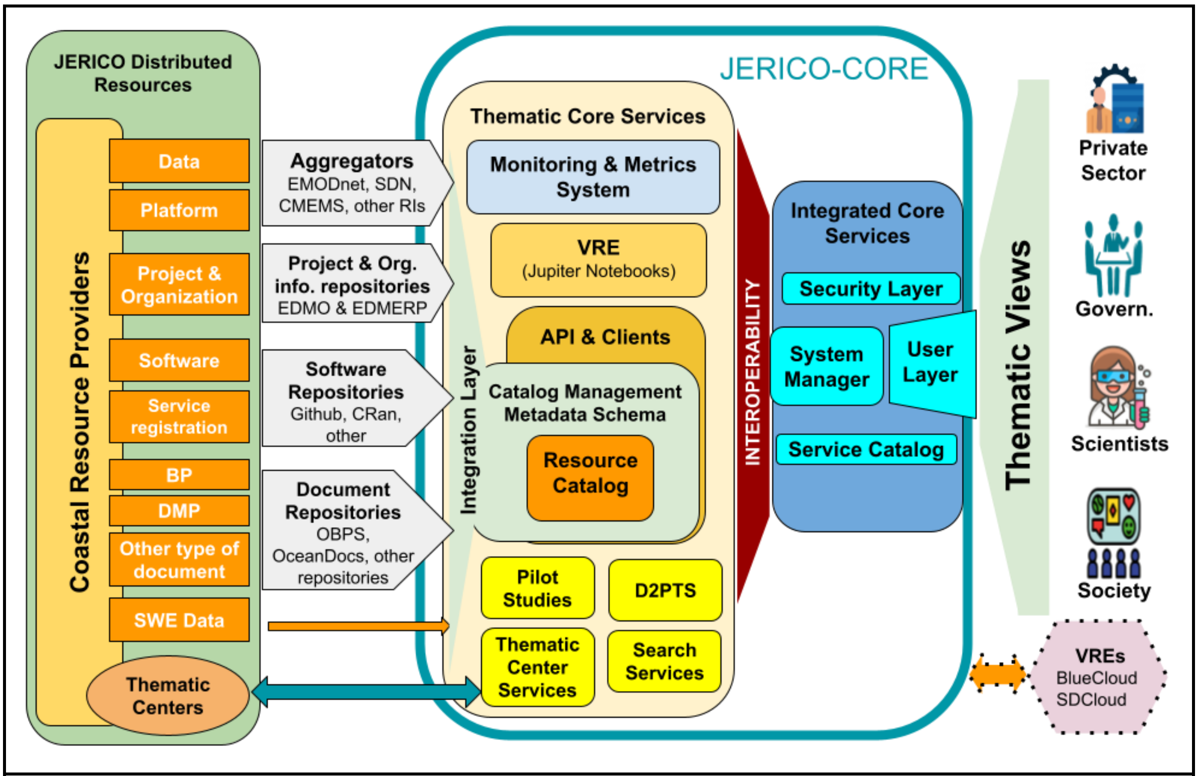

Figure 6: Conceptual model of the JERICO-CORE prototype. The schematic shows the essential components of the e-infrastructure and their relationship.

Figure 6 shows the final concept where the assets are federated among JERICO partners (left side) and exposed to the users (right side). This conceptual view aligns with the information flow explained in figure 4 which shows the details of the flow within the green box on the left of figure 6 representing the JERICO resources that are distributed among partners. Because of the complexity and diversity of these assets and their source, we focused the initial effort of the prototype in collecting information from key information providers that were identified during the co-design process. The information from these providers is stored in the resource catalogue through the integration layer. The information on the provider's side is heterogeneous and uses different types of standards. Knowledge in the resource catalogue is represented as an open graph but it was decided to follow a specific metadata schema. The JERICO-CORE metadata schema is described in appendix E. The integration layer is in charge of linking and mapping the information between both sides. The information from the JERICO-CORE resource catalogue is exposed for users and machines using Application Programming Interfaces (API) or client libraries that facilitate the discoverability and retrieval of resources. The importance in relation to the VRE of Blue-Cloud and SeaDataCloud was clearly identified during the conceptual definition. The SeaDataCloud project was discontinued but Blue-Cloud agreed to a collaboration and an MOU between the two projects was consummated for this purpose. The Blue-Cloud VRE was then integrated into the architecture as well as the follow-on Blue-Cloud 2026 project (see section Planning for more details). Thematic services that are implemented in the JERICO-CORE VRE take advantage of API and client interfaces. They can interrogate the catalogue to find datasets for advanced processing or information to explore assets. Similarly, JERICO-CORE services that are federated among the partner's infrastructures can take advantage of the information stored in the knowledge catalogue. All of these services - distributed or in the VRE - are findable because they are recorded in the catalogue. Mechanisms to register these services in the catalogue were not agreed in the conceptual model but were explored as various options during the design of the architecture.

	Services	Knowledge Catalogue	VREs	API	Searching Capabilities	Integration Layer	UI	Infrastructure
General	D2PTS	Expose Services			Expose Services			
D7.8		Catalog JIIM info		M2M Expose JIIM info				
T5.2.2		Contain IRS/PSS tools			Expose IRS/PSS tools	Harvest software		
T5.4.3		Contain Info for Dashboards	KPI Dashboards		Expose Info for Dashboards	Harvest platform and their status		
T6.2.2		Contain platform Info	Support creation of services for platform management			Harvest platform		
T7.4.4		Contain methodologies and services				Harvest methodologies and services		
T7.5.1		Contain diverse resources				Harvest diverse resources	View resources	Datarmor
T7.5.2	D2PTS							
T9.2.1		Contain services				Harvest services		
T9.3			Operate services					
T10.5.2								Datarmor
WP11		Contain resources						Datarmor

Table 3: Functionalities of each component of the conceptual design illustrated in figure 6 and their importance in responding to the requirements identified in table 2.

The monitoring and metrics system provides status of the e-infrastructure and access metrics to JERICO assets. Additionally, it can also benefit from the machine-to-machine link to the catalogue through these APIs. For example, statistics on the type of resources and their link provides an overview of the health of the metadata among JERICO assets. This is especially important to support other activities of the JERICO-S3 project such as task 5.4 that aims to harmonise Key Performance Indicators of platforms in the IRS and PSS regions. This is done through the definition of Key Platform Performance Indicators (KPPI) and Key Integration Performance Indicators (KIPIs) as shown below and their integration into JERICO-CORE:

- KPPI: "*for monitoring the operations of each platform will be defined taking into account the key issues identified in T5.2.1 and implemented in the functional networking tools of T5.2.2. These KPPIs will focus, first, on the operability and contrast between platforms and instrument payloads, and second, on the implementation of the JERICO Label.*"

- KIPIs: "*will be defined for estimating the integration capabilities of each observatory: number of EOVs/EBVs that could be integrated, per space-temporal scales (mesoscale, sub-mesoscale...), per region and during a specific period (D5.3).*"

The JERICO-CORE concept also includes user interfaces to share knowledge of the catalogue to various types of users. Even though no clear use case was defined that allowed the design of a specific frontend for specific users, we identified various critical components that were common to all user interfaces. Capabilities for adaptation to specific user classes are provided for in the architecture. A security layer would be responsible for providing access to resources depending on licences and agreements. The service catalogue layer would provide through an interoperability layer the access to resources and their metadata that respond to the specific purposes of the interface. This information would be exposed to users through the user layer. These components would be orchestrated by the system manager. It was envisioned that some user interfaces would require other types of components depending on the needs of the user. Because of the endless possibilities of the user interfaces, it was important to decouple the interface part of the system from the knowledge discoverability aspect. This fact, highlighted the need of an API to access the information in a general manner.

Table 3 summarises the way that the conceptual design responds to the requirements shown in the previous section. We observe that all of the requirements are satisfied with the conceptual design agreed with stakeholders. There are some aspects of table 3 that can only be satisfied with the appropriate processes as shown below:

- MS38: the demonstration was carried out in 2022 as described in section Demonstration, Testing and Deployment

- T10.4: the promotion of synergies with existing VREs was done at various levels in particular with EPOS and Blue-Cloud as shown in sections EPOS developments and integration and Service layer.

- WP11: JERICO-CORE needs to be promoted with outreach activities to increase access. Additionally, the operation must include maintenance and operation following the obligations stated for VA in WP11.

- T13.3: the process for collecting feedback related to services needs to be set up.

In summary, this conceptual model was the foundation for the architecture design. Existing technological solutions and e-infrastructures were explored as a foundation. Existing models of international e-infrastructures around the world were analysed to see how they could be re-used for the different parts of the conceptual model. The study of international e-infrastructures and available technologies is explained in the following section.

3.4. Study of Technological Solutions

In order to optimise the design and the implementation of the JERICO-CORE prototype that was planned for 2022 we established synergies with other projects that are operating similar e-infrastructures. Then, we studied other technologies that could be used to complement the available solutions. In this section we describe these two aspects of our preliminary study preceding the architecture design.

3.4.1. Analysis of existing data and knowledge e-infrastructures

For this analysis, we followed the methodology explained in the Studying available solutions section. We studied the following e-infrastructures.

- Enhanced Virtual Research Environment (e-VRE of the VRE4EIC project)
- European Plate Observing System (EPOS)
- Integrated Carbon Observation System (ICOS)
- Data Observation Network for Earth (DataOne)
- Australian Ocean Data Network (AODN)
- Australian Geoscience Data Cube (AGDC)
- European Marine Biological Resource Centre (EMBRC)

We summarised the results of this study in documents specific for each infrastructure (table 4). In these documents, we analysed the properties that we thought important to design an infrastructure that fits the requirements and conceptual design that was agreed with the partners. The matrix in appendix C summarises the important features of JERICO-CORE and the degree in which each infrastructure responds to these criteria. The architecture designer of eVRE (VRE4EIC) and EPOS strongly recommended using EPOS over eVRE due to the current status and future plans of these two projects. From our analysis and discussions with the technical team of the project, we concluded that AGDC is not a good technological solution for the purpose of JERICO-CORE. Therefore, eVRE and AGDC were hereby excluded from the analysis of this section and appendix C.

The criteria of appendix C are organised in five topics:

- Status of the project (see table C-1)
- Support of the distributed running services (see table C-2)
- Visibility of and Access to Distributed and Centralised Resources (see table C-3)
- Data Management Services (see table C-4)
- Architecture, Standards and Concepts (see table C-5)

Name	Document
Enhanced Virtual Research Environment (eVRE)	IREP_EXT_eVRE_Framework_JERICO-S3
European Plate Observing System (EPOS)	IREP_EXT_EPOS_Infrastructure_JERICO-S3
Integrated Carbon Observation System (ICOS)	IREP_EXT_ICOS_Infrastructure_JERICO-S3

Data Observation Network for Earth (DataOne)	IREP_EXT_DataOne_Infrastructure_JERICO-S3
Australian Ocean Data Network (AODN)	IREP_EXT_AODN_Infrastructure_JERICO-S3
Australian Geoscience Data Cube (AGDC)	IREP_EXT_AGDC_Infrastructure_JERICO-S3
European Marine Biological Resource Centre (EMBRC)	IREP_EXT_EMBRC_Infrastructure_JERICO-S3

Table 4: List of documents containing the technical descriptions of infrastructures

We describe in the next sections the conclusions of the analysis for each of these criteria.

3.4.1.1. Sustainability and scalability

The level of maturity and sustainability of these infrastructures is high. Their long-term continuity of their software development and support is critical if we adopt these solutions for the ESFRI roadmap. The possibilities for collaboration from a personnel point of view as well as from the availability of their software code are also comparable. Their technical leaders have been very responsive during the process and, in particular, the EPOS and AODN availability support was outstanding. The technical documentation of EPOS and DataOne are detailed in contrast with the technical documents from ICOS and AODN which are very basic. However, all of them have been very open to providing technical information when requested. From an architectural perspective, these infrastructures are all modular by using stand-alone services and dockerization. EPOS and DataOne added to their design monitoring systems of their infrastructures.

3.4.1.2. Management of federated resources

DataOne, ICOS and AODN are mainly focused in data management and dissemination. All three infrastructures provide tools to store data and/or metadata that are interesting to support services that require synchronising data with remote servers. ICOS data repositories and services are centralised and are not capable of handling federated resources. ICOS provides some capabilities to create workflows and share common python code through the Jupyter Notebooks. ICOS also includes a catalogue of resources that use non-homogeneous and non-standard ontologies. Using custom ontologies is usually an obstacle to integrate to the international community.

DataOne is focused on management of distributed data but not federated services. However, they plan to integrate distributed services in future developments in the context of the WhaleTales project. This functionality could be of interest to JERICO in the ESFRI roadmap but for the moment DataOne only provides some centralised tools via the Investigator toolkit. An interesting feature to highlight from DataOne is their effort to measure FAIRness of data along the entire data cycle.

AODN also has a very complex system allowing integration of distributed data infrastructures with a large diversity of characteristics. AODN integrates remote data services by exposing interfaces to web services such as Web Map Services (WMS) and Web Feature Services (WFS). They provide some very interesting services for ocean data that expose their metadata catalogue via Geonetwork (requires ISO19115/119/110 standards or Dublin Core) and Geoserver based on WMS, WFS, Web Coverage Services (WCS). The capabilities of

AODN to integrate remote services using the web services approach could be adopted to benefit from some specific services from AODN such as watch services or geo services.

EMBRC focuses on registering services that are associated with their datasets. They adopt existing services and components that can be provided on a national level or by related European e-infrastructures, such as LifeWatch, ELIXIR, EGI, INDIGO-DataCloud, EMODnet, EurOBIS, WoRMS and Euro-BioImaging.

EPOS' approach to knowledge management differs from the other infrastructures in the sense that they focus on exposing services and data that are federated among partners. These services return information about data and other resources. They are registered in a postgres database using The Common European Research Information Format (CERIF) standard which is compatible with other important standards such as ISO19115/119/110 standards or Dublin Core. CERIF is the comprehensive information model for the domain of scientific research. It is intended to support interchange of research information between and with the Current Research Information System (CRIS). EPOS does not use yet the full potential of CERIF to include all sorts of information (e.g. documents, equipment, organisations) but focus on registering services and their basic information.

3.4.1.3. Machine-to-machine and user interfaces

Diverse mechanisms can be used by users and services to access the information of the resource catalogue and to communicate with the available resources. User interfaces allow humans to interact with these infrastructures to facilitate use of the available services and access to the available outputs and other resources. On the other hand, machine to machine (M2M) allows direct communication between services using specified interfaces. These communications can be accomplished using API which are a type of software interface, offering a service to other pieces of software. In this particular case, API allow the discoverability and access of information recorded in these infrastructures as well as running services automatically. We highlight the following type of interfaces:

- Representational State Transfer (REST) or **RESTful API**: API that conforms to the guidelines of REST architectural style such as managing requests via HTTP, stateless client-server communication, uniform interface between components, …
- **SPARQL API**: API that offers query, update and the graph store protocol (GSP). The SPARQL Protocol and RDF Query Language (SPARQL) is a query language and protocol for RDF.
- **Web Services**: Web Map Services (WMS) and Web Feature Service (WFS) provide protocols to deliver map images in png, gif, jpg or GIS formats.

All these infrastructures except for EMBRC provide machine-to-machine capabilities. AODN uses interfaces between web services and all the others provide a RESTful API which allows a flexible interface for external requests. ICOS also provides a SPARQL interface which is an important piece to link the resource catalogue to the semantic web. They all provide a single portal user interface, often with visualisation tools. All of them have a security layer implemented and they have resilience mechanisms for their data and metadata such as B2SAFE of EUDAT (https://www.eudat.eu/b2safe).

3.4.1.4. Integration into JERICO-CORE

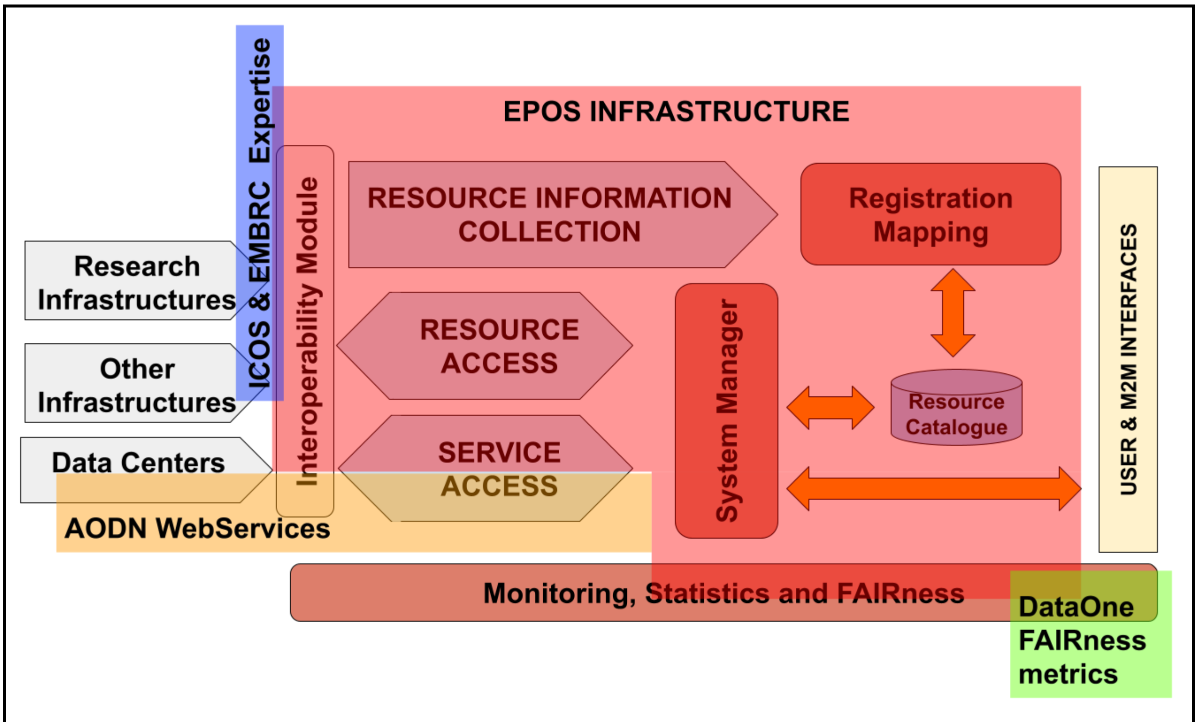

Figure 7: Illustration of the potential integration of infrastructure components in the conceptual design of JERICO-CORE

We used the concept overview in figure 5 to illustrate the conclusions of our analysis. Figure 7 shows how the infrastructures that are studied in this report are solutions to the components of JERICO-CORE concept. It clearly shows that EPOS is best suited for most of the components that were identified for JERICO-CORE. They have done an important effort on selecting a standard for managing and exchanging research data. Moreover, they have defined the interfaces needed to interact with the various services that are distributed among the partners. This includes interfaces to register partner's services in the main catalogue and interfaces to interrogate and display the information returned by these services when they are interrogated. Additionally, they have done an advanced user interface to display information and data. The system includes authorization and authentication with a large number of options including a VRE4EIC AAAI security layer. These systems are orchestrated with two managers for registration and for user requests.

From a strategic point of view, adopting EPOS provides strength to the European community by improving the collaboration between European partners and reuse of resources that have been funded by the European commission. Technically, we selected EPOS as the baseline for JERICO-CORE initial developments for the following reasons:

- Strong focus on resource catalogue centred design in line with JERICO-CORE strategy
- Well advanced work around comparing schema and selecting a catalogue standard, CERIF
- Architecture handling distributed resources and thematic services as for the context of the JERICO community

- Apparent robust, modular and flexible architecture that can be adapted to the JERICO-CORE needs
- Demonstrated interest in working with JERICO which can provide considerable support in the framework of a well-defined collaboration
- Re-use of work and enrich collaboration of European funded projects in order to enhance European research and infrastructures
- Connecting cross-domain infrastructure to enhance collaboration and data sharing across fields

There were some features that were not covered by EPOS that were required by JERICO-CORE:

- Automatic collection of information among partners that required a complex procedure with a human in the loop
- Collection of information of resources other than data and services
- Machine-to-machine endpoints such as Restful or SPARQL APIs that allow advanced searching capabilities of oceanographic resources. This complements the EPOS API to access the catalogue of registered services.
- Monitoring and metrics helping to define the health of the assets in the catalogue in relation to the standards in oceanography. EPOS has a metadata ingestion pipeline through EPOS-DCAT-AP with SHAPES/SHACL validation. EPOS also monitors 'liveness' of services (by 'pinging') and the usual monitoring of accesses, return to page, length of time on each page,... The monitoring and metrics system should update the features provided by EPOS to respond to the needs of JERICO partners.
- A VRE to allow users to (co-)develop and integrate services
- Interfaces to international or local coastal ocean knowledge providers

Additionally, the other e-infrastructures that were considered in this analysis provide some interesting features for JERICO-CORE. The mechanisms of AODN to handle federated services is an interesting approach for advanced service integration. It provides a set of interesting services for managing and processing data. AODN also deals with the data and metadata standards that are compatible with the European aggregators such as EMODnet, Seadatanet and Copernicus Marine Service. The pieces for metadata integrity and data QC from AODN could also be included in JERICO-CORE. Secondly, it would also be of interest to implement the mechanisms of DataOne to measure FAIRness as a way to define the quality of the data and related resources. ICOS and EMBRC expertise was interesting to understand the difficulties of collecting heterogeneous information from different sources.

Despite the advanced functionalities of these infrastructures, there was an important gap in the solutions to collect the information from providers of ocean resources. Therefore, it was necessary to investigate additional solutions to store and share coastal ocean knowledge. The study of complementing technologies for JERICO-CORE is shown in the next section.

3.4.2. Analysis of complementing technologies

From the analysis of existing e-infrastructures, we identified a series of gaps that these digital ecosystems do not cover. In particular, there are four aspects that were not completely addressed by these solutions. The main one is the processes to collect and store information from a complete range of ocean knowledge providers. These infrastructures focus on a specific set of providers and do not try to relate assets from different sources. The second aspect is the VRE. These infrastructures barely provide a complete environment for developing collaborative tools in an integrated manner with the knowledge catalogue. In addition, with these features, we also explored technological solutions to facilitate the operation and the monitoring of the infrastructures. In this section, we explore these aspects: harvesting, storage, VRE and supporting software.

3.4.2.1. Harvesting

Harvesters are services aiming to collect, prepare data, information and metadata and store them in a knowledge catalogue to allow persistence and discoverability of asset information. The process of harvesting was approached following the classical process of Extract, Transform and Load (ETL). In this three-stage process the information is collated from one or more sources in an automatic way with manual or recurring scheduled processes. We explored how to design JERICO-CORE harvesters to properly extract information from providers and guarantee the validity of the data source. Different information providers use different technologies to expose their knowledge. We created some preliminary code to study the performance and the specificities of each method protocol for accessing provider's information. We found the following general cases:

- **Standard software solutions and data servers**. For these providers we estimated that the extraction algorithms of the harvesters could be shared between multiple providers that use the same technologies. Only the mapping of the information needed to be adapted for the specific source of information. We studied the following solutions that are used by EMODnet physics, SeaDataNet and Copernicus Marine Service to serve datasets or OBPS or Aquadocs to serve documents.
    - **ERDDAP**: Software solution to serve datasets in common file formats such as Network Common Data Form (NetCDF).
    - **Thredds**: Software solution that provides virtual directories of available data and their associated metadata and serves them using OPeNDAP, OGC WMS and WCS, HTTP and other remote data access protocols.
    - **DSpace**: Software solution to create document repositories as a digital archive system focused on the long-term storage, access and preservation of digital content.
- **Services using Open Geospatial Consortium (OGC) standards** to represent and exchange geospatial information. Geoserver and mapserver are common solutions among the scientific community and they are extensively used in the private sector. We studied the following standards that are used by EMODnet physics and biology.

  - **Web Map Services (WMS)**: protocol used by map servers to deliver map images usually in raster tiles such as png, gif or jpg.
  - **Web Feature Service (WFS)**: protocol used to communicate geographic feature information in a true GIS fashion.

- **Services using SPARQL Protocols and RDF Query Language (SPARQL)**. We analysed services retrieving and manipulating information "subject-predicate-object" triples, i.e. a statement linking one object (subject) to another object(object) or a literal, via a predicate. The case studies for this analysis were the services provided by SeaDataNet though the European Directory of Marine Organisations (EDMO) and the European Directory of Marine Environmental Research Projects (EDMERP).

- **Custom API access**. Some providers have their own interface specifications. The logic of the harvesters collecting information through these interfaces is custom for that purpose. We explored the github and The Comprehensive R Archive Network (CRAN) APIs.

- **Sources of information without interfaces**. This scenario was necessary to complement information that was not accessible in any server provider. We explored processes to load information off-line from binary or text files such as Excel spreadsheets, JavaScript Object Notation (JSON), Terse RDF Triple Language (TTL) or JSON Linking Data (JSON-LD) files. We anticipated that this would be necessary to load platforms from EMODnet physics or Coriolis, software metadata from repositories and in general complementary information that is not represented by any provider.

There are two aspects to consider in the processes applied to large volumes of data. On one side, we focused on the performance, scalability and computation efficiency of the process. On the other, we considered the analytical aspect of the information itself for understanding and using the information. For this latest study, we analysed these processes from the perspective of the seven V's of big data:

- **Volume**: the capacity of managing large volume of data
- **Variety**: the ability to manage a large variety of types of data sources
- **Velocity**: the speed at which we process and access information
- **Variability**: the capacity of understanding and interpreting information that change constantly
- **Veracity**: the capacity to measure the accuracy of the information and their relationship.
- **Visualisation**: the ability to to represent the information in a meaningful way to understand the status of JERICO's assets.
- **Value**: the response to the needs of JERICO in particular related to the discoverability, the support to services and the JERICO label

In relation to the volume of data, we came to the conclusion that the volume of information to manage is not an impediment for the providers that we selected. The velocity should be considered for applications that require immediate access to real time data. For example, this is the case for emergency response applications. The collection of SWE sensor metadata may also requires a high velocity performance but in most of the cases, the amount of metadata from SWE sensors is small. In the context of this prototype, we did not consider velocity as a major hindrance of the design.

For each source of information, we concluded that the information is structured within each provider and it could be processed and stored in a fixed format. This facilitates the ingestion of resource metadata from each provider. However, we can say that the global knowledge as a whole coming from all the sources is semi-structures since it contains data that lacks consistency between different sources. In fact, the major cause affecting the speed is not technological. The variety and variability of the information makes it difficult to automatically collect and transform the information without a human in the loop. This makes the harvesting costly and time consuming. Veracity is affected by the fact that often no standard Personal Descriptor Identifiers (PDI) are used. Instead, providers often use custom identifiers or strings describing the metadata. For example, the name of a person named Brian Harold May could be represented in multiple ways such as B. May, B. H. May, May B., May Brian... The ideal scenario would use his ORCID id (https://orcid.org/0000-0003-3133-6749) to represent him as an author, publisher or other role. The same is true for organisations, the EDMO codes provided by SeaDataNet services (https://www.seadatanet.org/Metadata/EDMO-Organisations) must be used to identify the role of an organisation in order to link their assets. Unfortunately, this ideal approach was often not followed.

We identified the risk of lack of veracity and the large variability and variety of information. We explored various tools to visualise the graph of knowledge. The knowledge graph creates a digital twin of the JERICO-RI assets. It enables us to represent all of the assets of JERICO in a holistic view and use this knowledge to evaluate the status of a system or existing activities. For this purpose, it is critical to select the storage component adequately to technically allow the usage of tools compatible with graphs. We describe next the analysis that we performed related to the storage solutions.

3.4.2.2. Storage

We based our study on the results from ENVRI-FAIR [3]. They explored the range of existing approaches to store knowledge graphs. They identified four types of storing solutions:

- Relational database management system (RDBMS)
- General purpose graph databases
- RDF triple stores
- Virtuoso

In our study, we used the results obtained from ENVRI-FAIR and complemented the analysis with some tests using JERICO-RI assets.

RDBMS:

Among the RDBMS, we explored the CERIF standard. CERIF is a comprehensive model to represent all scientific domains and it can be adapted to oceanography. It is a representation of a fully connected knowledge graph often implemented (but not always) over a RDBMS. It uses n-tuples rather than triples for referential and functional integrity, for efficiency, and for increased expressivity in the linking objects. We evaluated CERIF over RDBMS and the flexibility of this solution to represent the information from the providers as well as the accessibility. CERIF was a very flexible representation of almost any information that we could need to represent JERICO assets. However, the insertion of the information was laborious because in many cases it required the expansion of the basic schema in order to represent this information. This also came at a price when searching for information which was needed during the harvesting of new entities from various sources. A more detailed analysis of the CERIF standard is described in the Integration into JERICO-CORE section.

Graph databases:

Graph databases as for triple stores are designed to store linked information. They both focus on the relationship between assets and thus, are well fitted for the purpose of JERICO-CORE. We analysed two cases of graph storage. Neo4J (https://neo4j.com/) and JanusGraph (https://janusgraph.org/) are scalable graph databases that are optimised for storing and querying big data. In particular, Neo4j is the most widely adopted solution in the industry. We concluded that graph solutions, in particular Neo4j, perform extremely well and they respond very well to the needs for Volume and Velocity. Additionally, the graphical interface of Neo4j is friendly and clearly supports Visualization and analysing the Veracity and Value of JERICO assets. Additionally, graph databases are usually more versatile than RDF triplestores and can store various types of graphs such as undirected graphs and weighted graphs. They are optimised for graph traversals. The drawback for these solutions is that they do not have native support for RDF or SPARQL.

RDF triple store:

The main advantage of RDF triple store is that they integrate with the semantic web and the standardised universe of knowledge such as DBPedia. This functionality is a feature that we strongly valued at the beginning of the project because it provides the opportunity to integrate to other sources of knowledge such as the ones provided by SeaDatanet about organisations, projects and observation programs. In fact, it provides the basis for reasoning about the meaning of an RDF expression and inference on data. Additionally, EPOS adopted TTL documents to ingest asset information in their infrastructure. TTL is a syntax and file format for expressing data in the RDF data model. Therefore, serializing RDF graphs into TTL is straightforward because it was designed for that purpose. Concerning RDF triple stores, we explored Apache Jena and rdflib.

Virtuoso:

Virtuoso (https://virtuoso.openlinksw.com/) is a multi-model hybrid-RDBMS that supports management of data as relational tables or property graphs. It combines a series of features including document and RDF store and a series of DBMS for graphs, native XML and relational. It is compatible with web technologies. We discarded this package in favour of free solutions. It is nevertheless an option that should be considered for the JERICO-DS design. It is recommended to use the ENVRI-FAIR analysis of Knowledge Base [3] and the lessons learned from their demonstrator.

3.4.2.3. Graph Visualisation

We were aware that visualising the graph correctly would help to clarify the status of the complex and large amount of JERICO data and thus, make sure to understand the value for the consortium. By visualising the graph of knowledge, it is easy to find for example resources that are not connected to others, clusters of resources, gaps in the description of resources. In fact, when unique identifiers were not used adequately we noticed duplicity of assets. For example, when datasets did not contain a standard identifier for platforms or organisations. These showed in the knowledge graph as a bubble representing the platform or the organisation that is only connected to a dataset.

Using visualisation tools to analyse JERICO assets is clearly beneficial and this can be supported by smart algorithms. While these tools are out of the scope of the pilot, we assessed some of them as the foundation for the JERICO-DS design. The visualisation tools are strongly linked to the selected storage solution which is an important aspect that we investigate in this phase. Due to the limitation of the scope of the project, we did not implemented any of these solutions but we studied the following ones:

- **D3SPARQL** (https://github.com/ktym/d3sparql): We evaluated this library as the primary alternative to embed a graph representation of the resources displayed in the EPOS UI, querying the information via Jena's SPARQL entrypoint . It's based on the popular visualization library D3.js.
- **D3.js** (https://d3js.org/): Excellent and powerful generic framework to represent all kinds of graphics. We evaluated this framework as an useful alternative to represent resources in a graph view into EPOS UI.
- **Graphviz** (https://graphviz.org/): Open source graph visualisation library very popular. We discarded it because it requires to export Jena RDF resources to native Graphviz format (.dot)
- **Protégé** (https://protege.stanford.edu/): Open source ontology editor with powerful features such as SparQL query parser, import facilities from the most popular serializable RDF formats and visualisation tools.
- **Gephi** (https://gephi.org/): The Open Graph Viz Platform written in python, focused on the analysis of networks. We consider this tool as relevant and useful for the project due to the existence of multiple plugins that allow exporting the graph to other formats such as GraphML, .dot compatible with Graphviz, js compatible with D3.js or csv.
- **Sigma.js** (https://www.sigmajs.org/): Modern JavaScript library for rendering and interacting with network graphs in the browser. It works in symbiosis with graphology, a multipurpose graph manipulation library. Useful to use in combination with Gephi to build an interactive graph viewer.
- **Cytoscape** (https://cytoscape.org/): Open source network and visualisation framework with data import capabilities from multiple formats such as GraphML or .xslt

- **Neovis.js** (https://github.com/neo4j-contrib/neovis.js?): Graph visualisation library to represent data in Neo4J.

- **Popoto.js** (https://www.npmjs.com/package/popoto): Graph visualisation library built upon D3.js to represent data in Neo4J.

3.4.2.4. VRE

We considered two options to respond to the need of a VRE. The first option was to deploy a JupyterLab (https://jupyter.org/) to allow users to collaborate in the creation of services. This option is simple and it decreases the effort of creating a VRE to the strict minimum. However, it lacks the features that other solutions allow such as having workspaces or deploying code for operational mode in python or other languages. For this reason, we explored the solution that Blue-Cloud offers. Blue-Cloud is a VRE to manage services and datasets. It is the theme of the European Open Science Cloud (EOSC) to support researchers to better understand the ocean. It is based on D4Science e-infrastructure which implements proven solutions for connecting to external services and orchestrates distributed services. We had a series of meetings with the Blue-Cloud and D4Science teams to better understand their technologies and the ways that JERICO-CORE could integrate to Blue-Cloud. Among other features, Blue-Cloud offers the possibility of creating Jupyter Notebooks in an environment that includes the Data Discovery & Accessibility Service (DDAS) to facilitate the access to datasets. This, in combination with the JERICO-CORE catalogue will facilitate the discovery and accessibility of datasets using advanced knowledge in relation to other resources. Blue-Cloud helped us testing their solution by creating a JERICO-RI Virtual Lab (VLab) in their D4Science Blue-Cloud gateway (https://blue-cloud.d4science.org/group/jerico_core). We used the JERICO-RI VLab to experiment in the creation of services as we will explain in the section Service layer.

3.4.2.5. Supporting technologies

In addition to the technologies used directly to provide the solutions responding to the requirements, an e-infrastructure requires technologies supporting development, deployment and operations. Nowadays, development and deployment of code have become increasingly agile due to user demands. This agility was important due to the high demand of finishing the JERICO-CORE prototype in a short period of time. Changes pushed to software repositories need to be frequently integrated in the operational environment. Continuous Integration (CI) is the practice of integrating new or updated code into the repository and building automatically in every commit. In this process, code can be tested to validate the new updates. Continuous Deployment (CD) is a step forward to CI in the sense that changes are frequently and automatically deployed. By combining CI/CD we were able to optimise the delivery of new software functionality. CI/CD is a critical component of DevOps that helps automatization. In fact, DevOps is a set of practices that aim to shorten the development time cycle while preserving the software quality. We incorporated these practices for the development and deployment in the test environment of JERICO-CORE. For this purpose, we incorporated in our development chain a set of technologies that we describe below:

- **Gitlab** DevOps software package: this application integrates software repositories with development, testing and deployment support. It facilitates collaboration as a complete DevOps platform. We compare this to the use of other git solutions. We installed and set up a server with Gitlab for JERICO-CORE development and testing processes.

- **Docker/Kubernetes**: Docker is a set of Platform as a Service (PaaS) products that help deliver software in the DevOps processes. It uses small services called containers that host the software through a host called Docker Engine. It facilitates the use of specific libraries in a custom made environment and the management of the dependencies. We incorporated docker into our CI/CD workflow to build docker images that are made available in the Gitlab Container Registry. Additionally, jobs are run seamlessly in local machines, testing environments or production servers. The workflow is explained in more detail in section Development and deployment procedures. Additionally, we explored Kubernetes as a possible solution to orchestrate JERICO-CORE containers. Kubernetes is a tool to facilitate development, scaling and management of software that is deployed using dockers. We explored that option but it was not prioritised for the JERICO-CORE prototype in order to reduce the complexity of the infrastructure set-up of the development and testing environments. It can be considered in the context of JERICO-DS.

- **Monitoring**: We explored various technologies to monitor JERICO-CORE performance and error tracking. Among others we explored Sentry, Zabbix and UptimeRobot. These technologies were considered for the development and testing environments. Additionally, we explored other technologies for the access metrics system framework of JERICO-CORE that was used under WP11 to support Virtual Access [2].

- **Logger**: The main solution that we explored for logging was the Elastic stack (ELK) which is a popular technology for managing big data. Other available solutions that complemented ELK and as an alternative were Fluentd, Prometheus and Graphana. In particular ELK is used in the Virtual Access Metrics System (VAMS) of WP11 [2]. ELK includes, among others:

    - Kibana: a web application that specialises in managing ElasticSearch databases and creating reports of the data they contain.

    - Logstash: a very flexible microservice that takes a series of events and modifies the data of each event according to a configuration. It can remove unnecessary or private data, it can discard events and it can transform and enrich the data in the original event (provide approximate geolocations, categorise URLs…).

    - Filebeat: a piece of software whose job is to monitor files and detect new lines each time they get added. It can perform basic filtering of the data before transmitting it over time to Logstash or ElasticSearch.

- **Configuration MGR:** HashiCorp Vault was explored to handle configurations in a secure manner using encryption services for authentication and authorization.

In summary, the preliminary study of e-infrastructures and complementary technologies were essential to understand all the pieces of JERICO-CORE. Among the available solutions of existing e-infrastructures, it was decided at the beginning of the project to adopt EPOS as the baseline for the developments. In parallel with the study of these infrastructures we explored the limitations of the knowledge collection process. This study included small proof of concept of harvesters, storing solutions and the use of visualisation tools. Additionally, package solutions to facilitate the development and the deployment processes were put in place. Lessons learned and the basic setup were the foundation of the architecture design that we describe in the following section.

3.5. Architecture Design

The architecture design (see figure 8) includes a series of layers to respond to the defined conceptual design shown in figure 6 as follow:

- **Harvester Ecosystem**: network of providers of resource information that form the integration layer of the conceptual design. It is in charge of gathering and storing the information from the providers.
- The **RDF Queue Layer** and **RDF Queue Ingestor Layer**: facilitate the ingestion of new information to avoid collision between various sources of knowledge. It is not represented in the conceptual design but they are key components between the integration layer and the resource catalogue.
- **Data Storage Cluster**: constitutes the storage solution of the resource catalogue. With a top layer it would potentially manage various types of storage simultaneously such as graph or rdf triple store.
- **Monitoring Layer** and **Auxiliary Service Layer**: supports the supervision of the status of the e-infrastructure and the communication between components. It responds to the monitoring and metrics system of the conceptual design.
- **Service Layer**: Integrates services such as D2PTS or other Thematic Services into JERICO-CORE. Blue-Cloud is the solution for the VRE but this layer manages the interaction of services developed in other environments and that are federated among JERICO partners.
- **EPOS Integration**: manages the information and the available services to be integrated in the EPOS UI. It includes the components of the Integrated Core services of the concept.
- **User Service Layer**: includes the different visualisations that are customised for the various types of users. The EPOS UI is one among others and allows to represent the information in the catalogue in a simplified manner.
- **API Layer**: facilitates machine-to-machine communication. It was identified in the conceptual design as a key component to help make automated tools and to decouple the front end from the Data and business layers. Additionally, it acts as a service provider for the EPOS UI to represent the information in the resource catalogue. This corresponds to the interoperability layer of the conceptual design.

Collecting and integrating information asynchronously from multiple heterogeneous sources in a single catalogue of linked knowledge is complex to centralise without well-defined and implemented metadata standards. In the context of JERICO-S3 where there was no previous metadata schema linking all the resources. The harvester ecosystem of the JERICO-CORE prototype was initially planned as a peer-to-peer (P2P) architecture with a network of equipotent nodes of providers (see figure 9). Peers make their metadata information available to the rest of the nodes without the need for a central server. Providers would be equally suppliers and consumers of resource information. Thus, the need for a development framework that creates an application layer over TCP/IP to facilitate the development of nodes that communicate with each other. This Software Development Toolkit (SDK) is a layer that is shared among the API, the node provider and the ingestion layer. Each node will contain a harvester with the following components:

- A P2P network component to manage the interaction with other nodes
- An RDF queue client to interact with the RDF queue layer
- A scheduler to program the times of harvesting
- The SDK that contains the framework to interact with the other components (API, storage, monitoring,...)
- The specific logic of the harvesters that is customised for the provider

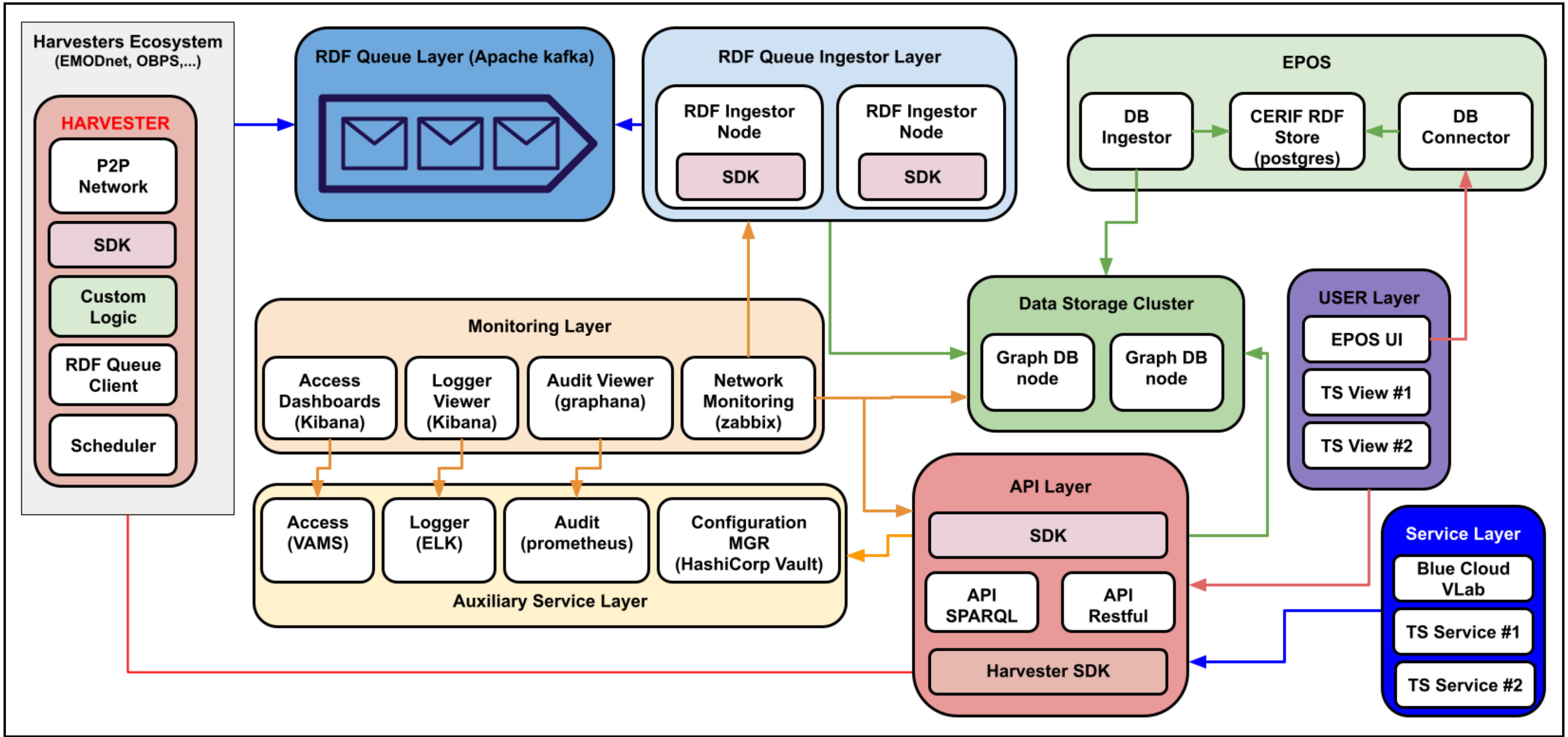

Figure 8: JERICO-CORE software architecture design

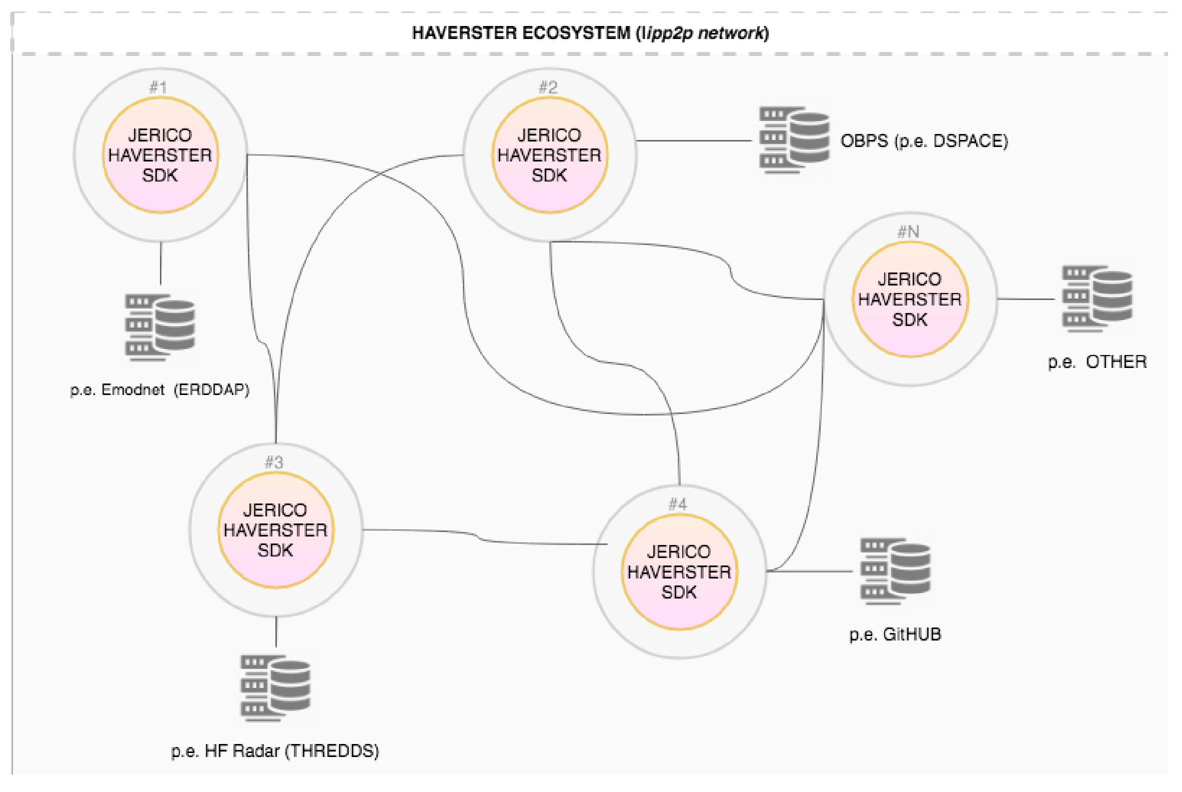

Figure 9: Illustration of the harvester ecosystem based on the P2P network.

The SDK is a key component of the architecture that supports the integration of the harvesters to the various components of the system (see figure 10). It contains the necessary libraries to allow the node to seemingly connect to the various layers.

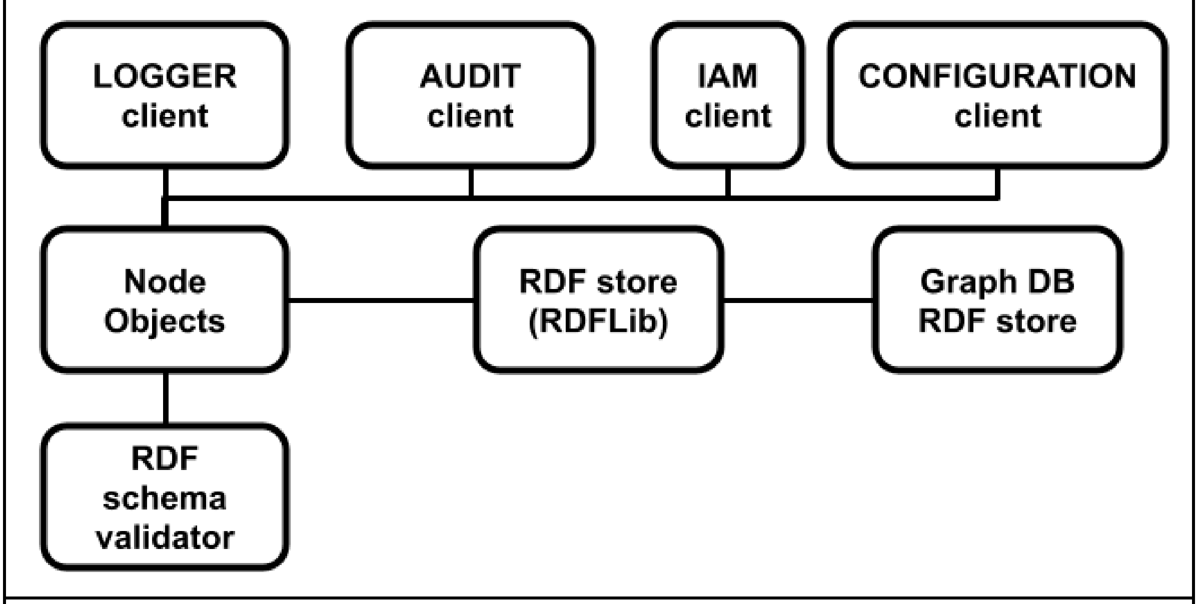

Figure 10: SDK libraries needed to connect the components of the JERICO-CORE software architecture (see figure 7)

In particular, one critical component is the RDF schema validator which guarantees the integrity of the available knowledge. When exploring technologies we identified a lack of standard ways to represent JERICO metadata among providers. Therefore, we adopted for this prototype the metadata schema of EPOS (https://github.com/epos-eu/EPOS-DCAT-AP) as the reference for all the providers metadata. We modified and added entities and attributes when needed to handle the JERICO resources. In fact, as mentioned earlier in section Analysis of existing data and knowledge e-infrastructures, is mainly centred on registering dataset and service information. One of the major additions is the inclusion of JERICO entities (see appendix E)

The disuse of standard identifiers among some providers was a hindrance of collecting from federated providers. During the preliminary study in section Analysis of complementing technologies we identified the diversity and often the lack of consistency of identifiers even within the same provider. We designed a system to manage and map identifiers that was improved in each run of the harvesters either with specific identifier match or with algorithms that allow the application of rules to improve the relationship between different entities. The Alias Management System (AMS) is a critical component of the Queue Ingestor Layer that helps improve the knowledge graph of the JERICO assets (see figure 11). AMS stores identifiers local to each provider and matches them to the more standard identifiers or to the ones created for the JERICO catalogue. In this process, information that is recorded to the catalogue is first compared with the available information. The resource is updated when available. Otherwise, an alternative identifier is searched so the new information is matched in the catalogue with another name. When the information is neither in the alias database, the information is input as a new entity.

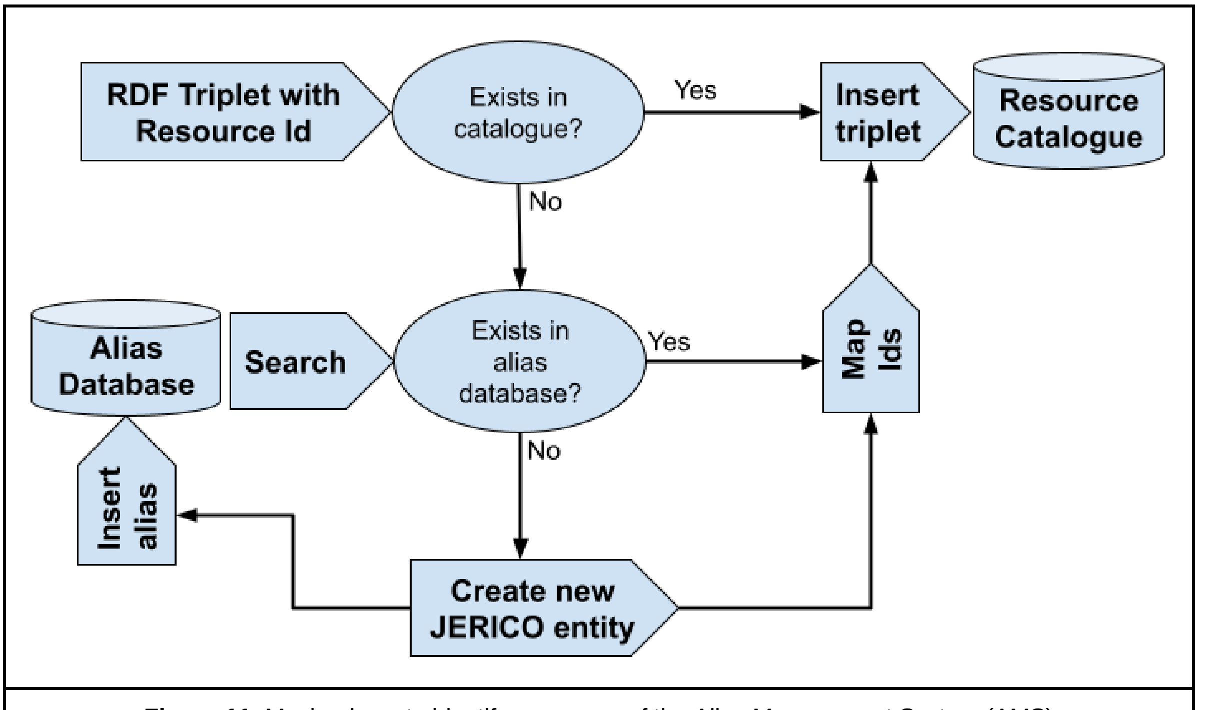

Figure 11: Mechanisms to identify resources of the Alias Management System (AMS)

In this architecture, there are two external pieces that are critical:

- EPOS provides the environment to visualise and browse the resource catalogue
- Blue-Cloud provides the VRE that is based on D4Science platforms

We elaborated and signed a Memorandum of Understanding (MoU) with each of these two infrastructures. These MoU allows us to integrate a part of their system in our architecture. We explain below the means of the integration of EPOS and Blue-Cloud to JERICO-CORE.

EPOS:

Figure 12 shows EPOS architecture. The compatibility and interoperability layer allows EPOS to collect the information of the thematic services that are integrated to their central hub as well as to communicate with these services and request information to be represented in their UI. The integration to JERICO-CORE required us to consider these two aspects.

- **TS registration**: EPOS register services to the CERIF catalogue using TTL files. The information of JERICO assets is contained in the Data Storage Cluster (see figure 8) and we provide the mechanisms for the EPOS system to retrieve the information directly in order to optimise the process.
- **TS communication**: JERICO assets are registered as services. We created an endpoint in our API as the service that provides the information of these assets in GeoJSON format which is the one required by the EPOS system.

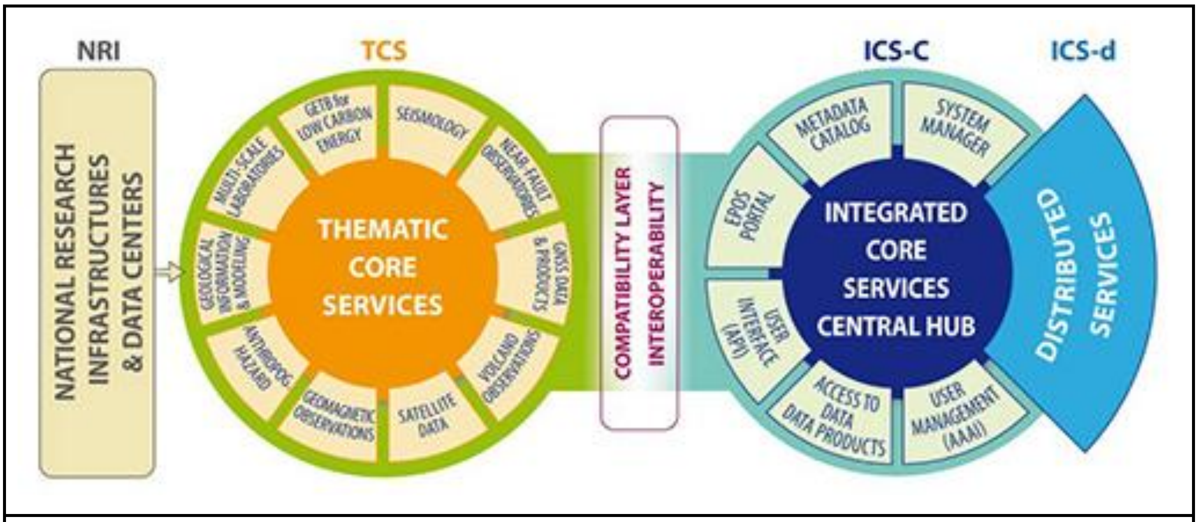

Figure 12: European Plate Observing System (EPOS) e-infrastructure architecture.

The integration into Blue-Cloud also has two aspects. One is related to the creation of tools that process advanced products with added value with respect to the existing data of JERICO partners (see figure 13). The second aspect is the creation of visualisation tools that support specific users to exploit the added value products (see figure 14).

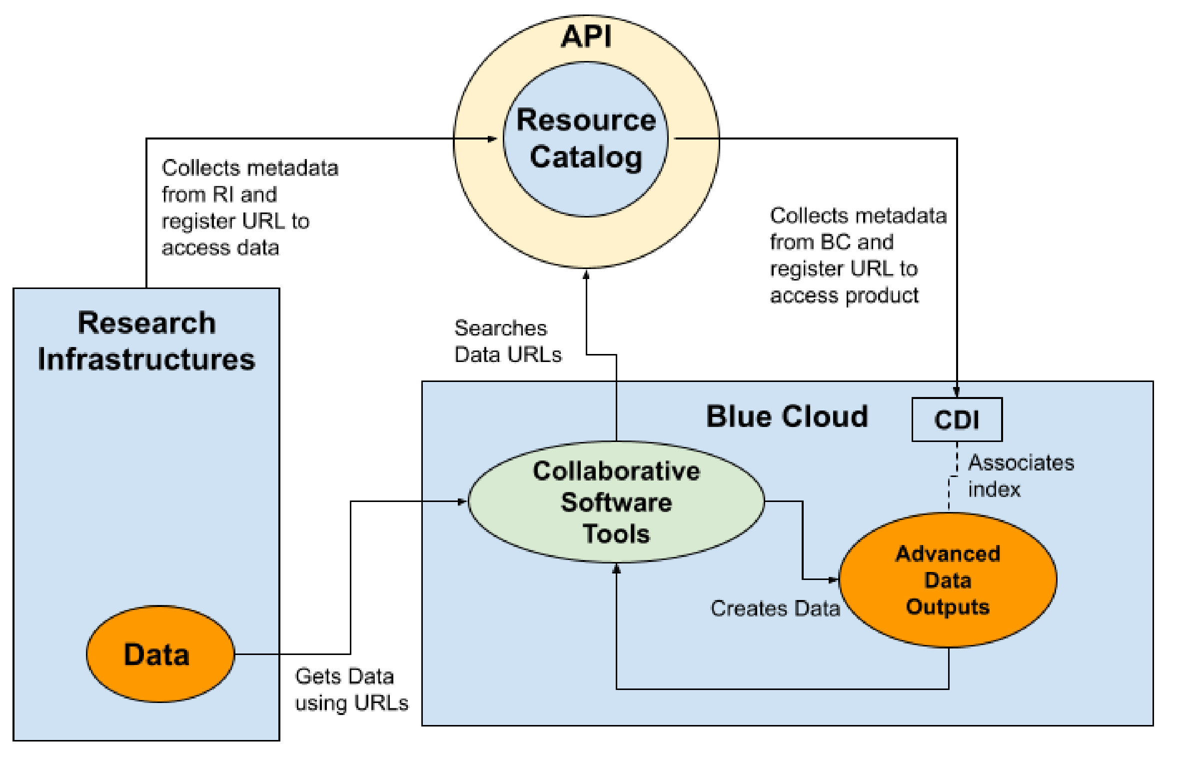

Figure 13: Integration of JERICO-CORE services in the Blue-Cloud VRE. The diagram shows the role of the resource catalogue and the integration to services that are hosted in Blue-Cloud.

Ideally, Data To Products Thematic Services (D2PTS) should be integrated with this workflow. These services and tools use data from different RIs and other federated sources. This data is discovered and accessed using the resource catalogue and API of JERICO-CORE or the Data Discovery and Access Service (DDAS). D2PTS and other collaborative tools are developed in the Blue Cloud VRE. The resulting advanced products are made available in the VRE and their information recorded in the JERICO-CORE catalogue to make it discoverable. D2PTS are pilot-focused thematic services to demonstrate the benefits of the JERICO-RI information life cycle. The resulting D2PTS were developed under Task 7.5.2 and they are explained in deliverable D7.5 [4]. Additionally, an IRS Pilot Study in Iberian Atlantic Margin was planned to demonstrate the capability of integration of JERICO-CORE into Blue-Cloud.
This pilot study consists of three services:

- Transboundary processes: Forcing slope currents and interactions shelf-slope circulation
- Extreme events: Impact of extreme events such as storms and hurricanes
- Long term variability and climate change: Ocean warming and heat waves

We highlight the importance of the API in the integration of the resource catalogue into other layers of the e-infrastructure such as EPOS, the Service Layer and the User Layer.

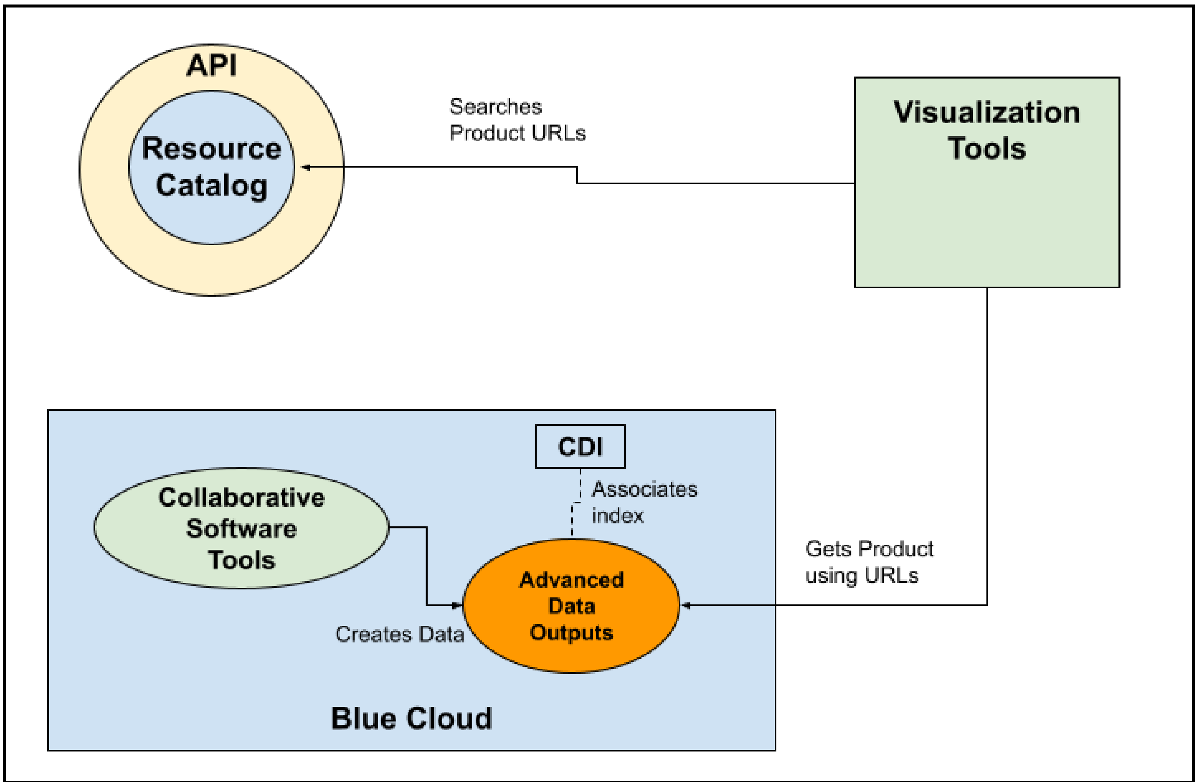

Figure 14: Integration of visualisation tools of advanced products with Blue-Cloud VRE and JERICO-CORE

The following section explains the timeline for implementation of the various parts of the architecture.

3.6. Planning

Figure 15 shows the timeline of the JERICO-CORE implementation as it happened during the duration of task 7.5. We differentiate in the planning various aspects of the architecture according to the distribution of the tasks among the members of the team as follow:

- The **thematic services** were developed by different groups other than the JERICO-CORE core team.
- Different partners were responsible for the front and back end layers. We created a group for the tasks related to the **backend** architecture, catalogue and API. Another group in the planning includes the tasks related to the **frontend** layer.
- **Interfaces and interoperability** were highly dependent on the interfaces of various components.
- We included in the diagram the context of the **operating environments** where JERICO-CORE was installed. Section Development and deployment procedures explains how these environments are managed. The operational code will be running in DATARMOR for the duration of the operational phase under WP11 of JERICO-S3.

The timeline diagram is colour coded depending on the nature of the tasks. We differentiated between the work for defining concepts and collecting requirements (dark brown) from the technical work. Technical work is also divided into design (light brown), implementation (blue) and integration (green).

The schedule was adapted to the circumstances along the two years (2020- 2022) of the JERICO-CORE implementation. It was a challenging situation that required applying agile methodologies to allow flexibility and adapt to the tight and changing circumstances. In particular, the CoVid situation altered the planning strongly. Co-designing and co-implementing this solution required a large number of interactions between multiple partners and external stakeholders. In particular, the baseline solution that we selected was tightly influenced by the signature of the MoU between EPOS and JERICO. The MoU was signed on April 8th 2021 and the code was received shortly after. EPOS provided support to install and understand their software. We received technical documentation on May 7th and 14th 2021. They offered training about the integration of Thematic Services on May 18th 2021. It took the team 7 months of intensive work to integrate EPOS and JERICO-CORE (see section 3.7.6 - EPOS developments and integration). These included important changes to EPOS as well as to the implementation of the catalogue and the API that were already done for JERICO-CORE. The details of the integration work are explained in section EPOS developments and integration.

The MoU between Blue-Cloud and JERICO was signed on June 28th 2022. However, the Blue-Cloud team provided JERICO with access to a Virtual Laboratory on December 15th 2021. This gave the JERICO-CORE team an opportunity to explore the possibilities that the D4Science platform could offer, in particular, in relation to the creation of thematic services.

Unfortunately, at this stage the D2PTS were already implemented and it was not possible to integrate those to the VRE. They were integrated in the JERICO-CORE catalogue. However, we were able to design the integration of the JERICO catalogue and services with the specific example of the IRS Iberian Atlantic Margin pilot study.

Most of the events that altered the planning were not easy to predict or estimate. This strongly affected the implementation and we had to adapt the scope of the architecture along the path. The details of the actual implementation are explained in the following section in which we compare the final product to the initial architecture.

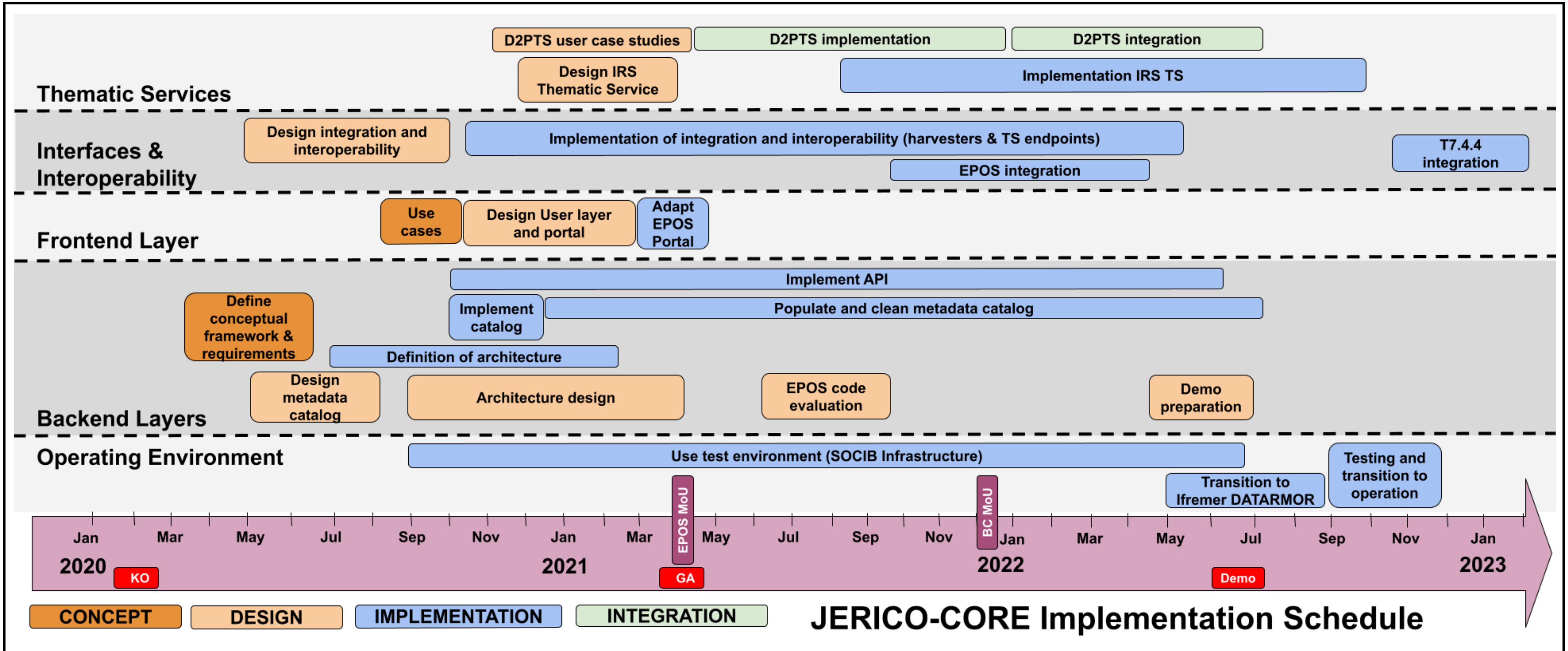

Figure 15: Planning of JERICO-CORE conceptualisation, design, implementation and integration. Tasks are separated in four different groups: thematic services, interfaces/interoperability, front end and back end. The operating environment is also shown in the timeline. Milestones are indicated as a reference. These are the JERICO-S3 kick-off (KO) meeting, the JERICO-S3 General assembly and the JERICO-CORE demonstration during the JERICO-Days in Lisbon. The times of signature of EPOS and Blue-Cloud MoU are also indicated.

3.7. Implementation

The design of JERICO-CORE is explained in the section Architecture Design. This architecture is the solution to the requirements and conceptual design that was co-designed with stakeholders as explained in sections Requirements and JERICO-CORE Concept. In the previous section, we explained the planning and the issues that we had to face during the implementation period. These obstacles influenced the final solution that was implemented for the JERICO-CORE prototype. In this section we describe the technical details of the final implementation of JERICO-CORE. We also explain the differences with the initial architecture.

3.7.1. JERICO-CORE Software Development Kit (SDK)

We implemented the SDK to support the harvester ecosystem, the knowledge ingestion and the API. The code is available in the SOCIB git repository at https://gitlab.priv.socib.es/data-center/ejerico_harvester. It contains the SDK libraries and a template for creating new harvesters. The SDK libraries are available for installation at The Python Package Index (PyPI), a software repository for Python programming language. Libraries can be installed with the Python PIP package manager as **pip install ejerico-harvester**.

The SDK contains the libraries to support:

- Interface to manage the RDF data storage
- Orchestration of harvesters works when running simultaneously
- Management of harvester configurations
- Management of workspaces shared among workers
- Expansion of rdflib library functionalities while taking advantage of original features (multi-store, use of new vocabularies,...)
- Internal monitoring of harvester execution (local sqlite db)

The most relevant method that were implemented are:

- Graph storage management:
    - save(entity)
    - delete(entity)
    - getEntityByURI(uri)
    - findByURI(uri)
    - findURIByName(name)
- Orchestrator (HarvesterManater): abstraction layer to create harvester objects, inject dependencies and execute asynchronous processes
- Configuration methods at various levels of harvester.ini file:
  execution path > code path > user home > /etc
    - get(key, default)
- Harvester status monitoring:
    - isNewOrUpdated(uri, date_updated, hash)
    - markAsVisited (uri, date_visited)
    - maskAsUpdated (uri, date_updated, hash)

3.7.2. Harvester SDK and RDF queue layers

We replaced the P2P strategy with a client-server model due to the limited resources to implement the routing and node discovery layer of the P2P network. This centralised solution was good for the JERICO-CORE prototype because the harvesters were well identified (see section 3.4.2.1). The P2P architecture needs to be re-evaluated in the context of JERICO-DS. Therefore, we replaced the P2P network layer and the Scheduler module of the harvester SDK (see figure 7) with a harvester executor that orchestrates the different harvester processes (see figure 16). The harvester executor may run simultaneous asynchronous workers to accelerate the harvesting process.

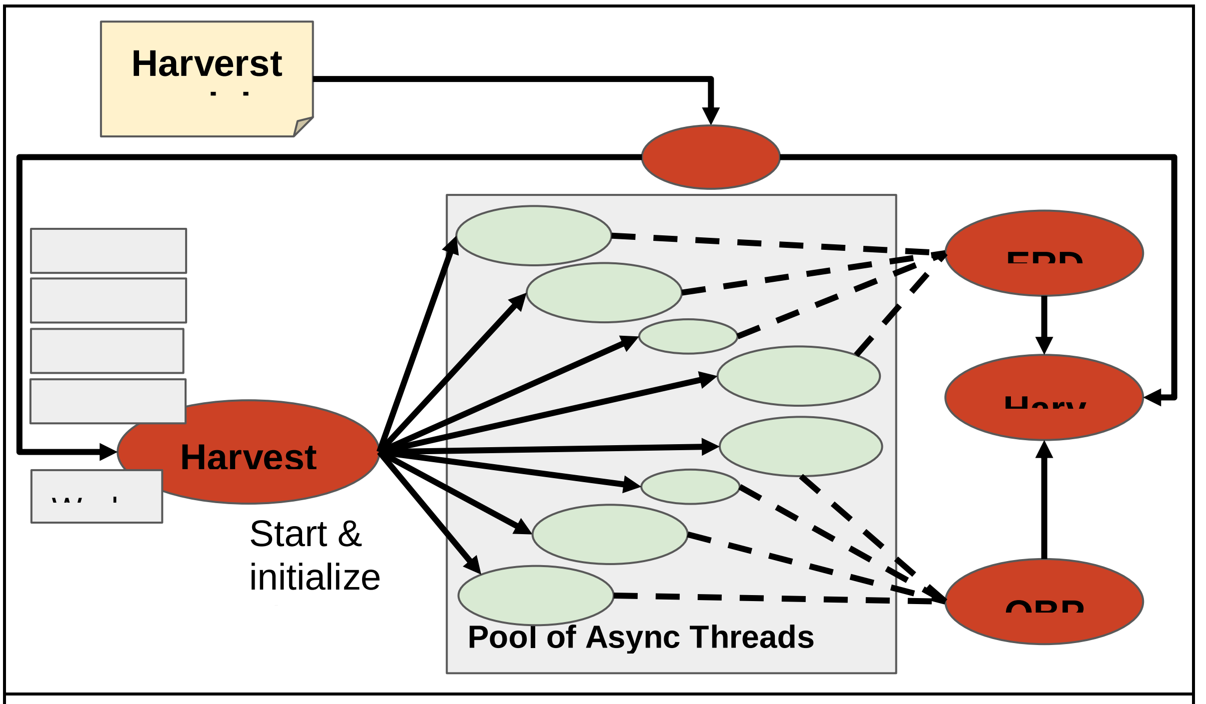

Figure 16: Illustration of the harvesting processes and the link between the orchestration of harvester workers and the SDK.

The harvester SDK uses the libraries of the JERICO-CORE SDK to connect to the catalogue store, to manage the configurations, to monitor the process and to handle errors (see figure 17). The SDK includes the configuration of the AMS and the entity mapping. It includes software components to manage configuration (configManager), the creation of instances to interact with the information storage (GraphFactory) and the management of log files (LoggingManager). Each of these components will create instances at various levels (Harvester Executor, Harvesters and Workers) to carry out specific jobs. Additionally, the StartManager and EntityMapper support the work of the harvesters in collecting and processing the metadata from the providers.

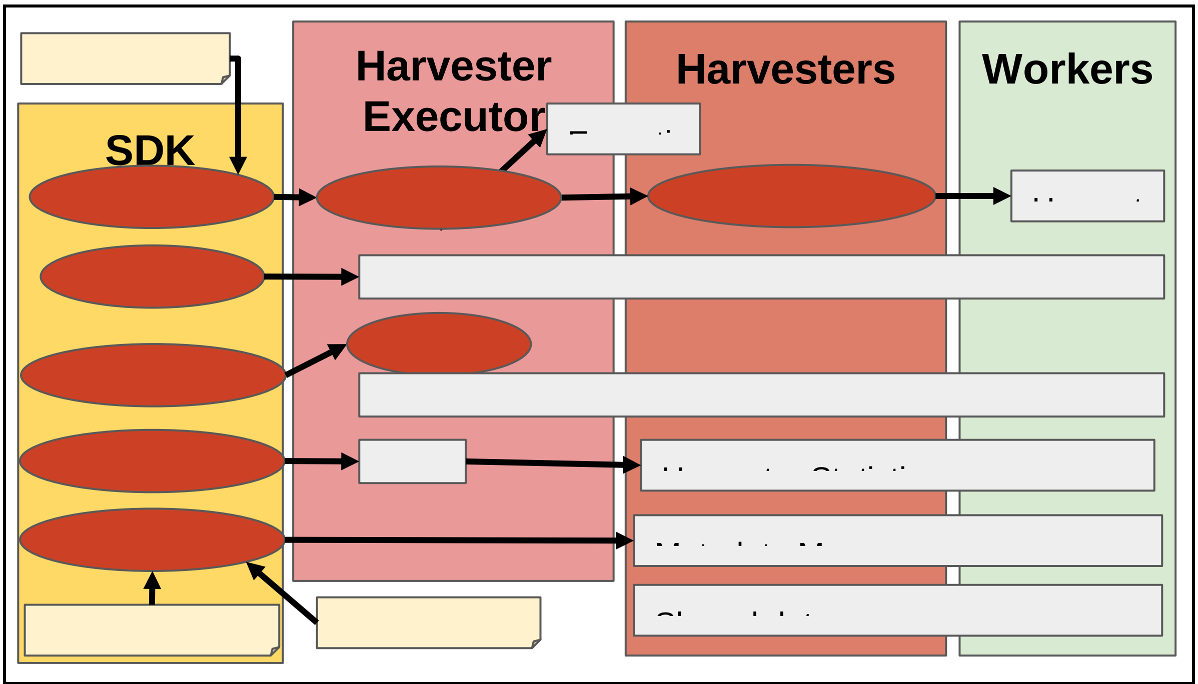

Figure 17: role of the components of the SDK in the orchestration and harvesting processes.

Figure 18 illustrates the structure of the harvester SDK. The RDF Queue layer and the RDF Queue ingestor layer were implemented at the harvester SDK level for practical reasons. Once the P2P approach was set aside, it was not critical to have a centralised mechanism to synchronise the information between the various nodes and the data storage cluster. The AMS was also incorporated in the harvester SDK to facilitate the development since the interface for the AMS was not required with this approach. The harvester SDK includes a template that providers or the JERICO-CORE team have to fill with the custom logic of the harvester. The logic can be set at three consecutive levels before, during and after the harvesting process. This strategy provides the opportunity to set the ground for the harvesting process if necessary and perform checks afterwards. The harvesting process is followed by the mapping and ingestion into the resource catalogue via the EntityMapper and SchemaChecker classes and the rdflib libraries. These libraries were customised to the needs of JERICO-CORE architecture.

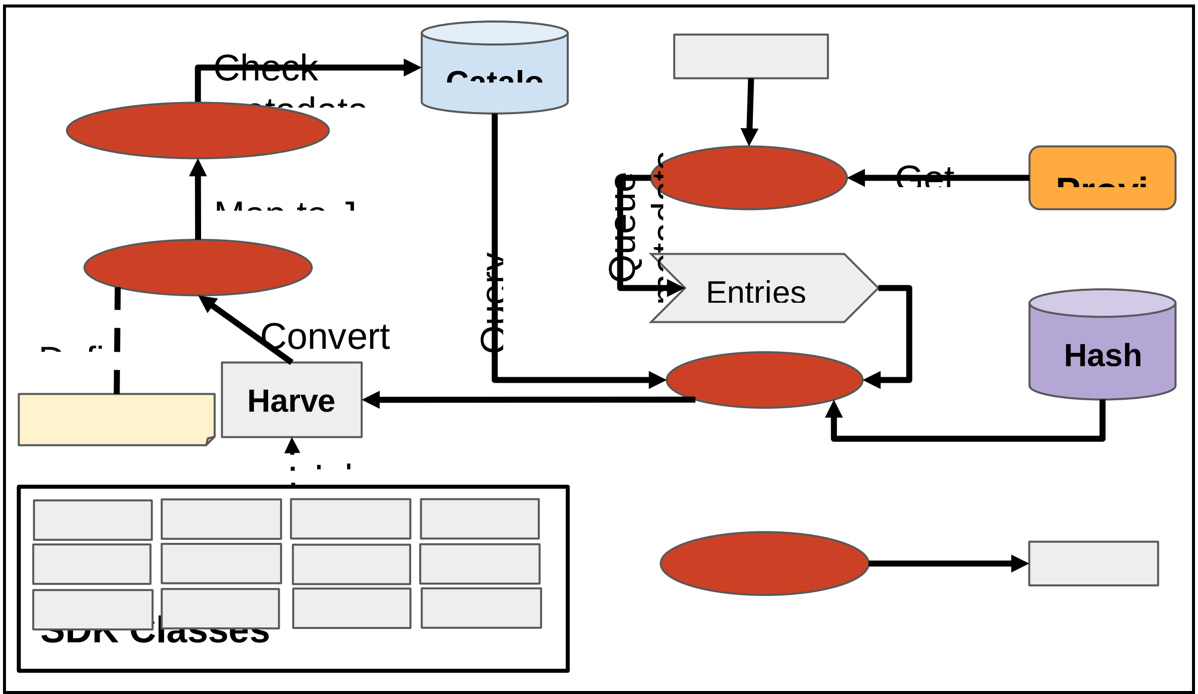

Figure 18: Work flow of the harvesting process and components involved in the process.

The harvester SDK and the general SDK were critical pieces of the architecture. In particular, we use them to create the harvesters that we explain in the following section.

3.7.3. Harvester ecosystem

As explained in the [Harvesting](#) analysis section, there are various cases of providers. We implemented the following cases:

- **ERDDAP harvester**: collects information from any ERDDAP server. Although it needs to be customised depending on the way the datasets are configured in the ERDDAP server. We used this harvester for the EMODnet physics provider.

- **WMS/WFS**: collects information from web services. It needs to be customized for the specific web services that are available. We used the WFS harvester to collect information from EMODnet Biology.

- **DSPACE harvester**: collects information from any DSPACE server. It is usually very generic and can be easily adapted to any DSPACE server because these servers can manage a large range of metadata using the Dublin Core schema. Adapting these harvesters requires a special focus on the metadata mapping. We used this harvester to collect information from OBPS.

- **Github harvester**: collects metadata from github repositories. This harvester was specially designed for the collection of software from https://github.com/. It harvests information of the repositories that are labelled as "jerico-ri" using the github api (see https://docs.github.com/en/rest). In addition to collecting metadata, it can complement the information using json files that are included in the software repository. The harvester will look for files named as "jerico-ri-metadata.json" (more generally as "jerico-ri-metadata-*[label]*.json) and extract the additional metadata that is described in this file when the standard Github fields are not enough. In particular, this is necessary when we need to link software to other assets that Github can't represent such as best practices.

- **Thredds harvester**: collects information from any Thredds server. Similarly to the ERDAPP harvester, it needs to be customised depending on the way the datasets are configured in the Thredds server. This harvester can be used for the EMODnet Thredds server providing HFR datasets of the EU HFR node or Thredds from JERICO partners such as Balearic Islands Coastal Observing and Forecasting System (SOCIB) or Puertos del Estado.

- **SPARQL harvester**: collects information from triple stores serving SPARQL standards. We used this harvester to collect organisations from EDMO and projects from EDMERP.

- **Excel harvester**: collects information from Excel spreadsheets. The definition of this harvester is very bound to the content of the spreadsheet. We used this harvester to collect the platforms from an Excel file that EMODnet physics delivered with the definition of their platforms. Platforms are also extracted from an Excel document obtained from the Coriolis website.

- **JSON file harvester**: collects metadata described in a json file. We created these json files to allow describing knowledge that is not available in the existing providers. For instance, this harvester was used to insert information about services and D2PTS. Json files can be located in local filesystems or a remote server such as github or ftp servers. We created a github repository to host the files describing this extra metadata (https://github.com/socib/jerico-core-metadata). Currently, we use these files to define the JERICO Virtual Access and D2PTS assets.

The technical limitations of these harvesters are the following:

- Management of resource availability: we should include automatic mechanisms to detect when resources are available in the catalogue. Currently, there is a strong human implication in the process to debug errors when information is missed.

- Incremental load features: the loading processes are very demanding to be managed by a single process. We need mechanisms to allow collection in multiple and lighter processes.

- Management of concurrent processes to allow information collection in near real time.

- Observability and metrics need to be implemented. We performed some testing using ELK but the implementation was not pursued because of the lack of resources.

In general, the biggest challenge is the Extraction step of the ELT that requires consistent metadata at the providers. We need to implement mechanisms that handle the inconsistencies that were found among the information sources. We also need to agree among the JERICO community the metadata standards to represent the resources in the JERICO knowledge catalogue.

3.7.4. Data Storage

As explained in the section of Storage Analysis No4j is the most optimal solution to manage knowledge graphs. However, Apache Jena Fuseki integrates SPARQL functionalities which are more appropriate to integrate with SeaDataNet services (EDMO, EDMERP, EDIOS,...) and with EPOS which mainly use TTL files to register information. Apache Jena Fuseki is a SPARQL server that can run as a docker server which helps the development and deployment processes.

Additionally, we adapted the metadata schema used by EPOS to help the integration to their system (see section EPOS Integration). Their schema is mostly defined for the purpose of recording services and datasets. CERIF is capable of recording other types of information but the EPOS UI had specific constraints to represent assets. We updated the ingestion process to adapt to the requirements of the EPOS UI and allow JERICO-CORE assets to be visualised. Moreover, we tried to assimilate the maximum information from JERICO-CORE providers by adding new attributes to the existing or new entities. These two changes were possible because of the strong flexibility of the CERIF standard.

3.7.5. API layer

In order to facilitate CI/CD (see section 3.4.2.5) the API layer was implemented in docker microservices. The API layer runs with three docker containers in order to separate the logic of the API from other functionalities such as the cache and the web server management. Figure 19 shows the three containers and their dependencies with respect to the mounted volumes, configurations and network. They are illustrated as green rounded squares labelled as nginx, api and db.

- **nginx**: the nginx container helps to handle the web access and serve the static files. It is a popular web server that reduces the load of the applications, cache content and offload SSL/TLS processing. Adding this container allows decoupling the API from the hosting infrastructure.
- **db**: The db container we used is a postgres docker that manages the cache. The API docker interacts with the postgres container to discover or synchronise information.
- **api**: The api container contains the proper logic of the API layer. It serves the information that responds to the requests in the appropriate format.

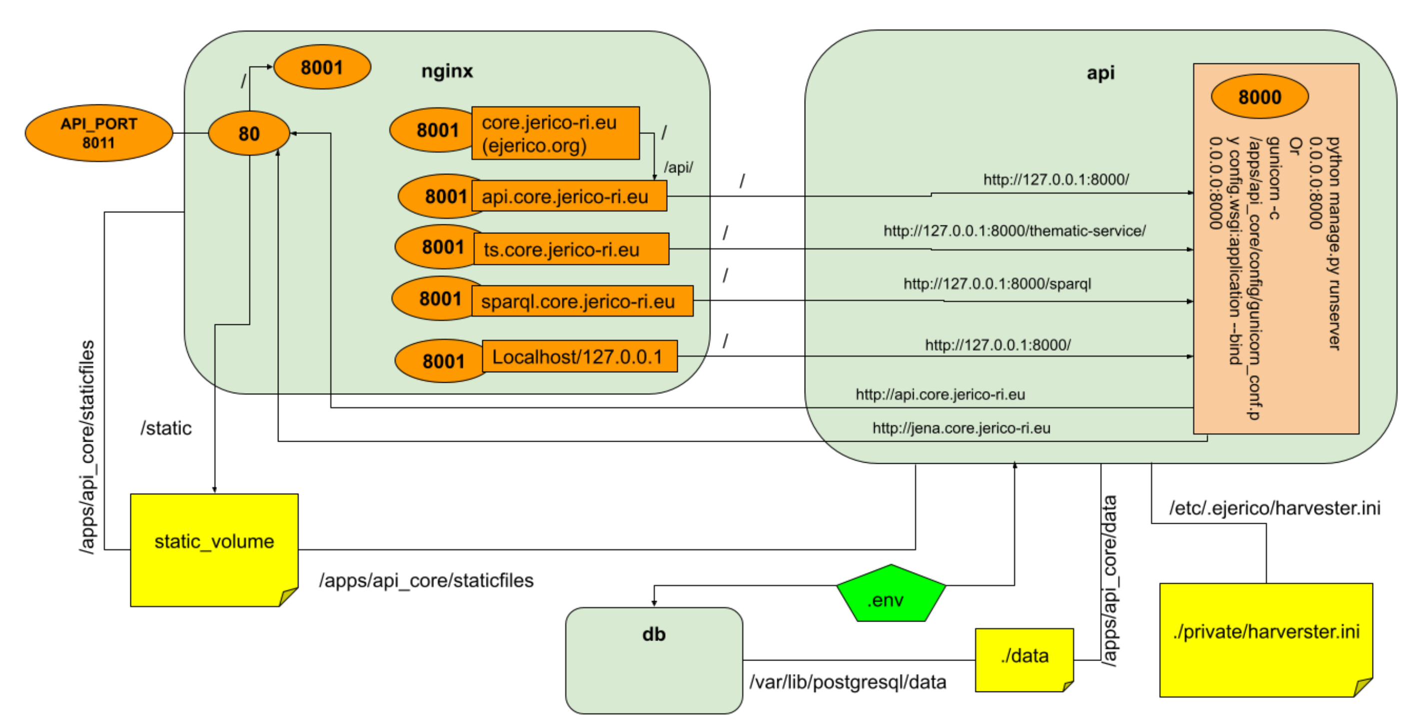

Figure 19: Docker containers of the API layer. The diagram shows their relation as well as the main configurations.

The API layer provides five main endpoints:

- **Restful API**: used to discover resources in the catalogue or to get information about a specific asset. This endpoint is the main point of discovering and accessing the catalogue knowledge. In particular, services developed in the Blue-Cloud context or outside should use this endpoint as the foundation to integrate to JERICO-CORE. This will be the case for service applications developed in the context of the Blue Cloud 2026 project (https://blue-cloud.org/blue-cloud-2026).
- **SPARQL browser**: used to perform raw SPARQL requests to the resource catalogue. This endpoint complements the SPARQL endpoint of the Jena Fuseki server because it is made available to the general public. It performs SPARQL requests through the SDK instead of the Jena functionalities.
- **Statistic**: provides information about the content of the resource catalogue.
- **Synchronisation**: allows the synchronisation between the jena storage and the postgres cache as explained below.
- **Thematic service endpoints**: uses the API and custom logic to respond to requests of the user layer about advance knowledge of the catalogue. It returns the information in a specific format. For example, it serves the EPOS requests.

Restful API

We used the Django REST framework to build the JERICO-CORE Restful API. It is a powerful toolkit to build web APIs. It is usually based on the models of relational databases but it is flexible enough to support non Object Relational Mapping. RDF triplet stores are databases outside the object-oriented paradigm. The REST framework allows customization of the model representation and serialisation to the framework provided by the SDK. We tried two approaches. In the first approach, the API layer interrogated the Data Storage Cluster for each request. However, performance of the RDF triples modelization was not good enough to respond to the needs of the service and user layers. In the second approach, we created a postgres cache that was populated with the information of the Jena storage through an endpoint. This solution performed well with the service and user layers although it requires careful synchronisation with the data storage cluster. This latest approach justifies the creation of the synchronisation endpoint. The documentation of the Restful APi is available at https://api.core.jerico-ri.eu/api/swagger. It explains the basic entities of the catalogue and general usage. A more detailed description of the content of the catalogue and the services providing this information is available in a more elaborate documentation at https://api.core.jerico-ri.eu/api/advanced_swagger.

SPARQL endpoint

The SPARQL endpoint accepts SPARQL requests and returns the output in three different formats: json, xml and csv. It can be used to integrate the JERICO-CORE catalogue into the semantic web. The semantic web is an advanced version of the World Wide Web (WWW) often called Web3.0. It expands it through standards set by the World Wide Web Consortium (W3C). RDF, Web Ontology Language (OWL) and SPARQL allows encoding information for sharing data in the semantic web. In particular, SPARQL is a protocol and query language for semantic web data sources.

Statistic

The statistic endpoint is a Restful API that is used by the development team to have an overview of the resources that are stored in the Jena catalogue, the CERIF database and the API cache. It is planned to be expanded to include statistics about the resources that could be useful to various users. For the moment, it includes the following options and parameters:

- https://api.core.jerico-ri.eu/stats/report: list of available assets and the number of each of them in the API layer cache and the data storage cluster.
    - No arguments returns: resource: #cache/#storage
    - Percentage argument returns: resource: percentage of resource in cache compared with data storage
    - brand=jerico returns: same as before for assets that are labelled as jerico. Numbers of the data storage are not filtered by brand.

- https://api.core.jerico-ri.eu/stats/sparql/: list of available assets in the data storage cluster

- https://api.core.jerico-ri.eu/stats/summary/: list of available assets in the API layer cache

- https://api.core.jerico-ri.eu/stats/jerico/: same as report but results are filtered for assets that belong to jerico. The difference with the brand=jerico parameter is that both cache of the API layer and the information from the data store are filtered.

Synchronisation

The synchronisation endpoint was critical to adopt an approach that optimises the performance serving knowledge. As explained above, we used a cache that contains the information of the data storage. In order to update the cache we can use four endpoints:

- https://api.core.jerico-ri.eu/db/clear/: cleans all the content of the cache
- https://api.core.jerico-ri.eu/db/sql/: allows to upload a sql file to update the information of the cache
- https://api.core.jerico-ri.eu/db/init/: same as before but clears the cache before uploading the sql file
- https://api.core.jerico-ri.eu/db/update/: updates the information of the cache based on the content of the data storage.

These endpoints accept various arguments:
- Token: mandatory to perform any update
- Entities: list of entities to be updated. Otherwise, all entities are updated
- Skip_entites: list of entities to be skipped during the update process. Otherwise, no entity is skipped. This option is useful to accelerate the synchronisation of the cache and the Jena catalogue because entities that are already up to date are not updated.

Thematic Service endpoints

We created two thematic service endpoints to support the EPOS integration as explained in section 3.5 (Architecture Design). The TS integration

- http://api.core.jerico-ri.eu/thematic-service/[resource_type]/[resource_id_hash]
- http://api.core.jerico-ri.eu/thematic-service/search-[resource_type]/

Both of these endpoints return a geojson that follows the description of the EPOS manual (see https://epos-ci.brgm.fr/epos/metadata-database). The first endpoint returns the information of a specific resource. The second one provides a list of resources of a specified type according to the search criteria. These criteria can be input as an argument following the filters provided by the Restful API. The thematic service endpoints were critical to integrate JERICO-CORE to EPOS. We explain in the next section the details of the integration.

3.7.6. EPOS developments and integration

We approach the integration of services to JERICO-CORE following the architecture provided by EPOS (see figure 12). As explained in section Architecture Design, the interoperability layer plays two roles. First, the ingestion of the resource metadata is performed through the ingestor. The Ingestor connects to the Jena Fuseki server and constructs the entities that are recorded in the CERIF metadata catalogue (see section 3.4.1.2) as well as their relationship. Second, the access to the information requested by the user is done through the TCS Connector which interrogates the thematic centre endpoints to collect the information to be represented in the User Interface. Figure 20 illustrates the integration between JERICO-CORE components and the EPOS core services from these two perspectives.

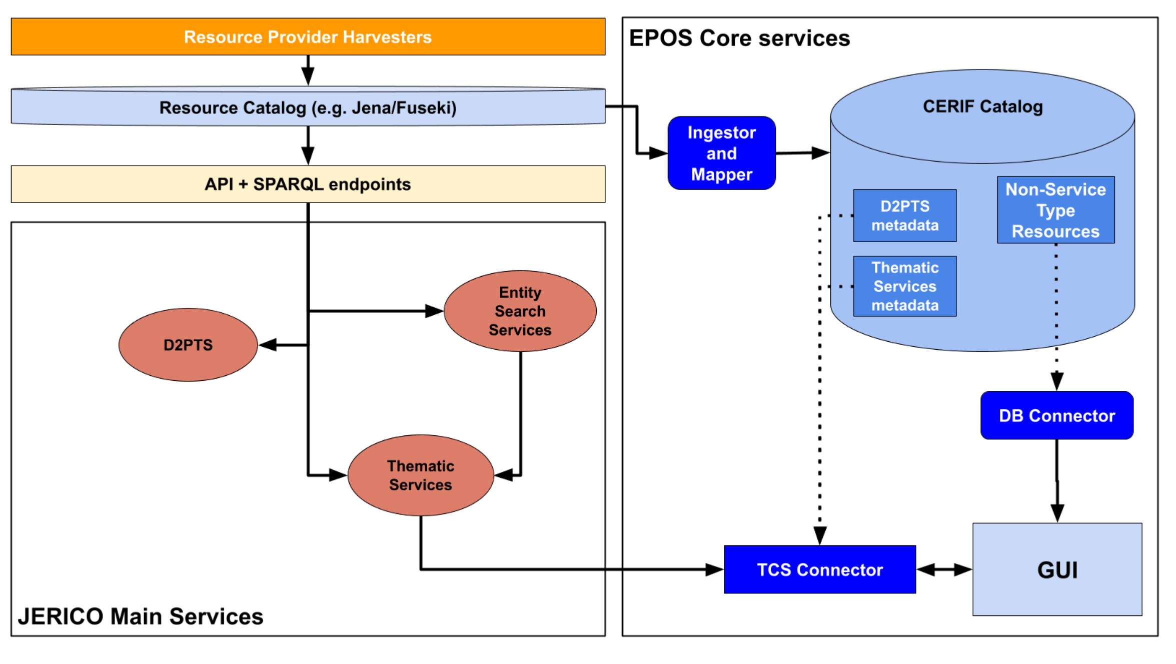

Figure 20: Integration of JERICO-CORE services and the EPOS components. The diagram shows the role of the API layer facilitating the connection of the JERICO Thematic Sercices and the EPOS Core services.

The method used by EPOS to ingest the metadata information in CERIF is to use their WebAPI to upload a TTL file which is sent to the ingestor. However, JERICO-CORE components are written in Java and Python, which condition the use of objects and structures of RDF-JSON type, to the detriment of the RDF-TTL used by EPOS. For these reasons, one of the first and main tasks was to extend the functionality of some EPOS components to add the possibility of ingesting information from a streaming source of RDF-JSON information, and then parse and map the RDF-JSON model to CERIF. Table 5 lists the changes that were carried out in the EPOS code to implement this functionality.

We explored mechanisms to ingest the information of the JERICO triple storage into the EPOS CERIF catalogue (see figure 21). We studied this mechanism (option 1 of figure 21) through a transitional database that hosts the TTL documents. We used ElasticSearch documents as an intermediate step where information is saved using json-ld format. This implementation required the changes of EPOS shown in Table 6.

After many tests with Elasticsearch as a RDF server, we realised that parsing became a little bit slower than RDF-JSON streaming. In order to easily speed up the parsing process, we explored retrieving the information from the original RDF Graph store, the Apache Jena Fuseki server (option 2 of figure 21). The second option was more efficient and reliable because it did not require additional mechanisms to guarantee the correct synchronisation of the Elasticsearch storage. These developments also required a considerable amount of changes of the EPOS source code (see table 7).

The integration between JERICO-CORE and EPOS needed a large amount of work related to the definition of the ontologies defining the metadata schema (see appendix E) that are registered in CERIF. For this purpose we implemented an additional OWL to represent resources of JERICO-CORE. This implementation required complementing the CERIF EPOS tables. We adapted the EPOS code to manage both ontologies simultaneously by adding new components that were complementary to the original ones:

- jsldreader.EPOSRDFReader.java => org.ejerico.jsldreader.EJericoRDFReader.java
- jsldreader.Reader.java => org.ejerico.jsldreader.Reader.java
- org.epos_ip.rdf.vocabulary.EPOS.java => org.ejerico.rdf.vocabulary.EJERICO.java
- Created new component: org.ejerico.rdf.vocabulary.SOSA.java

Additionally, we added two JERICO OWL resources and properties to the EPOS OWL in CERIF which are included in files SOCIB_CERIF_SETUP.sql and SOCIB_CERIF_TRUNCATE_AND_REFILL.sql. Moreover, we had to evolve aspects of the EPOS GUI to adapt the faceted search, advanced search and the different views (graphs, tables and maps) to the target topics of the JERICO project according to the types of resources that are available. Additional entities will need to be added and classified in future developments as the JERICO OWL evolves. It will require extending the Oceanography sub-classifications of JERICO-CORE CERIF schema. Additionally, it will have to catch additional properties and resources collected in the JERICO-CORE data storage.

Project	Component	Section/Method	Description	Purpose
WebApi	org.epos.api.controllers. InternalApi.java	*Interface definition / Swagger options*	*Adds new RDFJSON option for the API "internal/ingestor" endpoint.*	*RDF-JSON ingestion from an external URL.*
WebApi	org.epos.api.controllers. InternalApiController.java	*Internal/ingestor endpoint*	*Process the new RDF-JSON option passing the source URL to jsldReader.*	*Creates a new and differentiated call to jsldReader in order to distinguish the RDF-JSON ingestion.*
jsldReader	org.epos.system. IngestorHandler.java	*handle(...)*	*Interprets the RDF-JSON call from WebApi component and passes the call to the proper method of EJericoReader class.*	*Decides which the reader object is for the RDF-JSON URL.*
jsldReader	org.ejerico.jsldreader. EJericoReader.java	*IngestRDFJSON(...)*	*New method for the RDF-JSON streaming ingestion.*	*Obtains the RDF-JSON information from the source URL opening a streaming connection.*
jsldReader	pom.xml	*<dependencies>*	*org.apache.jena library reference.*	*Adds to the ingestor the possibility to use the Apache Jena libraries to manage RDF models.*

Table 5: List of components changed in EPOS code to implement RDF-JSON type collection.

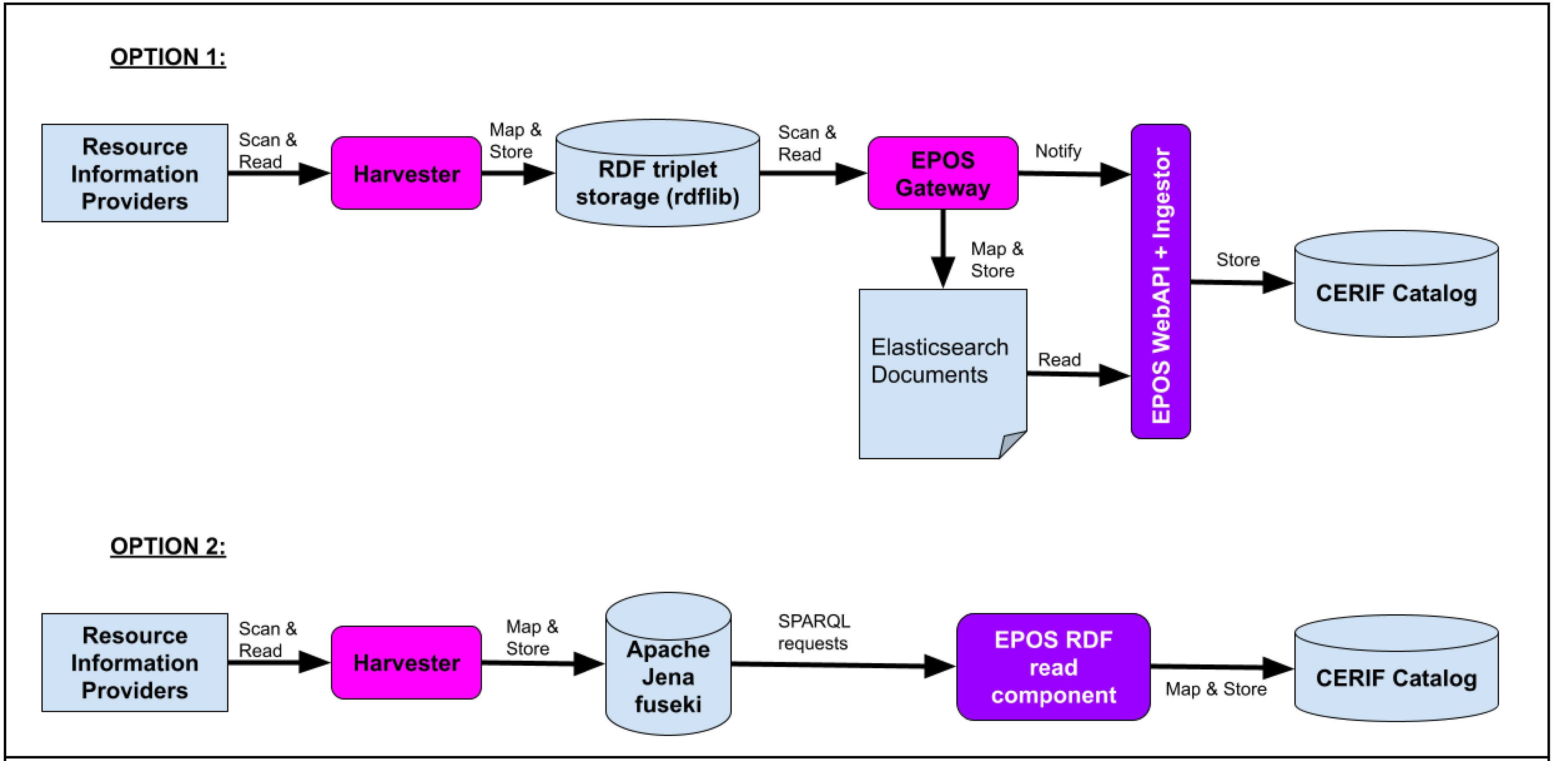

Figure 21: Two options of mechanisms for synchronisation of the CERIF catalogue of EPOS with the resource catalogue of JERICO-CORE.

Project	Component	Section/Method	Description	Purpose
WebApi	org.epos.api.controllers. InternalApi.java	*Interface definition / Swagger options*	*Adds new RDFJSONLD option for the API "internal/ingestor" endpoint.*	*RDF-JSONLD ingestion from an external ELK Server (URL).*
WebApi	org.epos.api.controllers. InternalApiController.java	*Internal/ingestor endpoint*	*Process the new RDF-JSONLD option passing the source URL to jsldReader.*	*Creates a new and differentiated call to jsldReader in order to distinguish the RDF-JSONLD ingestion.*
jsldReader	org.epos.system. IngestorHandler.java	*handle(…)*	*Interprets the RDF-JSONLD call from WebApi component and passes the call to the proper method of EJericoReader class.*	*Decides which the reader object and its method is for the RDF-JSONLD URL.*
jsldReader	org.ejerico.jsldreader. EJericoReader.java	*IngestESJSONLD(…)*	*New method for the RDF-JSONLD streaming ingestion.*	*Obtains the RDF-JSONLD information from a default eJerico-Elasticsearch catalog.*
jsldReader	pom.xml	*<dependencies>*	*org.apache.jena library reference.*	*Adds to the ingestor the possibility to use the Apache Jena libraries to manage RDF models, and connect to ELK server (an RDF-JSONLD information server).*
jsldReader	org.ejerico.jsldreader. cerifpersistence.*	*All DAO classes*	*Migration of persistence layer from JPA 2.0 to Jakarta JPA 3.0*	*Improves database I/O access performance.*
CERIF Database	DML Scripts	*Base definition information*	*Oceanography OWL added*	*Adds Oceanogaphy ontology concepts*

Table 6: List of components changed in EPOS code to integrate JERICO-CORE storage and CERIF.

Project	Component	Section/Method	Description	Purpose
WebApi	org.epos.api.controllers. InternalApi.java	*Interface definition / Swagger options*	*Adds new "eJerico" option for the API "internal/ingestor" endpoint.*	*RDF-JSON ingestion from the Apache Fuseki Server (URL + dataset).*
WebApi	org.epos.api.controllers. InternalApiController.java	*Internal/ingestor endpoint*	*Process the new RDF-JSON option passing the source URL + dataset name to jsldReader.*	*Creates a new and differentiated call to jsldReader in order to distinguish the Fuseki ingestion.*
jsldReader	org.epos.system. IngestorHandler.java	*handle(…)*	*Interprets the call from WebApi component and passes it to the proper method of EJericoReader class.*	*Decides which the reader object and its method is for the Fuseki server.*
jsldReader	org.ejerico.jsldreader. EJericoReader.java	*IngesteJericoRDF(…)*	*New method for the Fuseki streaming ingestion.*	*Obtains the RDF-JSON information from the eJerico-Fuseki server.*
jsldReader	pom.xml	*<dependencies>*	*org.apache.jena library reference.*	*Adds to the ingestor the possibility to use the Apache Jena libraries to manage RDF models, and connect to Apache Fuseki Server.*

Table 7: List of components changed in EPOS code to integrate collect the graph of the Jena catalogue without intermediate steps.

The resulting code was easily integrated into the JERICO-CORE architecture because the EPOS software is also packed in docker containers. Figure 22 illustrates the way the docker containers interact and integrate with the Data storage, the API and the USER layers. A series of documents explaining these components were shared by the EPOS team with us. These documents have restricted access and can be read at the following links:

- Workspace: https://epos-ci.brgm.fr/epos/Workspace (Master, 0.7.0)
- Ingestor: https://epos-ci.brgm.fr/epos/Ingestor (Master, 0.6.0)
- DBConnector: https://epos-ci.brgm.fr/epos/DBConnector (Master, 0.7.0, 0.7.1)
- QueryGen: https://epos-ci.brgm.fr/epos/QueryGen (Master, 0.6.0)
- Mapper: https://epos-ci.brgm.fr/epos/Mapper (Master, 0.6.0)
- TCSConnector: https://epos-ci.brgm.fr/epos/TCSConnector (Master, 0.6.0)
- WebApi: https://epos-ci.brgm.fr/epos/WebApi (Master, 0.6.0)
- metadata-database: https://epos-ci.brgm.fr/epos/metadata-database (Master)
- RoutingFramework: https://epos-ci.brgm.fr/epos/routingframework (develop)
- WebApi: https://epos-ci.brgm.fr/epos/epos-gui (Master, 0.20.5)

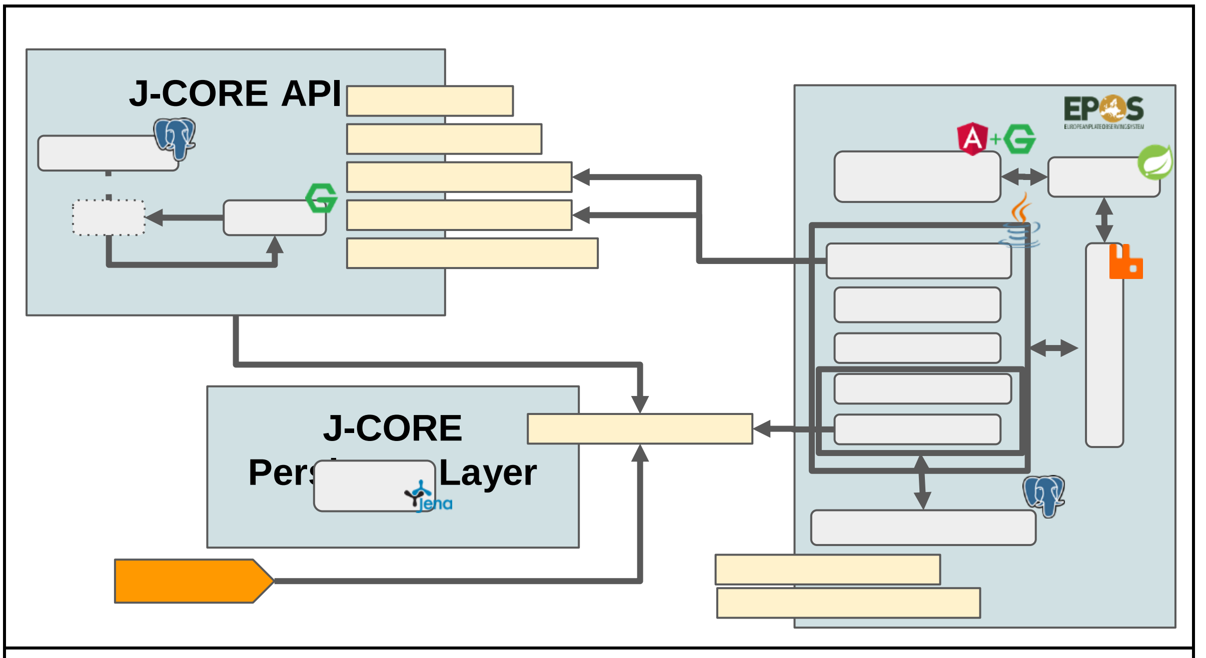

Figure 22: JERICO-CORE components and docker containers of each component

With the integration of EPOS, we implemented the user layer (see section 3.7.8) by providing the EPOS UI to explore the resource catalogue. This integration allows human users to search and discover JERICO resources interactively. Another important aspect to be integrated in JERICO-CORE is the Service layer which will allow partners to develop tools to respond to their specific needs. We explain in the next section how this layer is integrated with JERICO-CORE.

3.7.7. Service layer

The service layer must support services in two ways. On one hand it should integrate with the resource catalogue to provide advanced knowledge on demand. It must also provide the appropriate environment to develop tools in a collaborative way that use this knowledge and have access to datasets among the federated providers.

For these two aspects, the API and the searching capabilities of JERICO-CORE play a critical role. This functionality supports thematic services that are federated among partners such as the D2PTS as well as services that are co-developed in the Blue-Cloud VRE as an important part of JERICO-CORE (see figure 23). Among other examples of services in the Blue-Cloud VRE we can mention the IRS pilot study, the PSS west mediterranean water mass transport and the HFR quantitative framework.

The IRS pilot study was explained in section Architecture Design. In the current status, the service was demonstrated and will be integrated into the VRE in the context of the Blue-Cloud 2026 project. Similarly, the PSS West Mediterranean Water Mass Transport service will be developed in the context of this project. Some integration studies were carried out in the Jupyter notebook of the JERICO-RI Virtual Lab (JERICO-CORE_t1.ipynb). This study demonstrated the use of the API to collect appropriate datasets from federated providers. Finally, the HFR quantitative framework (HFR_Quantitative_Framework.ipynb) was developed to support the activities of the EU HFR node. It was presented in the EuroGOOS HFR Task Team workshop, held in Florence (November 21st-22th 2022). This example helped us to identify gaps in the metadata from HFR dataset providers.

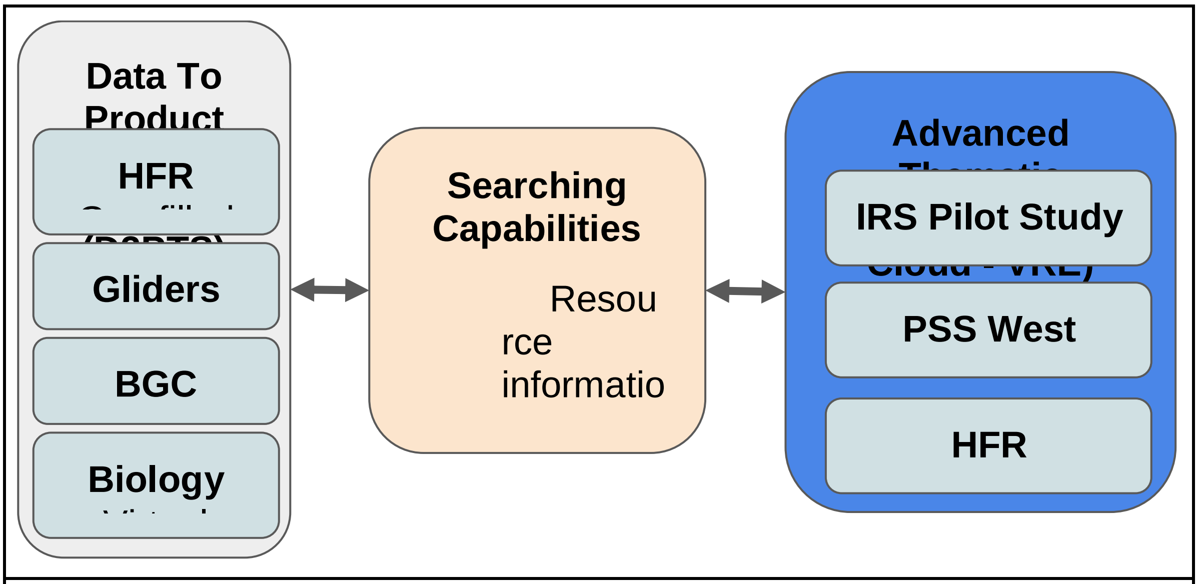

Figure 23: Support of searching capabilities to services internal and external to the VRE.

Figure 24 summarises the integration from a technical perspective of services to the Blue-Cloud VRE, the Catalogue, the JERICO-CORE API and EPOS. In this integration, the JERICO-CORE API plays a critical role by decoupling the three components from one another.

The JERICO-CORE VRE (https://blue-cloud.d4science.org/group/jerico_core/home) can be accessed with restricted access. An account is created after registration at https://blue-cloud.d4science.org/. Then, access to the JERICO-RI Virtual Lab needs to be requested to the VLab administrator. Figure 26 shows the various areas that users can access. The main features are explained in the tables 8 and 9. Details of each of these features can be found in the Blue-Cloud Demonstrator Users Handbook V2 [12].

Icon	Description
	Blue-Cloud Gateway Dashboards
	Workspace
	Catalogue
	Messages
	Search

Table 8: Icons of the top bar icons of the JERICO-RI Blue-Cloud VLab.

Labels	Description
JERICO CORE Vlab	Link to the homepage of the JERICO-CORE VLab.
Administration	Management of users, groups and dashboards.
Analytics	Access to metrics of usage of the VLab
RStudio	Collaborative space for developing R tools.
JupyterHub	Collaborative space for developing python tools. Multiple server options are available.
Social Feed	Access to social media style chats where collaborators can share their ideas and communicate with partners.
Members	Show list of members of the Vlab and their roles.
How-to	Show Blue-Cloud documentation

Table 9: Available options of the main bar of the JERICO-RI Blue-Cloud VLab.

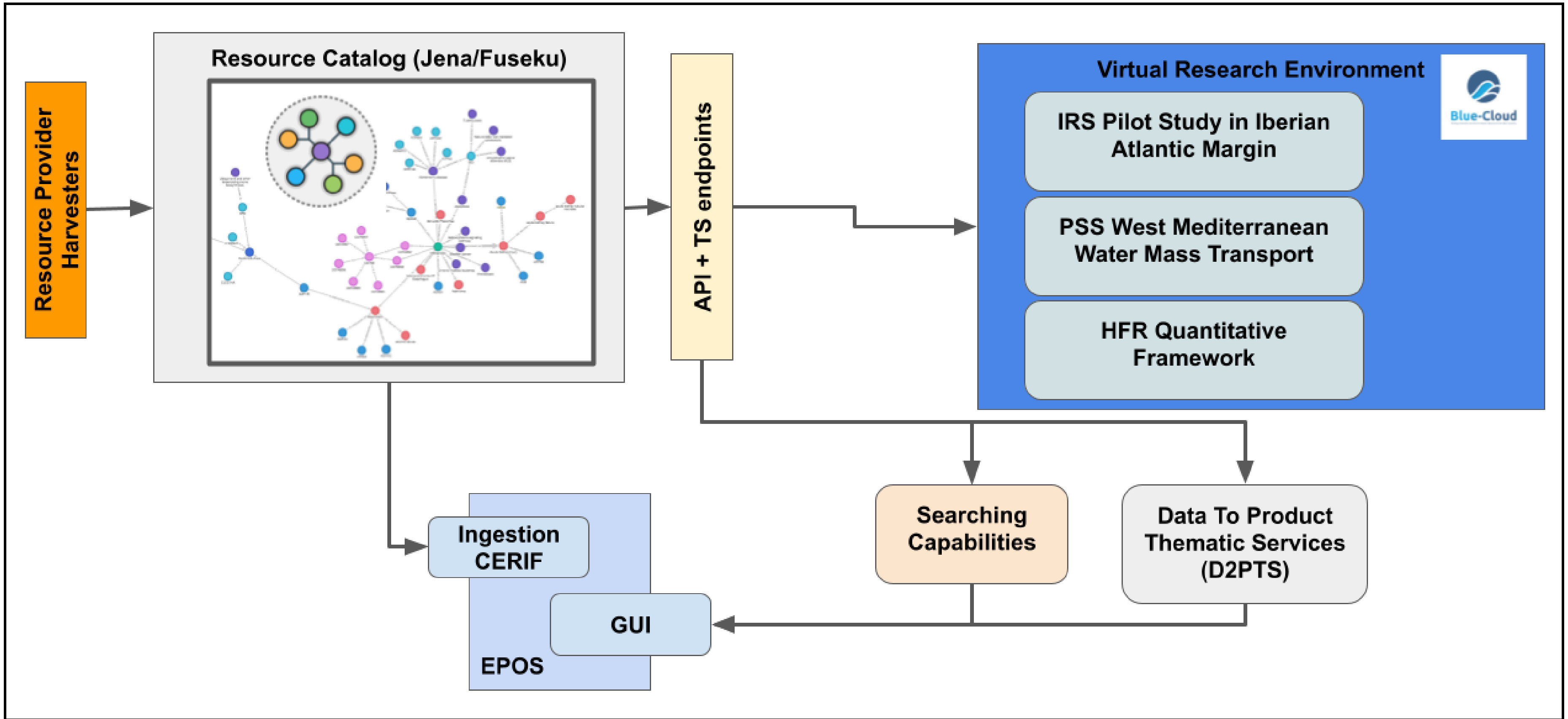

Figure 24: Information workflow for stand-alone and VRE integrated services.

We describe in particular three areas of the Blue Cloud UI for their relevance in relation to the co-development of JERICO services. The JupyterHub (see figure 27) and RStudio (see figure 28) areas are spaces where partners can create their own R or python tools. The Workspace area allows users to upload and browse files that are used by services. Datasets can be loaded using the Data Discovery & Access Service (DDAS). DDAS (https://data.blue-cloud.org/search) facilitates discovery and retrieval of marine data sets and data products from blue data infrastructures (BDIs) that are connected to Blue-Cloud such as Copernicus Marine Service, EMODnet, SeaDataNet (see figure 29). It servers federated discovery and access and allows importing results into Blue-Cloud workspace for further analysis or processing.

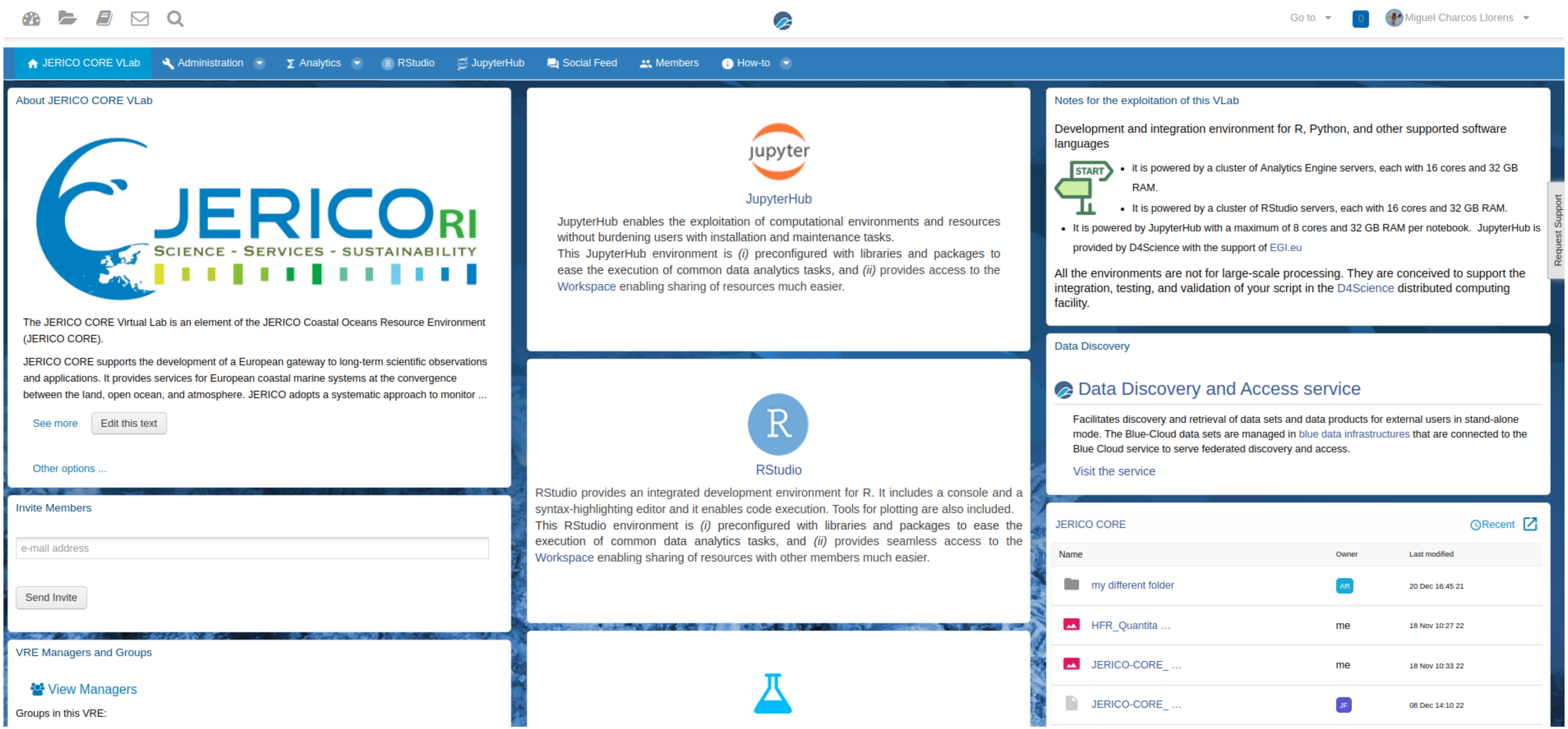

Figure 26: JERICO-RI Blue-Cloud Virtual Lab - Main page

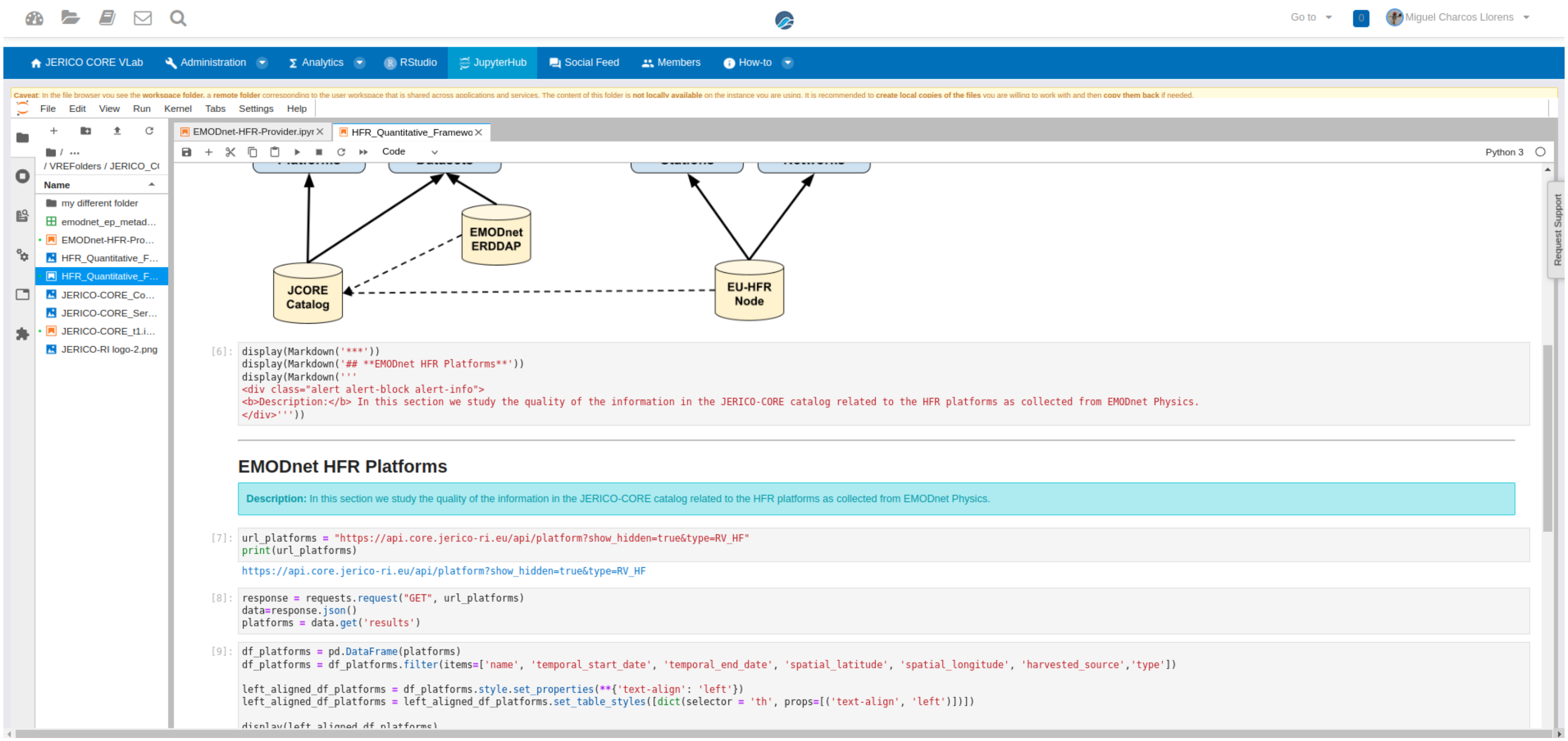

Figure 27: JERICO-RI Blue-Cloud Virtual Lab - JupyterHub

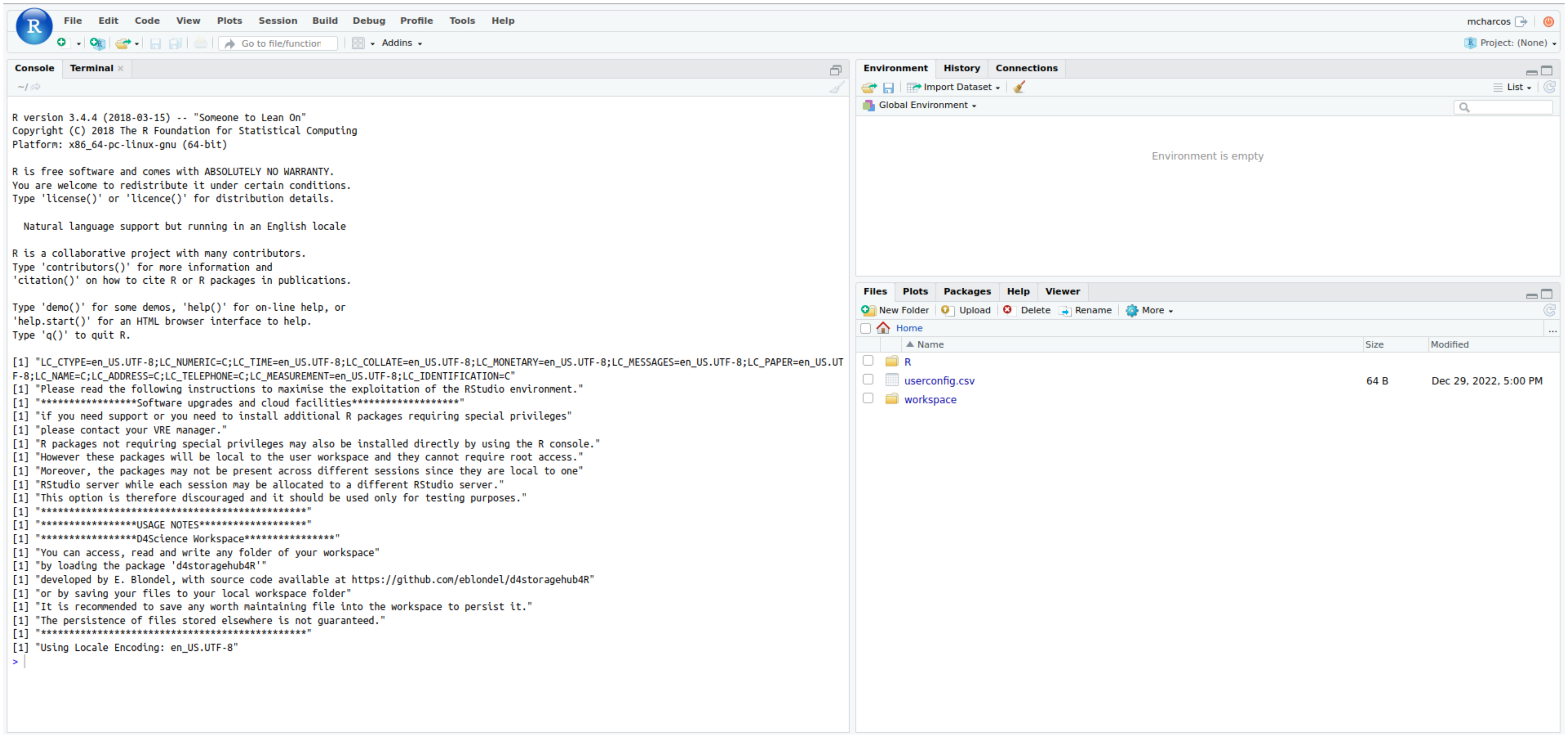

Figure 28: JERICO-RI Blue-Cloud Virtual Lab - RStudio

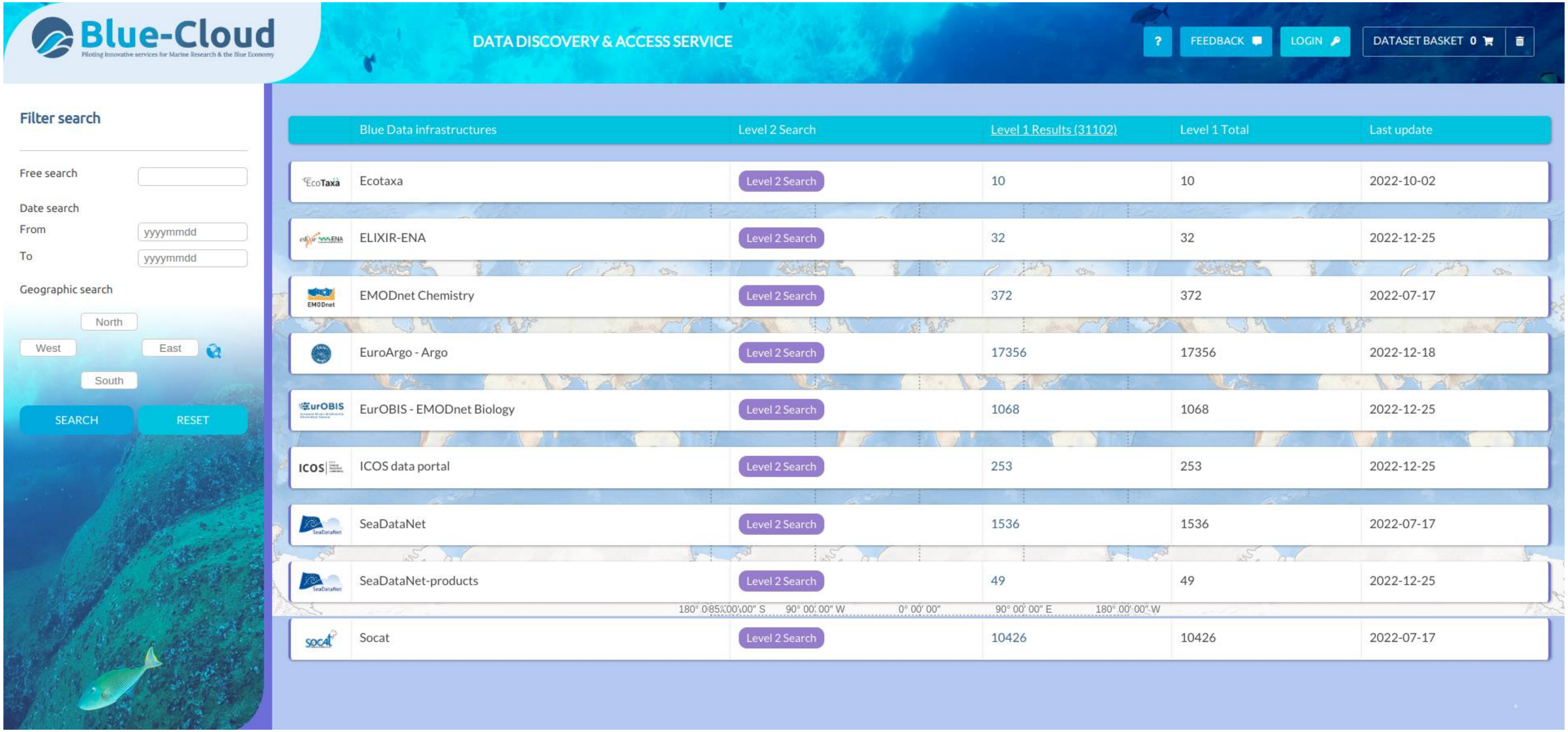

Figure 29: JERICO-RI Blue-Cloud Virtual Lab - Data Discovery & Access Service (DDAS)

3.7.8. User layer

A google form [5] was set up to collect user stories that help define use cases that respond to needs of the JERICO community. These use cases would define among other things the user interfaces that are needed. The user layer will include a variety of user interfaces for specific purposes. In the context of JERICO-S3 we scoped the development to two solutions: The HFR Online Outage Reporting Tool (HOORT) and the EPOS UI.

HOORT:

The HFR Online Outage Reporting Tool (HOORT) is a new tool to support operators tracking system outages (https://hoort.hfrnode.eu/). It was presented in the EuroGOOS HFR Task Team workshop, held in Florence (November 21st-22th 2022). HOORT was developed under JERICO-S3 T7.5 and EuroSea T3.6. It currently extracts the information through the HFR EU node database. An API was developed by SOCIB for this purpose that will be used to harvest the information for the JERICO-CORE catalogue. This API serves information recorded in the HFR EU node database. This additional step will enhance the possibilities of representing the incidences by linking outage to the data life cycle.

EPOS UI:

The EPOS UI is used to browse existing resources in the JERICO-CORE catalogue through a user interface. The main UI under the JERICO-S3 task is the EPOS UI that was adapted for JERICO-CORE (https://ui.core.jerico-ri.eu/), namely the EPOS JERICO-CORE User Interface (EJCUI). The EPOS UI provides a single point of access to data and services for specific Earth Science communities. As explained in section EPOS developments and integration, we have expanded EPOS capabilities to the JERICO-CORE goals of representing information including a larger variety of resources such as software, documents, equipment and more. Similarly, we have adapted the EPOS interface to allow discovering and accessing these resources. Assets can be searched by type or more advanced filtering that takes into account the relations represented in the JERICO-CORE knowledge catalogue. Additionally, a view of the details of the JERICO-CORE resources has been added to the API with the same look and feel as EJCUI. This last piece of code was written un javascript.

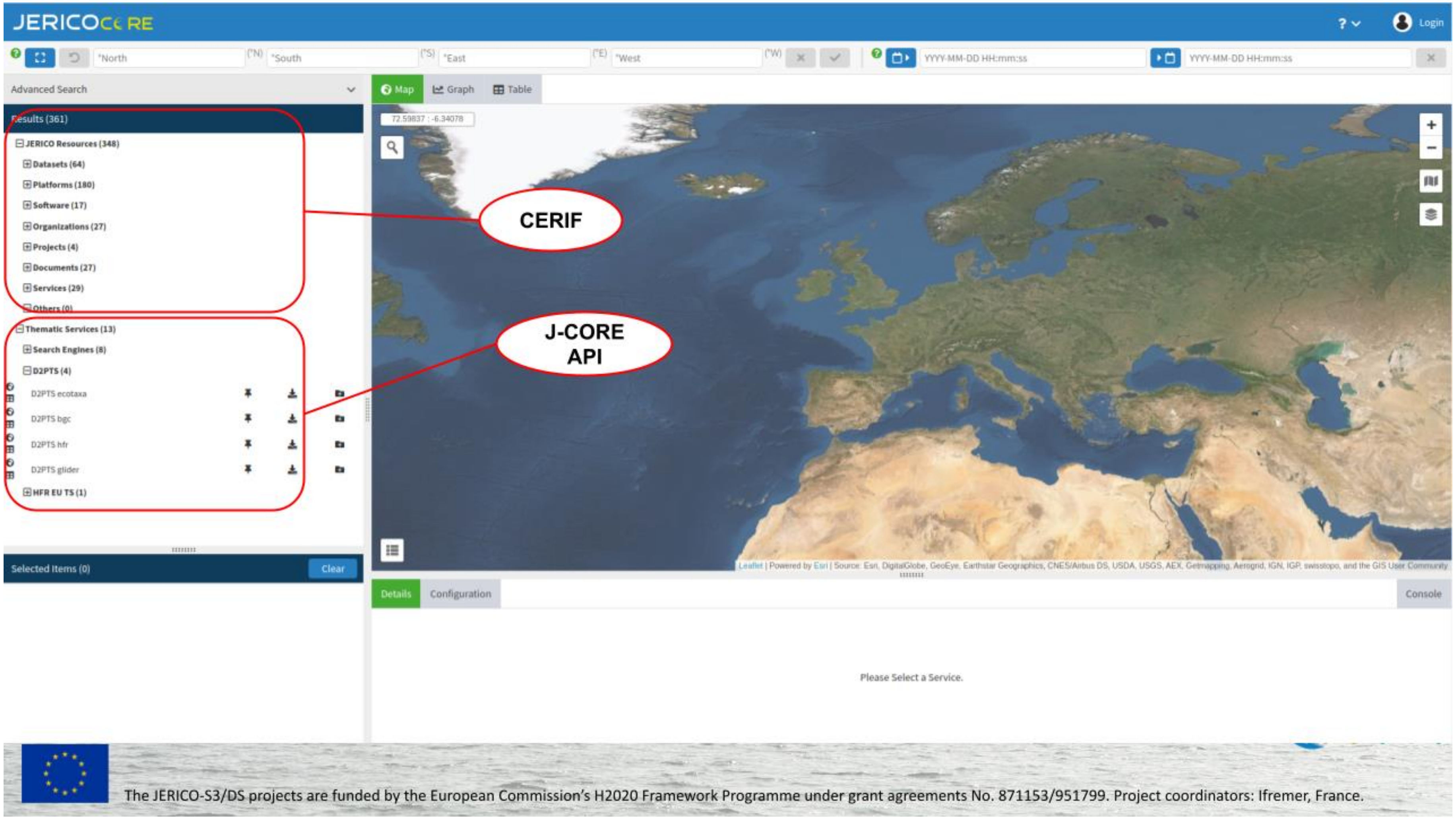

Figure 30: Main view of the EJCUI - The source of the stored information of the asset list in the left bar are shown.

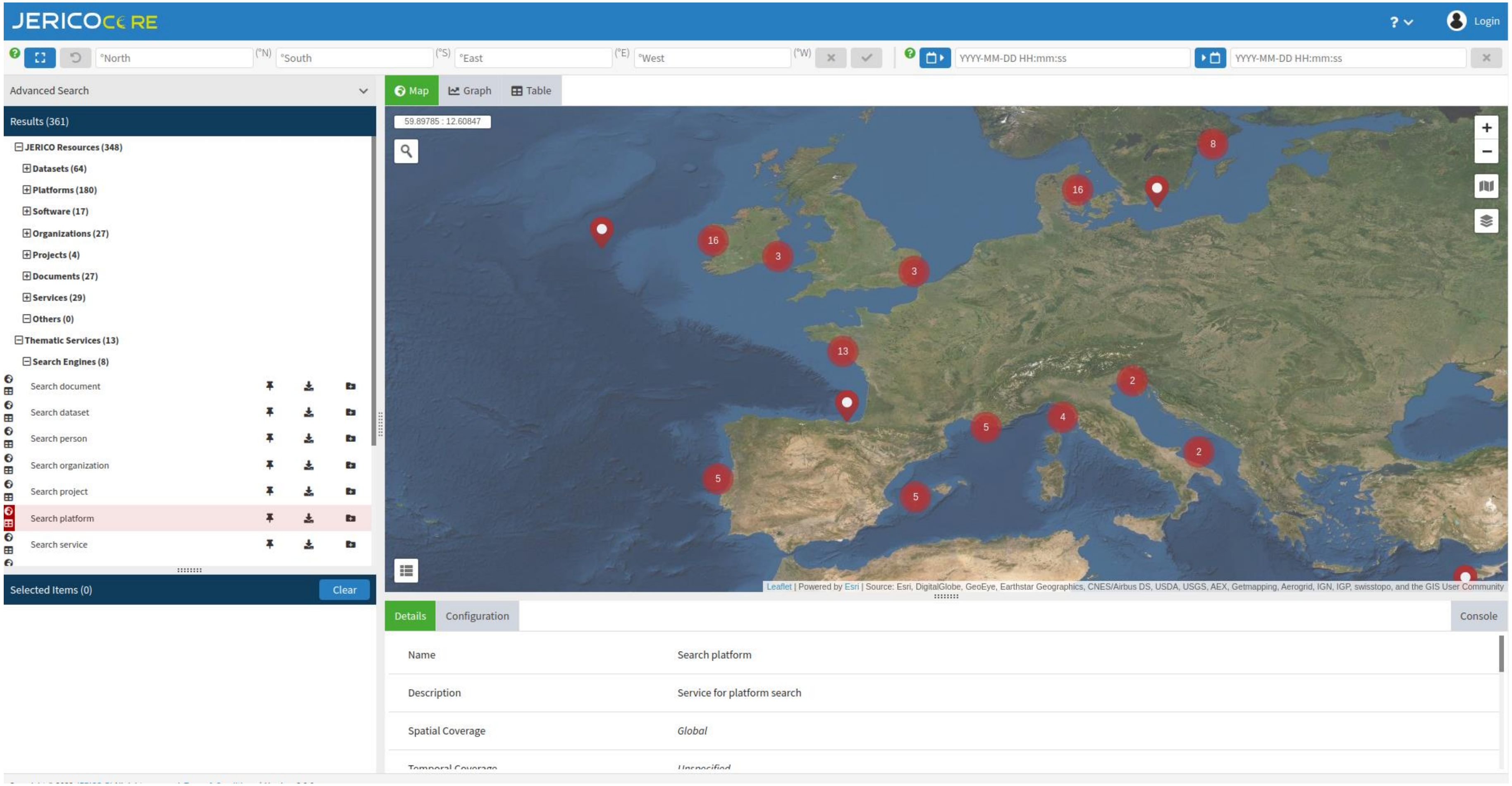

Figure 31: Main view of the EJCUI - The results of the platform service are shown on the map of the main panel.

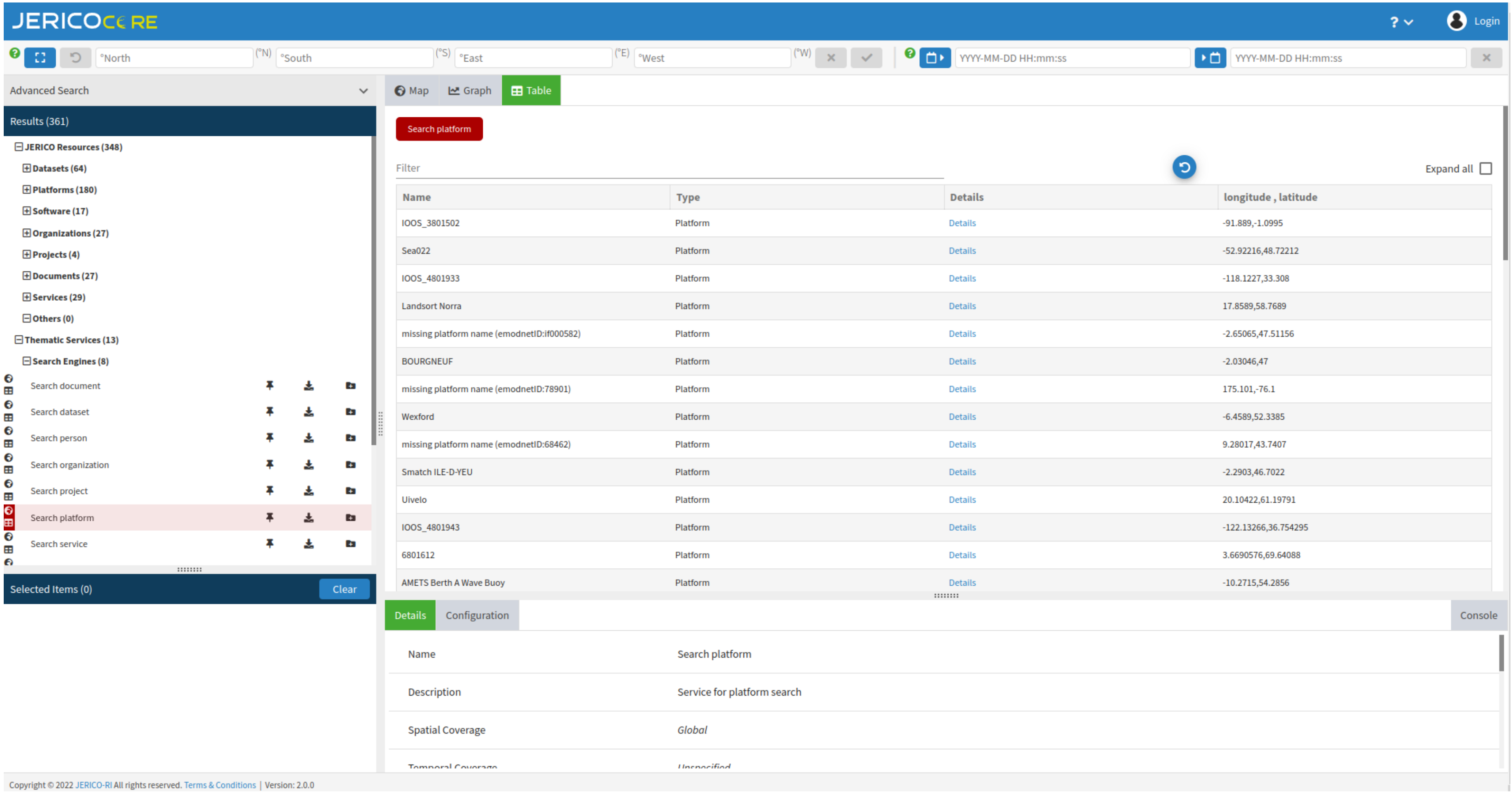

Figure 32: Main view of the EJCUI - The results of the platform service are shown on the table of the main panel.

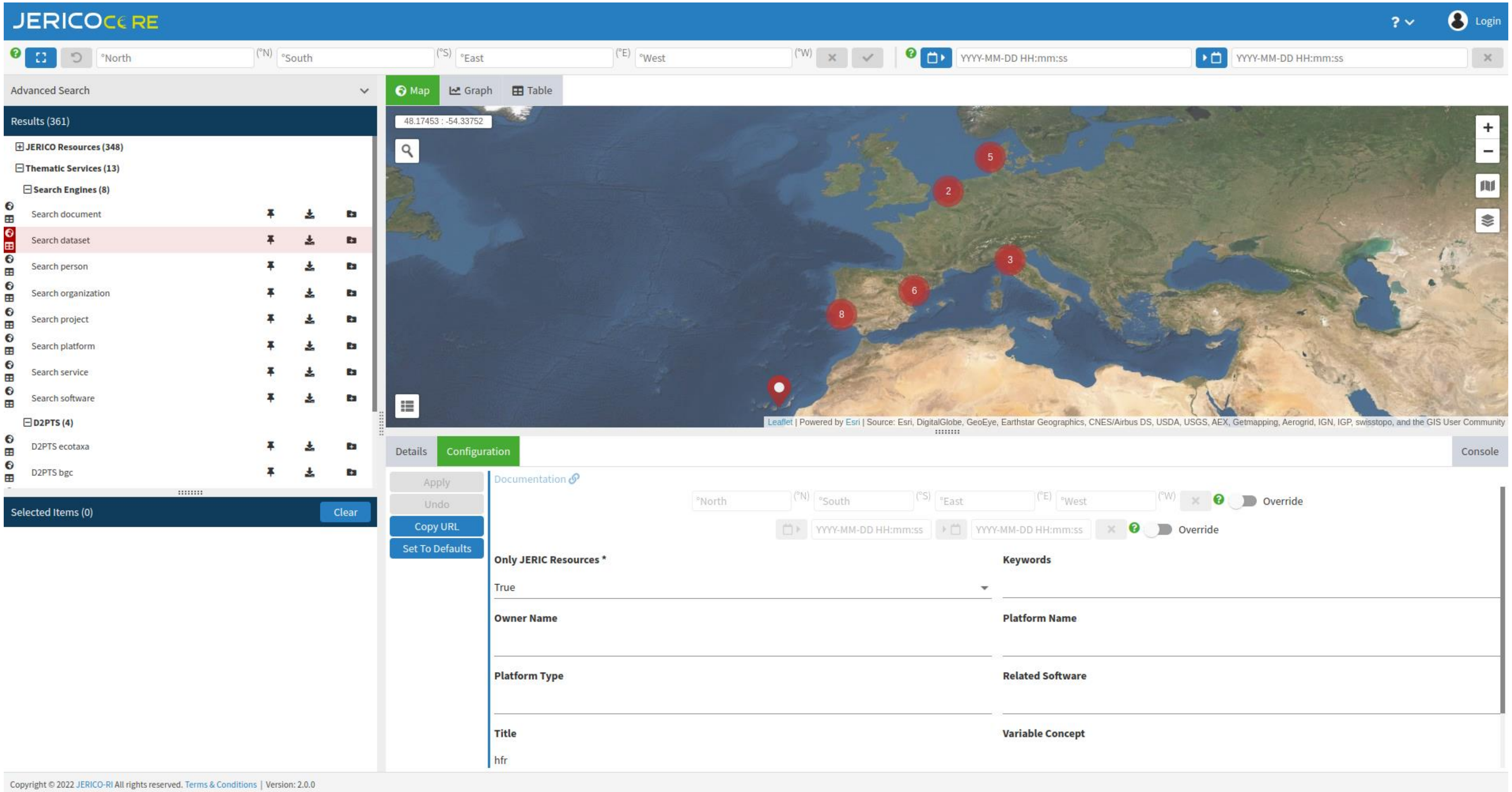

Figure 33: Main view of the EJCUI - The filters for the dataset search service are shown on the lower panel.

EJCUI has four main areas (see figures 30, 31, 32 and 33):

- Top bar: allows applying geospatial and temporal filters.
- Left panel: shows resources and services and allows filtering resources and services using free text. The asset information is separated into two categories (see figure 30). At the top, the list of JERICO resources are shown individually. The list is recovered from the CERIF catalogue. At the bottom, services are listed allowing users to perform advanced searches in the knowledge catalogue, including listing resources of the D2PTS. The represented information is retrieved through the JERICO-CORE API.
- Right-top panel: shows results in three different formats
    - Maps: shows geolocated assets and pop-ups with additional information (see figure 31)
    - Graphs: shows datasets when returned by a WMS or WFS. This functionality is not yet available in the JERICO-CORE APIs.
    - Tables: shows list of resources in the form of a table (see figure 32).
- Right-bottom panel: shows the information of the selected service or asset and allows configuring the service using different criteria.

3.7.9. Monitoring and Auxiliary Service Layer

In order to optimise resources for other activities of the architecture, we used the existing solutions that are available in Datarmore and under WP11 of JERICO-S3. Additionally, we used SENTRY to catch exceptions in the code and help diagnose and fix issues in the API code. We tried to incorporate SENTRY in EJCUI but unfortunately the EPOS libraries were not compatible with the ones needed for SENTRY. The change to integrate SENTRY was outside our possibilities under JERICO-S3.

Datarmor monitoring system is an in-house solution that uses syslog and Docker syslog driver in combination with an in-house script run in systemd. Datarmor architecture does not use a docker orchestrator and rely on redundancies and resilience provided by the IT team. In order to monitor the software processes they combine Nagios for alerting and Collectd/Grafana for metrics monitoring.

Concerning access metrics, a central system to monitor the access to each VA service has been created in the context of WP11. It is called the Virtual Access Metrics System (VAMS). As a VA of WP11, JERICO-CORE must integrate to VAMS which contains a database with information on how and when the infrastructure is accessed (i.e. Virtual Access) and one dashboard illustrating this information. VAMS collects metrics from a diverse range of data sources. The two main types of data sources are log files and Google Analytics API. For most of the infrastructures, VAMS collects metrics automatically. However, for security and legal reasons, the JERICO-CORE access metrics cannot be accessed automatically from outside DATARMOR. AWStats is a reporting tool to analyse data from the internet and report in tables and bar graphs. A Google Form [6] was created under WP11 to allow the DATARMOR team to monthly report access to the EPOS UI and the JERICO-CORE API. VAMS is also an important piece of JERICO-CORE that supports the activities of VA. Its purpose is to provide a central point where the access metrics can be viewed. More information of the architecture can be found in deliverable D11.1 [2].

The resulting services and tools developed in the context of task 7.5 were transferred to WP11 as two VA ID1.1 and ID33.2. These VA are described in the DoW as:

- **ID1.1**: "*Datarmor is an IT Infrastructure dedicated to Marine Sciences and located in Brest-France and funded in the framework of State-Region Plan by European Commission (FEDER funds), French Government, Brittany Region and local authorities in association with Ifremer and SHOM. Services currently offered by the infrastructure: host the JERICO e-Infrastructure during the project, including Virtual Research Environment and the associated data, when High Performance Computing and/or powerful access to data will be required.*"

- **ID33.2**: "*VA framework that provides access to the most important JERICO Resources (Catalogued) and Pilot Data-to-Products Thematic Services (D2PTS). Services currently offered by the infrastructure: maintain and operate the JERICO RI e-Infrastructure during the JERICO-S3 project, including the VA portal interface and backend, resources catalogues, access to new data, functionalities and both tools and services developed in TA, NAs and JRAs.Support for extraction of large data volumes of selected in situ data from PSSs and IRSs will be offered.*"

Therefore, the hardware infrastructure used for JERICO-CORE is Datarmor and will be operated under ID1.1. It will host the implemented components described in this section. The maintenance and operation of these components are done under ID33.2, including the VA portal interface and backend, resources catalogues, access to new data, functionalities and both tools and services. For this transition to happen, we developed development and deployment procedures. These procedures are explained in the next section.

3.8.Development and operation procedures and environments

The development and operation happen in two different infrastructures. While the production environment is provided by Datarmor in the context of D1.1 of WP11, the development and validation occurs in the Coastal Ocean Observing and Forecasting System located in the Balearic Islands (SOCIB) infrastructure. Therefore, the development and deployment procedures account for the existence of these two environments which are very different by nature (see figure 34). Both of them use CI/CD but perform their own validation and testing procedures under different gitlab environments.

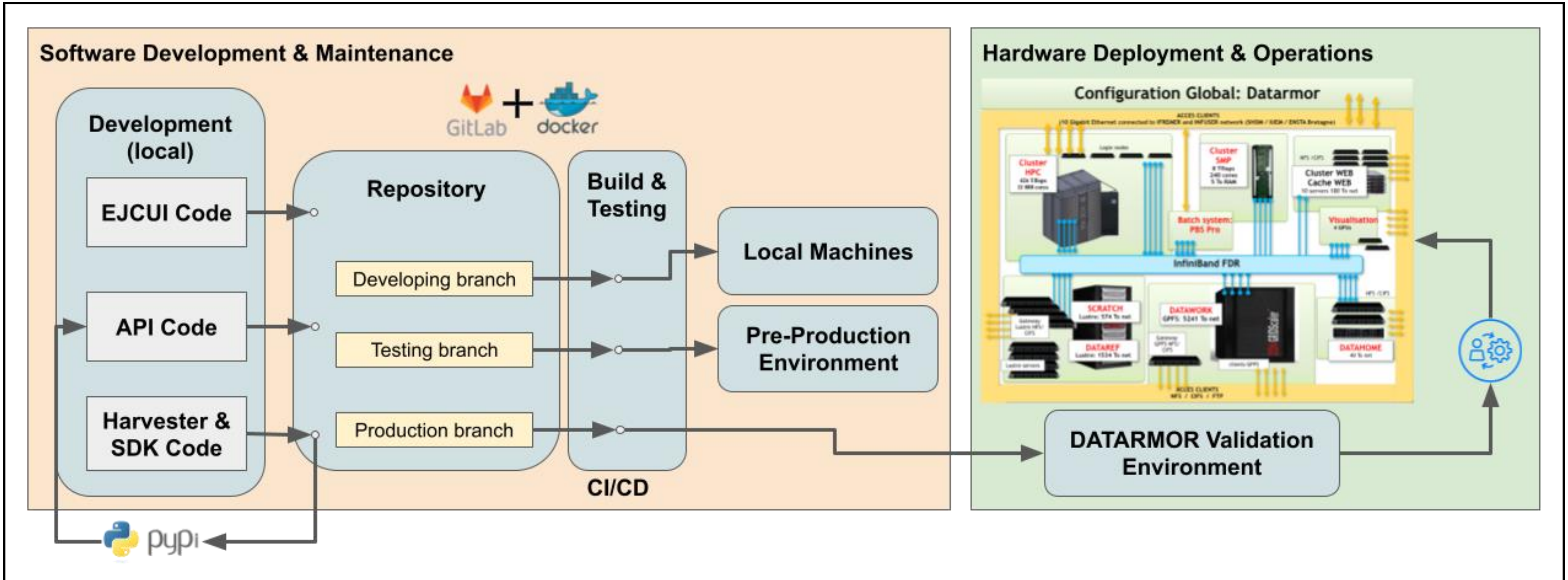

Figure 34: JERICO-CORE software life cycle. Development and deployment environments are shown in the CI/CD process.

Development of the EJCUI, API, harvesters and SDK codes happens in the SOCIB infrastructure (see figure 34). We created three stages of the software life cycle. Development happens in the developers' local machines. Because components are dockerized, environment variables are set using .env files. These are configured to act in the local environment. Once the code is pushed to the Gitlab repository, testing is performed automatically in the Gitlab environment and the software is deployed in the pre-production server when checks are passed. The configuration of the pre-production environment is set using Gitlab CI/CD variables. Once the pre-production version is validated by users the code is tagged and Gitlab creates an image in the Gitlab container registry. This version is transferred to Datarmor through the mechanisms explained below. We differentiate in the release and deployment processes a different scenario for each of the four codes: EJCUI, SDK, API and harvesters. These are explained in the next sections.

3.8.1. EJCUI release and deployment processes

Because EJCUI is not tight to the SDK, the release and deployment can be performed independently. The EJCUI code (https://gitlab.priv.socib.es/jerico/epos) is hosted in 10 Gitlab projects:

- DBConnector
- EPOS-CUI
- Ingestor
- Mapper
- Metadata-database
- Node_angular_epos
- QueryGen
- RoutingFramework
- TCSConnector
- WebAPI

Additionally, a repository (fuseki-tester) was created to test the connection between the EPOS components and the Apache Jena Fuseki server during the process used to ingest resource information into CERIF. Each of these components are released in their own container registry under their Gitlab project. A separate project EPOS DockerCompose was created to deploy EJCUI in the SOCIB infrastructure. The release of a new version requires the update of the docker-compose.yml file under this repository. This file defines which module images are deployed and the environment variables.

The deployment of EJCUI in the production environment of Datarmor. The configuration file https://gitlab.ifremer.fr/dev-ops/isival/isival_application/-/blob/master/group_vars/all/vars.yml defines the deployment images and variables. EJCUI is dockerized and the docker containers are defined as follow:

- Jfront_rabbitmq3
- Jfront_db
- Jfront_webapi
- Jfront_mapper
- Jfront_dbconnector
- Jfront_querygen
- Jfront_tcsconnector

- Jfront_ingestor
- Jfront_webgui

Due to the restrictions of Datarmor, the webapi module cannot be used to upload TTL files. We approach the problem by creating a docker image that includes the loaded resource catalogue in CERIF. This is not necessary in the pre-production environment and gives more flexibility for testing and validation before the final deployment. Unfortunately, the real time update of the resource catalogue in production cannot be contemplated in this context.

3.8.2. SDK release and deployment processes

The SDK code does not change frequently but updates may have a strong impact in other components such as APIs and harvesters. As for the other components, we use Gitlab to host the SDK project (https://gitlab.priv.socib.es/data-center/ejerico_harvester). We use Pypi to share the validated versions. A tagged version in the master branch is pushed as a new version at https://pypi.org/project/ejerico-harvester/. This version can be installed in other components using "pip install ejerico-harvester". Figure 35 illustrates the entire software cycle.

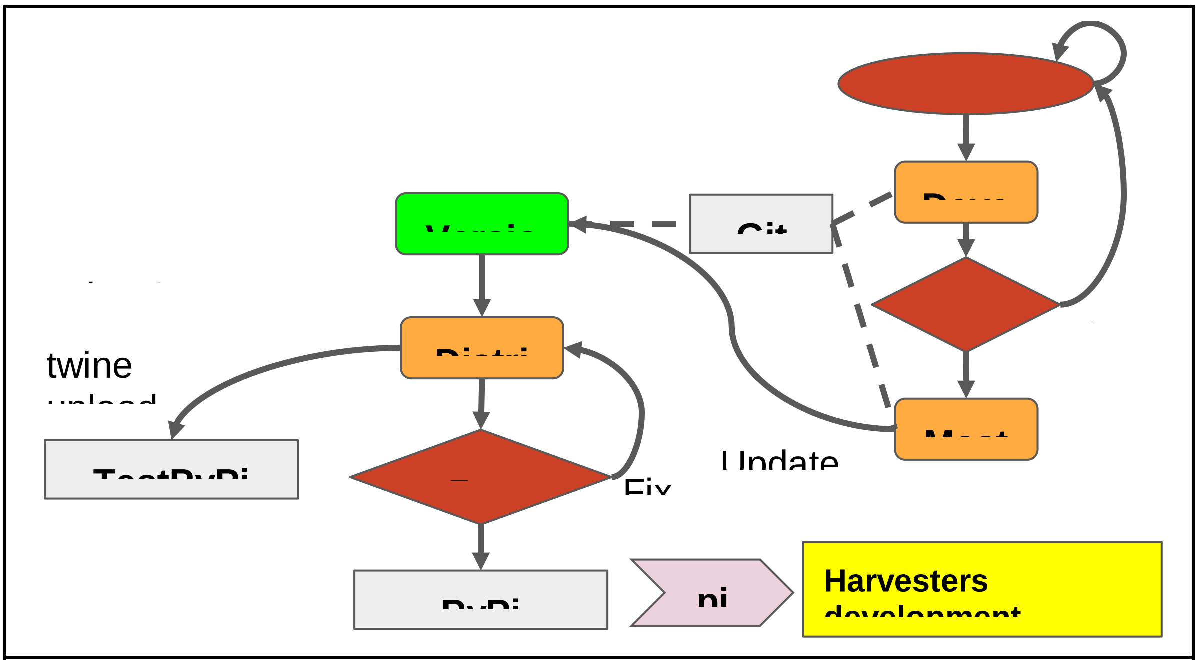

Figure 35: JERICO-CORE SDK development and release process

3.8.3. Harvester release and deployment processes

In order to develop harvesters it is necessary to install the SDK libraries that include the generic SDK and the harvester SDK with an empty template to facilitate the development. Figure 36 shows the mechanisms to set up the harvester development environment. The existing harvester Gitlab projects are as follow:

- ejerico_harvester_dspace
- ejerico_harvester_emodnet
- ejerico_harvester_emodnet_biology
- ejerico_harvester_epos_gateway

- ejerico_harvester_erddap
- ejerico_harvester_github
- ejerico_harvester_metadata_loader
- ejerico_harvester_ramadda
- ejerico_harvester_seadatanet
- ejerico_harvester_sos
- Ejerico_harvester_thredds

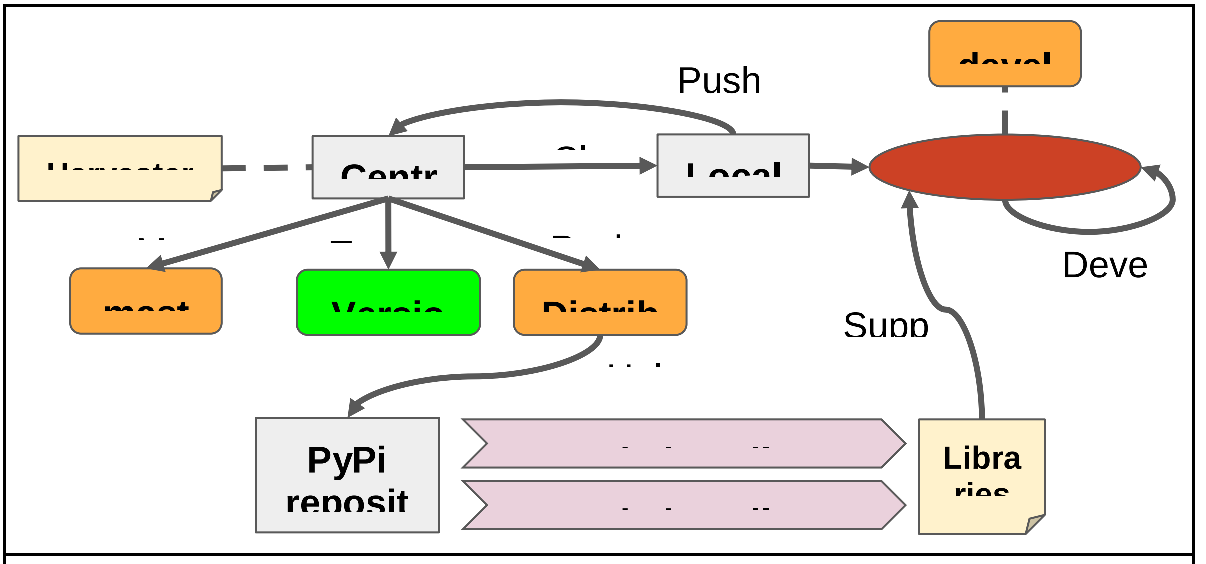

Figure 36: Setup Harvester Development Environment

For practical reasons, harvesters are deployed and operated in SOCIB. In fact, due to the variability of the metadata provided to the harvesters (see Harvesting), the process often requires human intervention which is not possible in the Datarmor infrastructure because there is no ssh available to track the harvester processing.

3.8.4. API release and deployment processes

The JERICO-CORE API code follows the complete software lifecycle defined in figure 34. In fact, an additional step is performed that instals the production image in a SOCIB server for local validations before the transfer to the production environment in Datarmor. This version is externally available at http://api.core.jerico-ri.test.socib.es/. In order to support this, we make available a jena server (http://jena.core.jerico-ri.test.socib.es/) and an EJCUI infrastructure (http://ui.core.jerico-ri.test.socib.es/). The availability of these links is not guaranteed since they still belong to the development environment. But they can be used with the user testing group to explore JERICO-CORE and provide feedback.

The deployment of the API in Datarmor is also configured using the file https://gitlab.ifremer.fr/dev-ops/isival/isival_application/-/blob/master/group_vars/all/vars.yml which defines the deployment images and variables. The JERICO-CORE API containers are as follow:

- jcore_db
- jcore_api
- jcore_nginx

The cache database is persisted and the update of a new API version does not affect the cached information. The cache should be updated using the db endpoint at the initial installation or when the resource catalogue of the Jena server is updated.

In summary, in this section we described the development and deployment cycle and procedures for the main components of JERICO-CORE: (1) SDK, (2) API and (3) EPOS UI. These procedures are consistent with the methodology that was studied in section 3.4.2.5 (Supporting technologies). The resulting implementation was deployed following these procedures and is available to the community at https://api.core.jerico-ri.eu and https://ui.core.jerico-ri.eu. The system has been shown to the community as shown in the next section.

3.9.Demonstration and future plans

JERICO-CORE was first demonstrated during the JERICO-Days in Lisbon on June 28th 2022. The purpose of this demonstration was to share the current status of the infrastructure, to collect feedback from JERICO partners and to perform a final validation of the solution before the transfer to operations. The demonstration was set in the production environment in SOCIB. We also performed an important manual cleaning of the resource catalogue to face the various obstacles encountered in the harvester mechanisms explained in section Harvesting. This includes:

- Merged entities that were duplicated due to the mismatch of identifiers between various providers
- Removed entities that did not contain metadata. These assets usually existed because they were in the metadata of another asset but not defined among the providers that were used for the JERICO-CORE prototype.
- Identified the JERICO resources. Although we need a clear definition of what is a JERICO resource at the consortium level, we made a certain number of assumptions. Organisations that were part of any JERICO project are considered as JERICO assets in the catalogue. Any asset that is linked to these organisations are considered JERICO. The graph is traversed except for some of the resource types such as the variables. Projects are also skipped because organisations could belong to a project that has partners that do not belong to JERICO. This would include assets of these organisations which clearly should not be labelled as JERICO.

The demo session included a hands-on demonstration and a presentation [7]. Notes and minutes are available in the Tuesday_JERICO-Core-JERICO-Days2022-SESSION [8] document. A certain number of questions and concerns were raised during the session:

- Difficulty of identifying JERICO resources. We need to clarify this at the JERICO-RI level. These issue was raised and it will be considered in future developments of JERICO-RI.
- The source of information should be expanded when resources are available in other projects. SeaDatanet datasets, Copernicus Marine Service and Argo are of particular interest.

- Difficulty of collecting data in real time because of the intensive manual work with the current approach. It is necessary to design a more flexible approach to collect resources that at the same time can adapt to the current variability and variety of metadata (see section Harvesting).
- Need to identify the key users and their needs. These needs must be documented for the long term plans in the ESFRI roadmap.
- Need to build on ENVRI-FAIR and engage with ENVRI community.
- Need to enhance through JERICO-CORE the role of JERICO in the coastal domain.
- Need to define the role of JERICO-CORE internally among the consortium partners.

These concerns must be addressed in the context of JERICO-DS WP3 for a long-term sustained infrastructure. Moreover, there are lessons learned along the way that should be considered. We include below a list of those and the developments that could not be implemented in the duration of the task 7.5.

- The harvesting process is of high importance but it is very delicate. We need another mechanism of harvesting the information that supports the existence of non-ideal metadata at the provider level. Ideally, this mechanism should support the process of identifying and correcting the gaps of the metadata at the provider level.
- It is important to perform careful work in defining the metadata schema that shows the relevant information of JERICO assets.
- The Blue-Cloud VLab currently is the system powering the VRE capacity of the JERICO-CORE prototype. This option will be studied in depth during the JERICO-DS project.
- We should update the architecture proposed in the context of JERICO-S3 based on new inputs of JERICO stakeholders. In particular, for the components that were not implemented in the context of task 7.5, we should consider if they need to be implemented in future versions of JERICO-CORE. These components are:
    - Peer-to-peer functionality should be re-evaluated
    - Complete Harvester SDK with scheduler
    - Enhanced monitoring capabilities with AUDIT, logger and network monitoring integrated.
    - Configuration manager of the different components of JERICO-CORE
    - RDF Queue ingestion layers
    - Clusterization of the Data Storage
    - Continue integration SWE sensors and Helgoland viewer
    - Integration/use of DDAS and/or ERDDAP for accessing datasets
    - Creation of a access layer to access federated resources
    - Develop API client
- We should continue creating services at the different working group levels. This will already be performed in the context of the Blue-Cloud 2026 project. But it is important that we promote the availability of JERICO-CORE to support the work of JERICO.

- We should propose other interfaces at the USER layer that are customised for specific needs of users.

- We should integrate JERICO-CORE to other external infrastructures. With the experience of integrating EPOS it will be possible to interoperate with the EPOS platform and create a Thematic Service that integrates coastal and oceanographic with Earth Science data. This will probably require some work at the API level and additional work creating WMS and WFS to return GIS datasets from the JERICO-CORE TS.

4. OUTREACH, DISSEMINATION AND COMMUNICATION ACTIVITIES

JERICO-CORE was presented within the JERICO community and among external stakeholders. For the JERICO community, we presented the project in individual meetings (see appendix B) and during general events such as JERICO-Weeks and JERICO-S3 General Assemblies. Due to the CoVid19 situation, most of the meetings happened remotely. The list of presentations in these general meetings in the context of JERICO-S3 are as follows:

- The JERICO e-infrastructure, e-JERICO - Remote, April 22, 2021
- Status and technical view of the development of JERICO-CORE, Remote, March 17th 2022
- Demonstration and Lessons learned, JERICO-CORE - Lisbon, June 28th, 2022

Concerning external stakeholders, we presented the infrastructure to various projects in addition to the stakeholders explained in section Analysis of existing data and knowledge e-infrastructures.

- The Center for the Fourth Industrial Revolution (C4IR) HubOcean https://www.hubocean.earth/
- The Glider European Research Infrastructure (GROOM) II project

5. CONCLUSIONS

In this document we describe the achievements and the process used under task 7.5 to design and implement JERICO-CORE, a digital infrastructure that aims to support the activities carried out by the JERICO community. In this process, we analysed technologies and infrastructure such as EPOS, Blue-Cloud, DataOne, AODN, ICOS, EMBRC, AGDC and e-VRE

We identified key features of these infrastructures available to implement features in the JERICO-CORE architecture in the short and long-term plans. In particular, we built synergies with EPOS and Blue-Cloud that we consolidated with two Memorandums of Understanding. We integrated the EPOS capabilities for the purpose of visualising the resources that were collected from federated infrastructures among JERICO partners. Additionally, the Blue-Cloud gateway was used to implement the JERICO Virtual Research Environment. Other features from EPOS and Blue-Cloud as well as from other infrastructures of this analysis are explained in this document to support the long-term design study in the context of JERICO-DS.

A scalable architecture was designed under task 7.5. We implemented parts of this design to explore the feasibility of the design and use lessons learned from this demonstrator as inputs of JERICO-DS. In particular, we implemented harvesters to collect the information from different providers, a Software Development Toolkit to support the creation of additional harvesters by providers and APIs to discover and access the resources that are collected in the JERICO-CORE resource catalogue. These components were integrated with the EPOS system to allow users to visually search and discover resources in the catalogue. Moreover, the Blue-Cloud was a key component to demonstrate the machine-to-machine capabilities of JERICO-CORE. Service demonstrators were programmed in this VRE to show the value of integrating all the capabilities of JERICO-CORE.

Finally, access to the system was done through three URLs:

- **https://api.core.jerico-ri.eu** provides access to the APIs interfacing with the JERICO-CORE resources catalogue
- **https://ui.core.jerico-ri.eu** provides users access to JERICO-CORE UI
- **https://blue-cloud.d4science.org/group/jerico_core/home** is the portal to the JERICO-CORE VRE that was implemented as a Virtual Lab of Blue-Cloud.

The current implementation of JERICO-CORE prototype was shown as a demonstrator (MS38) in Lisbon JERICO-Week. This milestone was key to inform the community of the existing status and features of the JERICO digital platform and allow the collection of new requirements and capabilities to be incorporated into JERICO-CORE.

The results and lessons learned of the development of the JERICO-CORE project are essential to further developments of the e-infrastructure. In particular, WP3 of the JERICO-DS project is studying the design options for the development of a JERICO-CORE that supports the JERICO-RI in the ESFRI roadmap. Additionally, JERICO is playing a strong role in the advancements of Coastal Ocean Research Activities with impact around the world. The concept and plans of the JERICO-CORE was brought to the international level under the UN Ocean Decade program. The Coastal Ocean Resource Infrastructure System (CORIS) was born from the experience gained during the development of the JERICO-CORE. CORIS was officially endorsed as a UN Ocean Decade Action hosted by the CoastPredict Programme as a core project in October 2022. The results and plans of JERICO-CORE and CORIS were presented at the CoastPredict General Assembly in Bologna in January 2023 and were recognized as a key infrastructure for the development of coastal activities worldwide.

6. ANNEXES AND REFERENCES

References

[1] Ocean Science Data - Collection, Management, Networking and Services Chapter Three - Data management infrastructures and their practices in Europe. Dick M.A.Schaap, Antonio Novellino, Michele Fichaut, Giuseppe M.R.Manzella
https://www.sciencedirect.com/science/article/pii/B9780128234273000074#!

[2] JERICO-S3 D11.1 First report on VA JERICO Resources access statistics and service provision. https://www.jerico-ri.eu/download/jerico-s3_deliverables/JERICO-S3-D11.1-First-report-on-VA-JERICO-Resources-access-statistics-and-service-provision-FINAL-V2-Compressed.pdf

[3] ENVRI-FAIR D7.3 - ENVRI-FAIR Knowledge Base for RI Service Interoperation and Competence. https://envri.eu/wp-content/uploads/2021/04/ENVRI-FAIR-D-7-3.pdf

[4] D7.5 D2PTS deliverable - https://docs.google.com/document/d/1SiSNcOgPkIOZZXBUSSmQUzEq0pFQLboP6-N_r0xi-TM/edit?userstoinvite=milla99fmi@gmail.com&actionButton=1

[5] Google form for JERICO Research Infrastructure User Stories. https://docs.google.com/forms/d/e/1FAIpQLSd4h0UQ9SI4DMZHK2l3VxzH7_UCl0idfHIbIiyf-T7HyOKMSA/viewform

[6] Google form for DATARMOR metrics of access to JERICO-CORE API and EPOS JERICO-CORE UI. https://docs.google.com/forms/d/e/1FAIpQLScEPIGOGrBD4-CO92Tf0q32-hcRaT61Y7n2dJ_nZk_oBjxaHg/viewform

[7] Presentation JERICO-Days - "Demonstration and Lessons learned JERICO-CORE". https://drive.google.com/file/d/1KAXx7hpJvKYkZ3JrriLpDxHYzCScHpVn/view?usp=sharing

[8] Notes of JERICO-CORE demonstration session https://docs.google.com/document/d/1OGUFPscHxwW9HxxG1ILG_VcuP15HtfNsJ4kFskGrXqg/edit#heading=h.dplq056pmkio

[9] JERICO-S3-D6.1- Data-Management-Plan Final https://www.jerico-ri.eu/download/jerico-s3_deliverables/JERICO-S3-D6.1-Data-Management-Plan_final_2.pdf

[10] Nature Ecology & Evolution volume 6, pages 1262–1270 (2022) - A global horizon scan of issues impacting marine and coastal biodiversity conservation. James E. Herbert-Read, Ann Thornton, Diva J. Amon, Silvana N. R. Birchenough, Isabelle M. Côté, Maria P. Dias, Brendan J. Godley, Sally A. Keith, Emma McKinley, Lloyd S. Peck, Ricardo Calado, Omar Defeo, Steven Degraer, Emma L. Johnston, Hermanni Kaartokallio, Peter I. Macreadie, Anna Metaxas, Agnes W. N. Muthumbi, David O. Obura, David M. Paterson, Alberto R. Piola, Anthony J. Richardson, Irene R. Schloss, Paul V. R. Snelgrove, Bryce D. Stewart, Paul M. Thompson, Gordon J. Watson, Thomas A. Worthington, Moriaki Yasuhara & William J. Sutherland. https://www.nature.com/articles/s41559-022-01812-0

[11] 3 Oceans and Coastal Ecosystems and Their Services - IPCC - Oceans and Coastal Ecosystems and Their Services. Sarah R. Cooley (USA) and David S. Schoeman (Australia). https://www.ipcc.ch/report/ar6/wg2/downloads/report/IPCC_AR6_WGII_Chapter03.pdf

[12] Blue-Cloud Demonstrator Users Handbook V2 - Dominique OBATON, Gilbert MAUDIRE https://blue-cloud.org/support-centre/support-material/blue-cloud-demonstrator-users-handbook-v2

Glossary

Abbreviation	Description
AMS	Alias Management System
API	Application Programming Interface
BC	Blue Cloud
CERIF	Common European Research Information Format
CRIS	Current Research Information System
D2PTS	Data to Product Thematic Service
DD&AS	Data Discovery and Access Service
DoW	Description of Work
EDMERP	European Directory of Marine Environmental Research Projects
EDMO	European Directory of Marine Organisations
EJCUI	EPOS JERICO-CORE User Interface
ESFRI	European STRategy Forum on Research Infrastructures
GSP	Graph Store Protocol
HOORT	HFR Online Outage Reporting Tool
IRS	Integrated Regional Sites
JERICO	Joint European Research Infrastructure for Coastal Observatories
JSON	JavaScript Object Notation
JSON-LD	JSON Linking Data
KIPI	Key Integration Performance Indicators
KPPI	Key Platform Performance Indicators
M2M	Machine-to-Machine
MoU	Memorandum of Understanding
NetCDF	Network Common Data Form
PSS	Pilot Super Site

REST	Representational State Transfer
RI	Research Infrastructure
ROOS	Regional Operational Oceanographic Systems
SDK	Software Development Toolkit
SOCIB	Balearic Islands Coastal Observing and Forecasting System
SPARQL	SPARQL Protocol and RDF Query Language
TS	Thematic Service
TTL	Terse RDF Triple Language
VRE	Virtual Research Environment
WFS	Web Feature Service
WMS	Web Map Service
OWL	Web Ontology Language
WWW	World Wide Web
W3C	World Wide Web Consortium (W3C)

Appendix A: Aspects of JERICO-CORE collected from JERICO and JERICO-CORE deliverables

In this appendix we organise the information of the documents listed in table 1. We separate the information in three categories:

- Table A-1: Main components and interfaces
- Table A-2: Data integration
- Table A-3: Service integration

JERICO-NEXT	JERICO-S3	Reference	Project	Notes
List of data providers and platforms per European region	Resource catalogue DB	Deliverable 2.3	FP7	Probably outdated or incomplete
MyOcean catalogue	Resource catalogue DB (UI ??)	Deliverable 5.3	FP7	
Survey of availability indicators of VA services	Indicators and resource catalogue DBB	Deliverable 5.16	NEXT	
Aggregated catalogue of platforms and instruments	Resource catalogue DB	Deliverable 5.2	NEXT	In Sextant tool and JERICO-NEXT webpage
Dedicated interface[1] to input resources	Resource catalogue interface	Deliverable 5.2	NEXT	Data collection with questionnaire??
JERICO-NEXT website	e-Infrastructure UI	Deliverable 8.7	NEXT	
JERICO-NEXT Dashboard & monitoring tools	KPIs, data monitoring tools	Deliverable 8.8	NEXT	
Integrated WMS and WFS		Deliverable 8.8	NEXT	
Thredds catalogue		Deliverable 8.8	NEXT	
ERDDAP catalogue		Deliverable 8.8	NEXT	
Observatory operator console requirements and design	Metadata recording (resource catalogue)	Deliverable 5.6	NEXT	

Table A-1: Outcomes from JERICO and JERICO-NEXT that could potentially be used for the development of the core and interfaces of JERICO-CORE in the framework of JERICO-S3.

[1] According to deliverable 5.2, "The catalogue provides a dedicated interface in order to allow operators to describe their platform, the instruments, and in this way publish and advertise the work of data acquisition"

Previous JERICO	JERICO-S3	Reference	Project	Notes
JERICO data	Resource catalogue DB and data discovery	Deliverable 5.7	FP7	
JERICO data RT and DM flows	Data management plans	Deliverable 5.6 Deliverable 5.8	FP7	
Enhance and simplify the information and access of JERICO-NEXT data	Data discovery	Deliverable 5.2	NEXT	Should evaluate if we will provide this service
JERICO-NEXT data portal	Catalogue to data and data products	Deliverable 8.8	NEXT	Dynamic map page

Table A-2: Outcomes from JERICO and JERICO-NEXT that could potentially be used for the development of integration of data systems of JERICO-CORE in the framework of JERICO-S3.

JERICO-FP7	JERICO-S3	Reference	Project	Notes
Demo of global service	IRS products	Deliverable 2.4	FP7	
Data tools and services	Service discovery and tools	Deliverable 7.1	FP7	
VA services visualisation map	Service discovery and visualisation tools	Deliverable 8.2 Deliverable 8.3	NEXT	
Benthic Non Native Species Tool (private github)	Visualisation tool (service?)	Deliverable 8.3	NEXT	adapted from another application
PHYTO-OPS displays ferrybox data (private github)	Visualisation tool (service?)	Deliverable 8.3	NEXT	
List of services	Service integration list	Table p8 of deliverable 5.1	NEXT	

Table A-3: Outcomes from JERICO-FP7 and JERICO-NEXT that could potentially be used for the development of integration of service systems of JERICO-CORE in the framework of JERICO-S3.

Appendix B: JERICO-CORE co-design meetings and outcomes

List of meetings carried out during the initial co-design process when collecting requirements and defining the conceptual framework with JERICO partners and external stakeholders. Column T describes the type of meetings. Values are P for meetings in person or R for remote meetings. These meetings derived from the interfaces identified in section 3.1.2.

Date	Description	T	Outcome
2020-02-10 2020-02-11 internal	Preparation for the kick-off (KO) meeting. Discussions of JERICO-CORE requirements Resources to be catalogued Discussion WP interfaces Discussion of D2PTS Schedules and milestones First study existing infrastructures	P	Definition of JERICO-CORE framework. Strategy to approach the design of the infrastructure. List of priorities and actions for the initial phase of the project. KO meeting materials: architecture framework, architecture components, T7.5 initial timeline,
2020-02-17 KO meeting	WP7 focussed on task 7.5 to coordinate with WP6. Requirements gathering. Strategy during first year	P	
2020-02-17 KO meeting	Reach a common understanding of WP7 objectives and schedule Team building for product developments Learn on WASP status and connectivity requirements Discuss and log potential issues and risks, plan accordingly Learn on constraints for installing new sensors on JIIM Learn on sensors/samplers to inform on their connectivity requirements. Identify the technical gaps to be resolved for end-to-end observing chain Opportunities for instrument availability (CNR camera, BRAAVOO, NeXOS hydrophones / optics, … Software methodologies study and development	P	
2020-02-17 KO meeting	Interaction with WP1 WP3 and WP4 & Discuss possible use case(s) for the PSS /IRS demo with WP1	P	
2020-02-17 KO meeting	WP5 review of the status on existing BPs; Other parallel projects/initiatives (EuroSEA, OBPS..); Gaps in BPs; How to converge towards a homogenised handbook; KPIs; ling	P	

	with WP7-VRE		
2020-02-17 KO meeting	Interaction with WP1 WP3 and WP4	P	
2020-02-18 KO meeting	WP7 focussed on task 7.5 (e-infrastructure): - Coordinate (and invite) with WP4 - Requirements gathering - Strategy during first year	P	
2020-02-18 KO meeting	WP7 focussed on task 7.5 (e-infrastructure): - Coordinate (and invite) with WP5 - Requirements gathering - Strategy during first year	P	
2020-02-18 KO meeting	WP1-WP7 Objectives and mechanisms for WP1-WP7 interactions Feedback from WP7 into Task 1.3 – Task 1.4 (mid-term, can also be done later on, around M6)	P	
2020-02-20 KO meeting	WP7 focussed on task 7.5 (e-infrastructure): - Coordinate (and invite) with WP3 - Requirements gathering - Strategy during first year	P	
2020-03-06 internal	Review KO meeting and current status of T7.5 - Roles of various elements in resource flow - Discussions about steering committee, outreach aspects, IMDIS papers - Management and planning - Technical solutions	R	Diagrams Data Flow Diagram and Resource Catalogue Interfaces JERICO-CORE Service Components diagram Steering committee name list
2020-03-19 EMODnet	Discuss role of EMODnet in data flow: - Review flow and component diagrams - Review platform survey spreadsheet - Review roles of key player infrastructures - Review interfaces with EMODnet (WS)	R	EMODnetPhy@_M2M_JS3Pat.pptx
2020-03-23 internal	Discuss T7.5 status: - Interfaces to other WPs - Architecture - Review discussions and outcomes from meeting with EMODnet	R	
2020-03-25 2020-04-06	Interface T7.5 and WP6 - Review T7.5 diagrams	R	Refined architecture diagrams

WP6	- Discuss schedule and deadlines (T7.5) - Review WP6 and interface of WP6 tasks to T7.5 - Discuss schedule and deadlines (WP6) - Interface with SDN - Discussion of role of EMODnet concerning metadata (equipment and data) and information access - Discuss JERICO vs European coastal data sets - Agreement for next common actions (list with items and dates)		Data Flow Diagram and Resource Catalogue Interfaces JERICO-CORE Service Components Update schedule and deadlines Agreement on flow of various resources in particular related to data sets. Diagram with SOS Proposal from Dick Schaap to integrate main infrastructures in Europe
2020-03-26 internal	Review T7.5 status	R	
2020-03-31 EMODnet	Interface 7.5 and EMODnet	R	
2020-04-01 WP5 & WP6	Interface T7.5, WP5 and WP6	R	
2020-04-02 internal	Discuss T7.5 status: - Review Peter's suggestions (see diagram) - Discuss infrastructure baseline and select one - Create a duplicate of the overview - storyboard (do before meeting) and then talk about phase 1. - Look at details of the storyboard to simplify if possible.	R	Plans of components to be developed Data Flow Simplified Diagram and Resource Catalogue Interfaces
2020-04-07 internal	Interface T7.5 and WP5: - Review JERICO-CORE Resource Flow - Review WP5 and interface of WP5 tasks to T7.5 - Discuss schedule and deadlines (WP5) - Discuss schedule and deadlines (T7.5) - Agreement for next common actions (list with items and dates)	R	Defined workflow for resources related to KKPI, KIPI, calibrations and operation manuals
2020-04-14 internal	Review of existing infrastructures for JERICO-CORE use: - Review document about the study of existing e-Infrastructures (this document) - Review schedule and planning - Plan next steps	R	Initial decision of the role of each infrastructure concerning JERICO-CORE New diagrams: JERICO Resource Flow Diagram (Big Picture)

2020-04-15 WP6	Interface T7.5 and T6.4.1.2	R	Software and Libraries survey spreadsheet Plans to collect information
2020-04-15 52North	Interface T7.5 and SWE sensors: - Overview of direction of T7.5 - Plans of 52N in JERICO - Demo of Helgoland - Review T6.4.1.2 survey to collect existing software info in JERICO	R	Common understanding of the role of components involved with SWE sensors Discuss and improve Software and Libraries survey spreadsheet
2020-04-16 D2PTS	Interface T7.5 and D2PTS services: - Review of current thinking for the JERICO-CORE virtual infrastructure - Phased development of D2PTS: HFR tailored products What are we doing? on what schedule (particularly for 2020 and early 2021)? - Phased development of D2PTS: sea water masses and transport (idem) - Phased development of D2PTS: BGC state of coastal areas (idem) - Phased development of D2PTS: JERICHO-Eco Taxa (idem) - Discussion - Summary of Actions.	R	Common understanding of objectives of each D2PTS and the role that the JERICO-CORE infrastructures plays in relation to them Templates
2020-04-16 internal	Status of T7.5: - Discussion about D2PTS meeting - Review JERICO-CORE Service Components V2 diagram - Review other material	R	JERICO-CORE Service Components (V2)

Table B-1: List of meetings during the co-design phase of JERICO-CORE

Appendix C: Summary of results of the analysis of the existing e-infrastructures

These tables show the definition of the feature in the first column. The way each e-infrastructure responds to these features are shown in subsequent columns. We colour coded the second colour to identify visually the level of implementation of the specific feature of the same row. Green indicates that the e-infrastructure fully responds to the needs of the feature of the row. Orange indicates that there is a partial match of this feature with the solution of the e-infrastructure or there is a possibility of adapting this solution to respond to the feature. Red indicates that the feature is not implemented and there is little chance to adapt it in the framework of JERICO-S3.

	EPOS	ICOS	DataOne	AODN	EMBRC
POC	Massimo Cocco Keith Jeffery	Alex Vermeulen	Matt Jones	Sebastien Mancini	Nicolas Pade
Status of Project	Transition to operational	Portal fully operational	Released Jul. 2012 and on-going improvement	On going and fully operational	EU RI
Help Support		Unknown			
Long-term Sustainability					
Potential Collaboration					
Code Access	Github	Github	Github SVN	Github	MDA and IMIS: the code is open source

Table C-1: Summary of the analysis of the e-infrastructures concerning reliability and reusability

	E	I	D	A	E
Support distributed Nodes (Data Centers, Integrated Regional Services and Pilot Super Sites) by providing access to and information of their resources					
Support Integration of distributed Services that run in the partners infrastructure but can be accessed from a single portal					
Integrate services or material repositories for education and outreach either in the main infrastructure or distributed					
Level of effort to register and integrate new infrastructures and their services to the system.					
Single Portal					
Machine-to-Machine capabilities					

Table C-2: Summary of the analysis of the e-infrastructures concerning service support

	E	I	D	A	E
Catalogue of linked Resources					
Includes a repository where partners can upload their data for public retrieval.					
Includes services to discover data living in remote locations such as data centres or aggregators.					
Includes a repository where partners can upload, install and run custom services.					
Includes services to discover existing partner services that live in remote locations. It also allows running these services in a transparent way.					
Includes a repository where partners can upload and share their custom code including software and libraries. Optionally, includes services for discovering code in external repositories.					
Allows partners to discover, make and share custom workflows and data pipelines that use the main or distributed services that are catalogued by the infrastructure.					
Includes a repository where partners can upload their document for public retrieval.					
Includes services to discover documents living in remote locations such as external organisations or other repositories.					
Includes services to discover publications living in remote locations such as external organisations or other repositories.					
Includes a repository where partners can upload their best practices for public retrieval.					
Includes services to discover best practices living in remote locations such as external organisations or other repositories.					

Table C-3: Summary of the analysis of the e-infrastructures concerning resource visibility and access

	E	I	D	A	E
Provides support for data management either with tools or services in the main infrastructure or among the distributed partners					
Provides support for data processing either with tools or services in the main infrastructure or among the distributed partners					
Provides support for data distribution to European aggregators either with tools or services in the main infrastructure or among the distributed partners					
Provides support for data visualization tools either in the main infrastructure					
Provides support for QC tools either in the main infrastructure					
Confirm the integrity of the distributed data or metadata.					
Data/Metadata Resilience and Preservation by having external storage or servers in other locations					

Table C-4: Summary of the analysis of the e-infrastructures concerning resource usage and sharing

	E	I	D	A	E M
Open Code					
The architecture design allows to scale the number of distributed nodes from a design point of view and when running increasing requests					
The Architecture contains a security layer that protects the infrastructure from cyberattacks					
Modular Architecture that is easy to adopt and flexible so it can be adapted to various needs (Easy Adoption)					
Documented Architecture					
Technical Performance					unclear
User perceived performance					unclear

Table C-5: Summary of the analysis of the e-infrastructures concerning architecture

Appendix D: Evolution of JERICO-CORE conceptual designs

This appendix shows various JERICO-CORE conceptual designs before the final concept (see figure 6) that were considered during the co-design process with JERICO partners and other stakeholders. In these views, various aspects of the resource providers and the collection methods were considered.

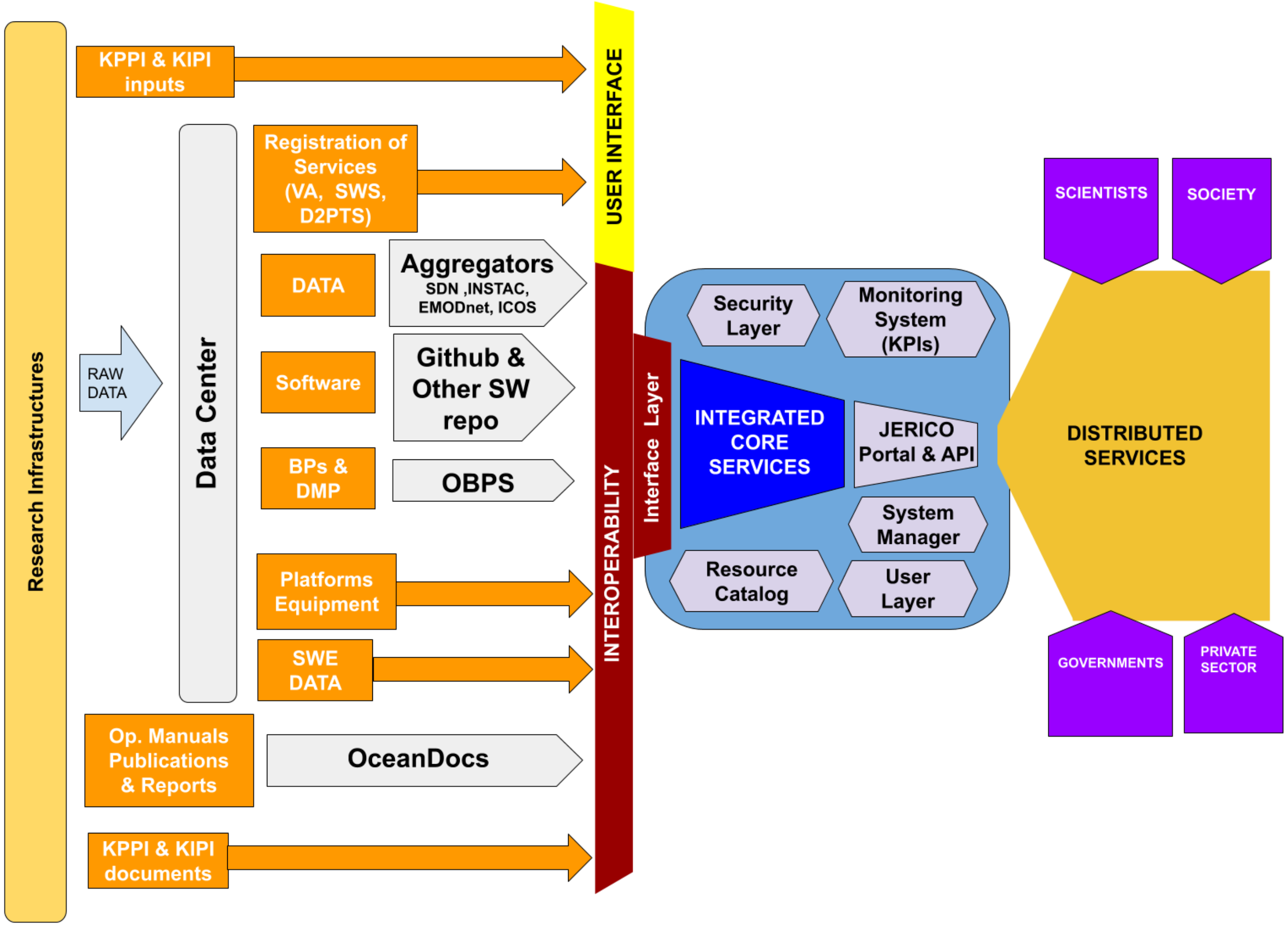

Figure D-1: General view of the flow of information in the process of collecting the data from the various locations. In the initial phases, we considered data centres of individual RI of JERICO partners as key elements for collecting resources. Inputs of indicators and registration of services were approached through a specific user interface for manual registration. The main core system on the right side was inspired by the EPOS e-infrastructure.

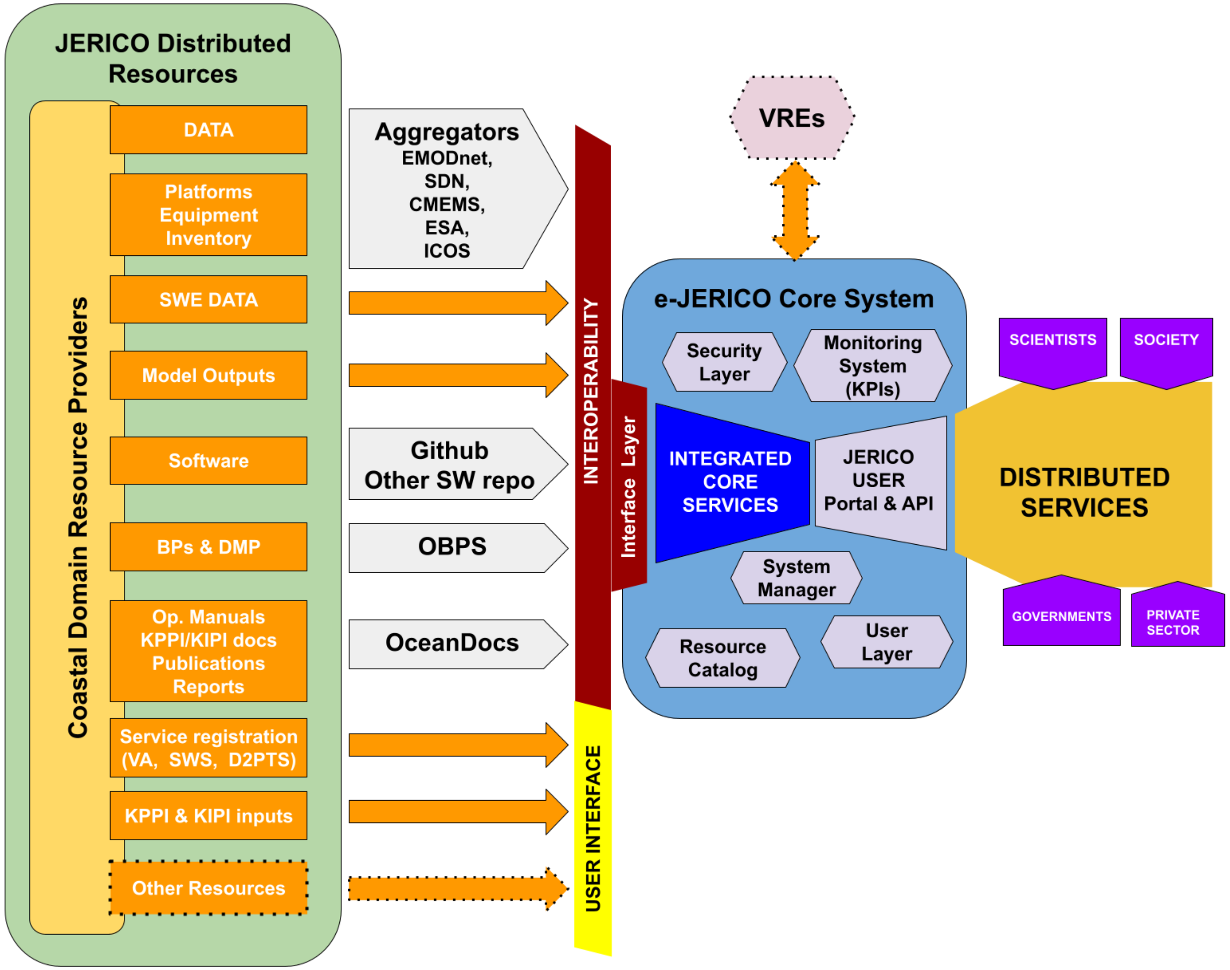

Figure D-2: General view of the flow of information in the process of collecting the data from the various locations. This view is similar to view of figure D-1 except that data centres are not considered as main providers among the RIs. We also considered the possibility of entering any type of resource through the UI that were not following the standard information flow.

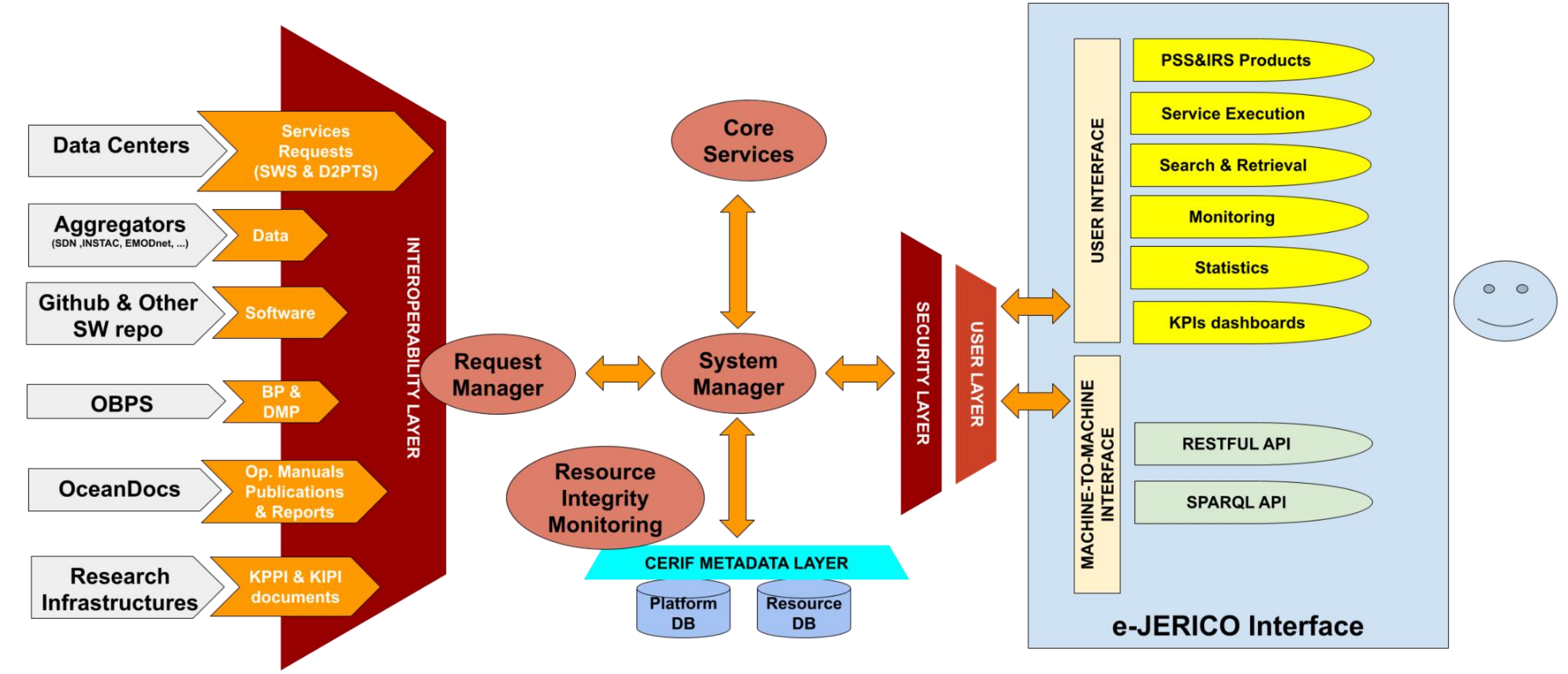

Figure D-3: General view of the discovery phase of the JERICO resources and access to distributed services. In this concept, we underestimated the role of the RI and we focused the harvesting in main provider infrastructures. Data centres still play a role in registering the information. The focus of this concept is in the way the information is provided to the users. We recognised the need to provide machine-to-machine endpoints and identified the main services to users. We also recognised the gap of a standard platform catalogue that would be automatically populated.

Appendix E: JERICO-CORE metadata schema

The following lists show the attributes of the entities of the metadata schema representing the JERICO assets as represented in the JERICO-CORE resource catalogue. All the entities include common attributes that are not represented in these lists. These attributes are the following:

- Name: name of the entity
- Keywords: main keywords used to label the entity
- Description: text describing the entities in a human readable format
- Spatial: link to a spatial entity representing the geographic location of the asset
- Temporal: link to a temporal entity representing the temporal range of the asset
- Language: language used to describe the entity
- Brand: context of the entity, usually used to identify JERICO resources
- Program: the project to which it belongs
- Concept: the semantic representation of this entity. This is not extensively used in the JERICO-S3 context

List of entities and their attributes are as follow:

- Api
  - Operation
  - Status_code
  - Title
  - Url
- ApiOperation
  - Method
  - Parameter
  - Url
- ApiOperationParameter
  - Method
  - Default_value
  - Label
  - Min_value
  - Max_value
  - Parameter
  - Required
  - Range
  - Value_pattern
- Address
  - Street_address
  - Postal_code
  - Locality
  - Country
  - Region
- Catalog
  - Accrual_periodicity
  - contactPoint

    - Creator
    - Dataset
    - Document
    - Equipment
    - Issued
    - Landing_page
    - Relation
    - Resolution
    - Service
    - Theme
    - Title
    - Variable
    - Owner
    - Contributor
    - Publisher
    - Author
    - Project
    - Distribution
- Concept
    - Alt_label
    - Broader
    - Change_note
    - Definition
    - Editorial_note
    - Example
    - Hidden_label
    - History_note
    - Member
    - Member_list
    - Narrower
    - Notation
    - Note
    - Pref_label
    - Related
    - Semantic_relation
    - Scope_note
    - In_scheme
- ConceptScheme
    - Alt_label
    - Definition
    - Hidden_label
    - Notation
    - Pref_label
    - Title
- ContactPoint
    - Email
    - Lang
- Dataset

  - Accrual_periodicity
  - contactPoint
  - Creator
  - Document
  - Editor
  - Equipment
  - Issued
  - Landing_page
  - Page
  - Processing_level
  - Provenance
  - Relation
  - Resolution
  - Service
  - Summary
  - Theme
  - Title
  - User
  - Variable
  - Software
  - Platform
  - Owner
  - Contributor
  - Publisher
  - Author
  - Project
  - Distribution
- Distribution
  - Accessurl
  - Bytesize
  - downloadURL
  - Format
  - Issued
  - Lincese
  - Mimetype
  - Rights
  - Title
- Document
  - Abstract
  - Citation
  - Creator
  - Dataset
  - Issn
  - License
  - Mimetype
  - Pages
  - Published
  - Title

  - Url
  - Volume
  - Software
  - Owner
  - Contributor
  - Publisher
  - Project
  - Distribution
  - Catalog
- Equipment
  - End_date
  - Filter
  - Orientation
  - Period
  - Range
  - Resolution
  - Start_date
  - Serial
  - Theme
- Facility
  - Title
  - Theme
  - Address
- Frequency
- Organization
  - Email
  - Lei_code
  - Phone
  - Url
  - Address
  - Vcard
- Person
  - Affiliation
  - Email
  - familyName
  - givenName
  - Nationality
  - Phone
  - Qualification
  - Url
  - Address
- Platform
- Program
- Project
  - License
  - Title
  - Organization
  - Owner

    - Contributor
    - Partner
    - Responsible
- Spatial
    - Geometry
    - Location
    - Latitude
    - Longitude
    - Max_latitude
    - Min_latitude
    - Max_longitude
    - Min_longitude
    - Reference_system
    - Unit_latitude
    - Unit_longitude
    - Address
- Sensor
- Service
    - Url
    - Terms_of_service
    - Software
    - Owner
    - Provider
    - Contact
    - Project
- Software
    - Abstract
    - Creator
    - downloadURL
    - License
    - Memory_requirements
    - Published
    - Operating_system
    - Storage_requirements
    - Processor_requirements
    - Programming_language
    - Runtime
    - Size
    - Suite
    - Updated
    - Url
    - Organization
    - Owner
    - Contributor
    - Publisher
    - Author
    - Distribution
- SoftwareCode

  - License
  - Programming_language
  - Url
- Temporal
  - End_date
  - Start_date
- Variable
  - Alternate_name
  - Max_value
  - Measurement_technique
  - Min_value
  - propertyID
  - Unit_code
  - Unit_text
  - Url
  - Value
  - Value_reference
- Vcard
  - Country
  - Email
  - Fax
  - Locality
  - Phone
  - Postal_code
  - Region
  - Url
  - Address
- WebSerivce
  - Api
  - License
  - Published
  - Url
- WebPage
  - Abstract
  - Creator
  - License
  - Mimetype
  - Published
  - Title
  - Url
  - Owner
  - Contributor
  - Publisher
  - Project
- Website
  - Abstract
  - Creator
  - License

- Mimetype
- Published
- Title
- Url
- Owner
- Contributor
- Publisher
- Project