



OpenWebSearch.EU

“Piloting a Cooperative Open Web Search Infrastructure to Support Europe’s Digital Sovereignty”

Deliverable D5.1 Launch of the Pilot Infrastructure

Version 1.0

Open Web Search 

The Project is funded by the EC under GA 101070014



Funded by
the European Union



OPENWEBSEARCH.EU

1. Introduction.....	7
1.1 Vision and Goals for the OWS-FDI:	7
2. Infrastructure Architecture Elements	9
2.1 Single Sign On (SSO) feature via B2ACCESS:	11
2.2 Layer 1: Interfaces.....	11
2.3 Layer 2: Authentication and Authorization Infrastructure layer (AAI)	14
2.4 Layer 3: Crawling, Pre-processing, Index Generation.....	14
2.5 Layer 4: Compute infrastructure and Storage	18
3. Implementation Status and Roadmap	21
4. Conclusion and Outlook	26
5. Appendix.....	27

Preliminaries

i. Project Info

Project number	101070014
Project acronym	OWS.eu
Project name	OpenWebSearch.eu – Piloting a Cooperative Open Web Search Infrastructure to Support Europe's Digital Sovereignty
Call	HORIZON-CL4-2021-HUMAN-01
Topic	HORIZON-CL4-2021-HUMAN-01-05
Type of action	HORIZON-RIA
Responsible unit	DG CNECT
Project starting date / Duration	01/09/2022 – 31/08/2025 (36 months)
Project reporting period	1
Project Coordinator	Prof. Dr. Michael Granitzer, University of Passau

ii. Project Partners

Acronym	Partner
UNI PASSAU	University of Passau
BADW-LRZ	Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities
RU	Radboud University
WEBIS	Leipzig University
TUGraz	Graz University of Technology
DLR	German Aerospace Center
IT4I@VSB	Technical University of Ostrava
CERN	Organisation européenne pour la recherche nucléaire
OSF	Open Search Foundation e.V.
A1	A1 Slovenia
CSC	Tieteen tietotekniikan keskus Oy (IT Center for Science)
NLNET	Stichting NLnet
BUW	Bauhaus-Universität Weimar (Associated Partner)
SUMA-EV	SuMa e.V. - Verein für freien Wissenszugang (Associated Partner)

iii. Deliverable Info

Due Date / Delivery Date	30/02/2024 (M18)
Deliverable Lead	CERN
Deliverable type	Report
Dissemination level	PU
Document Status / Version	V1.0
Work-package / Lead Partner	WP5/CERN
Authors	Noor Afshan Fathima, Andreas Wagner, Martin Golasowski, John Truckenbrodt, Katja Mankinen, Stephan Hachinger, Michael Granitzer
Approval	The deliverable expresses the opinion of the authors and has not yet been approved by the EC.
Related Documents	D 1.1 Crawler Coordination Software Stack & Demonstrator V1, https://doi.org/10.5281/zenodo.10355322 D 3.3 The OpenWebSearch Hub and the Open Web Index Y1, https://doi.org/10.5281/zenodo.10369512 D4.2 Report of privacy, transparency, and trust models for search applications V1, https://doi.org/10.5281/zenodo.10369590 Granitzer et al. Impact and development of an Open Web Index for open web search ¹ , https://doi.org/10.1002/asi.24818

iv. Deliverable Summary

This document provides a detailed overview of the first deliverable of Work Package 5 (WP5) entitled "D5.1 Launch of the Pilot Infrastructure", which is part of the OpenWebSearch.eu project. This project is funded by the European Commission under the grant agreement GA 101070014 within the Horizon Europe Framework Program. The "OpenWebSearch Federated Data Infrastructure" (OWS-FDI) spans five data centres, each playing interconnected roles. The document describes in detail the development and current state of implementation, including key metrics. It also defines the relationship of the infrastructure to other work packages that benefit from this setup. The discussion covers the stages from the initial proposal to the current state of implementation and also future plans for the remainder of the project.

¹ Granitzer, M., Voigt, S., Afshan Fathima, N., Golasowski, M., Guetl, C., Hagen, M., Hecking, T., Hendriksen, G., Hiemstra, D., Martinović, J., Mitrović, J., Mlakar, I., Moiras, S., Nussbaumer, A., Öster, P., Potthast, M., Senčar Srdič, M., Sharikadze, M., Slaninová, K., Stein, B., de Vries, A., Vondrak, V., Wagner, A., Zerhoudi, S., 2022. Impact and Development of an Open Web Index for Open Web Search. <https://doi.org/10.1002/asi.24818>.

v. Document Management

History of Changes

Name	Version	Partner	Publication date	Changes
Initial Draft	0.1	CERN	Nov 2023	Draft structure and plan for the deliverable
Draft of Sec 1-6	0.2	CERN	22 Jan 2024	Introduction, Layers description, Implementation, Roadmap, Conclusion and Outlook
Draft after initial review version	0.3	CERN, IT4I, LRZ, DLR	15 Feb 2024	Initial Review within WP5
Update draft	0.4	UP	22 Feb 2024	Update Observability and Custom UI by Michael Granitzer, Crawling Components update
Pre-Review-Version	1.0	CERN	23 Feb	Integrated reviewer feedback

vi. Document Approver(s) and Reviewer(s)

NOTE: All Approvers are required. Records of each approver must be maintained. All reviewers in the list are considered required unless explicitly listed as "Optional".

Name	Role	Action	Date
Andreas Wagner	WP5 Lead	Review	2024-02-28
Martin Potthast	WP2 Lead	Review	2024-02-29
Martin Golasowski	WP5, Task 5.2 Lead	Review	2024-02-29
Michael Granitzer	Project Co-ordinator	Approve	

Executive Summary:

This report provides a comprehensive overview of the architecture, implementation, operation, and management of the "OpenWebSearch federated data infrastructure" (OWS-FDI) developed in Work Package 5 (WP5) as it stands at the 18-month milestone (M18). Our ambitious vision for the OWS-FDI, which serves as the foundational element of OpenWebSearch.eu (OWS.eu), was originally outlined in the project proposal and included Tasks 5.1 to 5.4. The success of both the technical and non-technical work packages depends on the resources and services provided by the various data centre partners.

The document describes:

- The project vision and goals
- The Key architecture elements of the infrastructure, providing insights into their design and operational strategies.
- The joint federated storage
- The current implementation and deployment status, providing the technical details from a *system administration* perspective.
- A forward-looking roadmap for the second half of the project.

During the first half of the project, WP5 was instrumental in developing the initial end-to-end pipeline for the creation of the Open Web Index (OWI). This collaboration included supporting Work Packages 1-3 in the creation of the basic versions of the crawler, crawl frontier, preprocessing and content analysis pipeline, indexer, and aggregator for logs and metrics. All of these components were efficiently deployed on the OWS-FDI and the shared storage set up by WP5. This basic version of the pipeline, which is now available, forms the basis for further iterative improvements in the second half of the project. This will include identifying and fixing operational issues, automation, scaling, and expanding and refining the functionality of each component.

In addition to focusing on the results of WP5, which includes the development of the OWS-FDI, we also report on the work of Work Packages 1-3, whose efforts were central to establishing the now operational technical pipeline of the Open Web Index (including the crawling, pre-processing and indexing steps). By detailing these processes, we aim to increase the transparency of our collective progress and provide a brief insight into the methods and processes we developed together that were critical to our operational success in the first half of the project.

Statistics:

Metrics	Value
Frontier (i.e.URLs to be crawled)	ca. 9.8 billion URLs
Web pages crawled	~ 1.23 billion
No. of different languages	185
Storage	~ 77 TiB

Table 1: Few important metrics and their values

Confidentiality and Endorsement Note:

As this document will be publicly available, it will not contain specific details about our infrastructure hardware components, services, deployment locations, and other related information. This action is being taken to maintain the integrity and security of our systems while providing an overview of our technological capabilities, strategies, and successes to date, as well as the many lessons we have learned during their development.

The existing OWS-FDI architecture is based on long-established principles and practices in the field of distributed computing and storage. The specific applications, frameworks, tools, and platforms listed in Table 4 of the Appendix (Section 5.3) have been selected according to recommended open source best practices. We illustrate their use at a high level. This is not to be understood as an outright endorsement, but as adherence to current open source best practices.

1. Introduction

As we approach the halfway point of the OpenWebSearch.eu (OWS.eu) project, we present an overview of our vision and progress to date on the OpenWebSearch Federated Data Infrastructure (OWS-FDI). This report highlights our strategic approach, the design, implementation, and operational status of the technical architecture, and the ongoing improvements of the OWS-FDI, which is envisioned as the backbone of OWS.eu.

Work Packages 1-3 (WP1-3) focus on the development of a pipeline for web crawling, pre-processing, and indexing on modern heterogeneous architectures to be utilised by applications in WP4. WP5 provides and manages computing and storage resources along with monitoring the usage on which the pipeline runs.

Some infrastructure partners provide access to classical *High-Performance Computing (HPC)* resources and others provide *Infrastructure-as-a-Service-cloud (IaaS-cloud)* resources. The pilot OWS-FDI takes the advantages of convergence between *HPC*, *Cloud computing*, and *Big Data* (crawled data in this case) with user friendly interfaces to achieve its goals of providing a cross-data centre infrastructure for creating an Open Web Index (OWI). Such an infrastructure can conceptually demonstrate how Europe's vast HPC/cloud resources can be utilized with Big Web Data, an infrastructure that goes beyond search, and that also supports web data analysis, AI etc. Hence the efforts of WP5 are collaborative in nature. All search applications built from the web indices created or datasets downloaded to train *Large Language Models (LLMs)* support the development and validation of the OWS-FDI in real-world use cases.

This document offers a succinct overview of each service pertinent to the OWS-FDI, as depicted in Figure 1. It adheres to specific typographical conventions for Abbreviations², Services³, Fundamental Concepts⁴, Technology/Framework/Tools/Applications⁵, enhancing clarity and conciseness throughout. Additionally, the semantics of the figures are briefly clarified in their respective sections, enabling readers to easily correlate the text with the relevant illustrations.

1.1 Vision and Goals for the OWS-FDI:

OWS.eu aims to develop and pilot the core for a European OWI and the foundation of an open and extensible European Open Web Search and Analysis Infrastructure by bringing together strong European players, who jointly define, develop, and pilot an open technological backbone for cooperative web search. The OWS-FDI in operation demonstrates, how search applications and web-based data products can be realized through cooperative crawling, analysis, storing, and indexing of web content.

² Abbreviations are introduced alongside their full forms on first occurrence and compiled in Table 1 in the Appendix (Section 5.1). Subsequent occurrences use only the abbreviations. The abbreviations OWS.eu and OWS are used interchangeably.

³ Services in OWS-FDI are displayed in ***bolded italics*** and detailed individually in Section 3, usage e.g., ***Crawling Queue*** (Section 3.4: Crawling Queue). For repeated occurrences of service names, ***bolded italics*** are applied on the first occurrence or depending on the context.

⁴ Fundamental Concepts are Marked in *italics*, such as **OpenSearch cluster**, *object store*, etc. Each italicized term is included in the Glossary table (Table 3) in the Appendix (Section 5.2) with definitions. *Italics* are used on the first mention of repeated fundamental concepts.

⁵ Technology/Framework/Tools/Applications are identified in **bold**, like **OpenSearch**, **Grafana**, etc. These are catalogued in the Technology/Framework/Tools/Application Information table (Table 4) in the Appendix (Section 5.3). When mentioned multiple times, **bolds** are used only on the first instance. Additionally footnotes provide links to their official websites. Bold is used for highlighting sometimes other than this.

Consequently, the project collects large amounts of web data, primarily text and metadata, on the order of several petabytes. OWS.eu also aims to consider ethical and societal requirements as well as legal constraints set forth by the European legislation.

The use and demand for powerful computing resources is determined by the steep increase of data crawled for processing and indexing, later used for analysis, training, and building search applications. We envision that the total volume of data crawled from the public web will reach the set goal of 1 PB in the second half of the project. The ability to pre-process and index this crawled data, and make it available on the **OWS-Engine-Hub** and **OWS Download** interfaces, will be the key driving factor for the sustainable growth of the Open Web Search initiative. To ensure scalability, we focus on building an infrastructure that scales horizontally. Consequently, we aim to identify bottlenecks for scaling the system further and identify potential solutions for overcoming scalability issues.

We currently utilize 2 cloud and 3 world-class HPC infrastructures with the vision of expanding in the future to a pan-European HPC/cloud infrastructures, with the 3rd parties on-boarding to address the requirements of interdisciplinary users of WP1-4, and future 3rd party stakeholders, currently one of them being the **Open Console**⁶. The federation of resources remain one of our top priorities of WP5 with building industry standard interfaces that in the future will link to other European/3rd party initiatives.

The OWS.eu project and, in extension the OWS-FDI, envisions to give the world a pilot alternative to commercial search infrastructures and this report gives the reader a glimpse of how things have been so far, how they can be and what we want to strive for.

⁶ <https://open-console.eu/index.html>

2. Infrastructure Architecture Elements

Figure 1 illustrates the structured design of the OWS Federated Data Infrastructure (OWS-FDI), which is architected to manage data processing and distribution across distributed computing and storage facilities. The architecture is segmented into five distinct layers for clarity and organization:

- **Layer 1** which are all presented within green boxes are collectively referred to as **Interfaces**.
- **Layer 2**, depicted in a purple box, is dedicated to Authentication and Authorization Infrastructure layer, also known as the **AAI layer**.
- **Layer 3** includes the **Crawling, Pre-processing, and Index Generation layer**, all of which are housed within ochre boxes.
- **Layer 4**, shown in a yellow box, constitutes the **Compute Infrastructure layer** necessary for data processing tasks.
- **The Data Distribution Layer**, visualized in a cyan box (bottom most), handles the dissemination of data across the network.

Supplementing these layers is the

- *Single Sign-On* (SSO) feature, which integrates with nearly all components to streamline user authentication and system access.
- Logs Aggregation and Monitoring Service which overlays on most components.
- The two backend components Crawling Queue and Compute workflow orchestration engine. (explained in the layers they belong to)

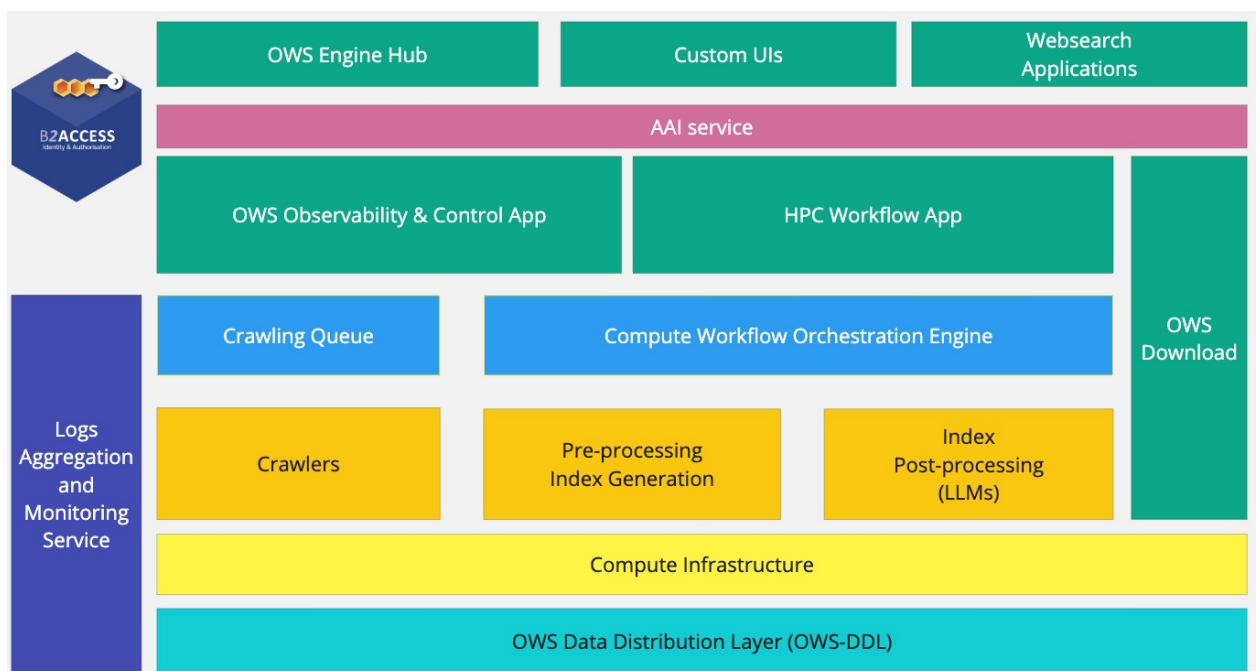


Figure 1: Federated Data Infrastructure

In this section, we describe the components in each of the layers, followed by the implementation & deployment status and road map of each of the components within these layers in Section 3. The word “component” and “service” is used interchangeably to refer to a self-contained unit within the OWS-FDI that performs a specific function. Each service is represented by a box in Figure 1 and contributes to the OWS-FDI’s overall operations. It can be composed of sub-components/sub-services.

The following guidelines will help the reader navigate this section and in extension Section 3 better:

Layered Structure:

Figures 2 to 6, each offering an expanded view of specific services within the comprehensive architecture outlined in Figure 1. These figures are also structured as layers, mirroring the hierarchical design established in Figure 1. Each of these figures provide a detailed breakdown of a particular service, depicts how it interfaces with other external services and also within its own layered hierarchical structure.

Interface with External Services:

When layers or components in these diagrams extend beyond the figure's boundary (consistently the left boundary), it signifies their capacity to interface with other external services, highlighting it's modularity and connectivity.

Common Layers:

The OWS Data Distribution Layer (OWS-DDL) and the Compute Infrastructure Layer are consistently present across all figures, reinforcing their fundamental role in the system's architecture.

Logs Aggregation and Monitoring Service:

This layer, appearing intermittently in Figures 2 to 6, indicates that the service it appears in, interfaces with the **Logs Aggregation and Monitoring Service**, which is an external component.

Authentication and Security Icons:

The inclusion of the B2Access icon, the only hexagonal shaped component instead of a box occurring in all the figures, signifies an *authentication* mechanism protecting the service or software it overlays upon. Similarly, a firewall icon – an illustrated fire and brick wall with a green tick icon, appearing intermittently within figures 2-6, points to secure communication channels among components. When the firewall icon is present on the top right corner of the figure, it is indicative of the instances itself that runs the service is currently secured via IP address restrictions and thus not accessible from outside the participating services or computing centres.

Colour coding consistency:

Figures 2-6 follow the colour coding scheme established in Figure 1. For instance, The **Crawlers** and **Pre-processing & Index Generation** components shown in Figures 4 and 5, respectively, share the same layer and background colour, consistent with their representation in Figure 1.

To provide an eg, this colour correspondence can be understood by comparing the internal layers of Figure 5 with the components of Figure 1: the OWS-DDL layer in both figures is represented with the same colour, indicating they are the same layer. Similarly, the HPC Batch Job Infrastructure layer in Figure 5, marked with a specific colour, corresponds to the Compute Infrastructure layer in Figure 1, indicated by the same colour, and so on and so forth.

Documentation of Services:

Each component, along with its associated automation code and configuration, is thoroughly documented, creating a comprehensive repository for developers and system administrators and is stored in a **GitLab**⁷ instance. This instance is hosted by one of our data centre partners ensuring controlled access and security. Currently, the access to this documentation is restricted and not publicly available, a deliberate measure to mitigate potential security risks that could arise from exposing intricate details of our system's internal workings. Looking forward, we are planning at developing a user-centric documentation. This will enhance the accessibility and usability of our system for 3rd parties, while continuing to safeguard the technical intricacies that are crucial for each internal team's operations.

⁷ <https://about.gitlab.com/>

2.1 Single Sign On (SSO) feature via B2ACCESS:

For the OWS, we needed a solution that provided a SSO mechanism adhering to security standards, and that offered various authentication options like **eduGAIN**⁸ and **ORCID**⁹, particularly suitable for research communities, primarily in the European Union. **EUDAT B2ACCESS**¹⁰, operating within the **EUDAT**¹¹ Collaborative Data Infrastructure (**EUDAT CDI**)¹² ecosystem and managed through an *OpenID Connect (OIDC) Server*, fulfils these requirements. It serves both as an *OpenID Provider (OP)* and a bridge for different *Identity Providers (IdPs)*. B2ACCESS's *proxy model* facilitates *credential translation* and manages *trust and authorization policies* efficiently, including *Transport Layer Security (TLS)*, mutual client authentication for services using *X.509 certificates*.

B2ACCESS authentication is envisioned to be the primary login gateway as depicted in Figure 1, which is positioned at the forefront of both public facing interface components described in Section 2.2 and interfaces aimed at developers, such as the login to individual **iRODS**¹³ *zones*, thereby extending its reach to the entirety of OWS-DDL. This approach ensures not only security but also adherence to legal compliance.

2.2 Layer 1: Interfaces

OWS Engine Hub:

The OWS project yields two significant products: The OWI and the OpenWebSearch Engine Hub (OWSE-HUB). The OWSE-HUB is envisioned as a platform akin to **Docker Hub**¹⁴, but in the context of search engines. It is envisioned as web-based Graphical User Interface (GUI) system that provides a suite of comprehensive search engine stacks. This setup is intended to facilitate the rapid and efficient development of new search verticals. The groundwork for this system will be laid by WP5, which is responsible for establishing the necessary infrastructure. In collaboration, WP3 takes the lead in the actual development of the OWSE-HUB. It is envisioned to access the OWS-DDL as depicted in Figure 1, from where pulling of the *index* with specifications would be facilitated.

Custom UIs:

This interface layer considers special purpose applications like Web Analytic Applications, that use the provided Web data. We aim to support a wide range of use cases beyond search, and very different technological setups. Central for any Custom UI application is to authenticate itself or its users for getting data access. Data access is then facilitated in one of two ways:

1. **Download:** Datasets / Files in Datasets as provided and partitioned by the **HPC Workflow App** (mainly pre-processed and indexed files) can be downloaded and used locally. Download mechanisms are more open to the technology used at the client to store and manage data, but are less flexible in terms of data requested as filtering can be done only on a partition / file-basis. Thus, data filtering needs to be done client side.
2. **Offload:** Via the OWS Observability & Control App layer authenticated users can configure and schedule data push to a set of storage technologies. While the set of the storage technologies is limited, the filtering and data selection can be done beyond individual files. Thus, offloading data

⁸ <https://edugain.org/>

⁹ <https://orcid.org/>

¹⁰ <https://eudat.eu/service-catalogue/b2access>

¹¹ <https://eudat.eu/>

¹² <https://eudat.eu/eudat-cdi>

¹³ <https://irods.org/>

¹⁴ <https://hub.docker.com/>

can be more bandwidth efficient and can support use-cases, where the client has only very limited storage.

Both data provisioning facilities depend on the underlying access layers, the **OWS Observability & Control App** as well as the **HPC Workflow App**. Note that within the project we will not realize every possible application, but offer the options for accessing the underlying data such that third parties will be able to build corresponding applications. This also includes that third parties developing apps bring their own resources and use/enhance the data for their own needs.

Web search applications:

The OWS-FDI is designed to facilitate the creation of diverse search applications utilizing the OWI data. Unlike broad-ranging search engines like Google¹⁵ or Bing¹⁶, vertical search engines focus on specific areas or purposes, offering the possibility to refine search and retrieval methods. WP4 is dedicated to aiding the development of these specialized search engines (search applications) through the OWI, providing multi-level support. Firstly, technical guidance is offered by illustrating the process of building a search application using the OWS-FDI. This involves WP4 supplying technical documentation and a prototype application, which acts as a model for future developments. Secondly, within WP4, two search applications are under development to showcase the practicality and value of this search application concept and the OWS-FDI. This support approach aligns with another primary aim of the OWS project, which is to establish a network of vertical search engines and other related applications. WP5 will deliver the complete infrastructure where both the prototype and the two search applications will be hosted and accessed from.

OWS Observability & Control App:

The OWS Observability & Control App is an application for *monitoring* and controlling the system, providing analytics and operational oversight. It is developed as single-page web application on top of a set of *cloud services*. The application is currently accessible under <https://owler.pads.fim.uni-passau.de> and the functional overview is shown in Figure 2.

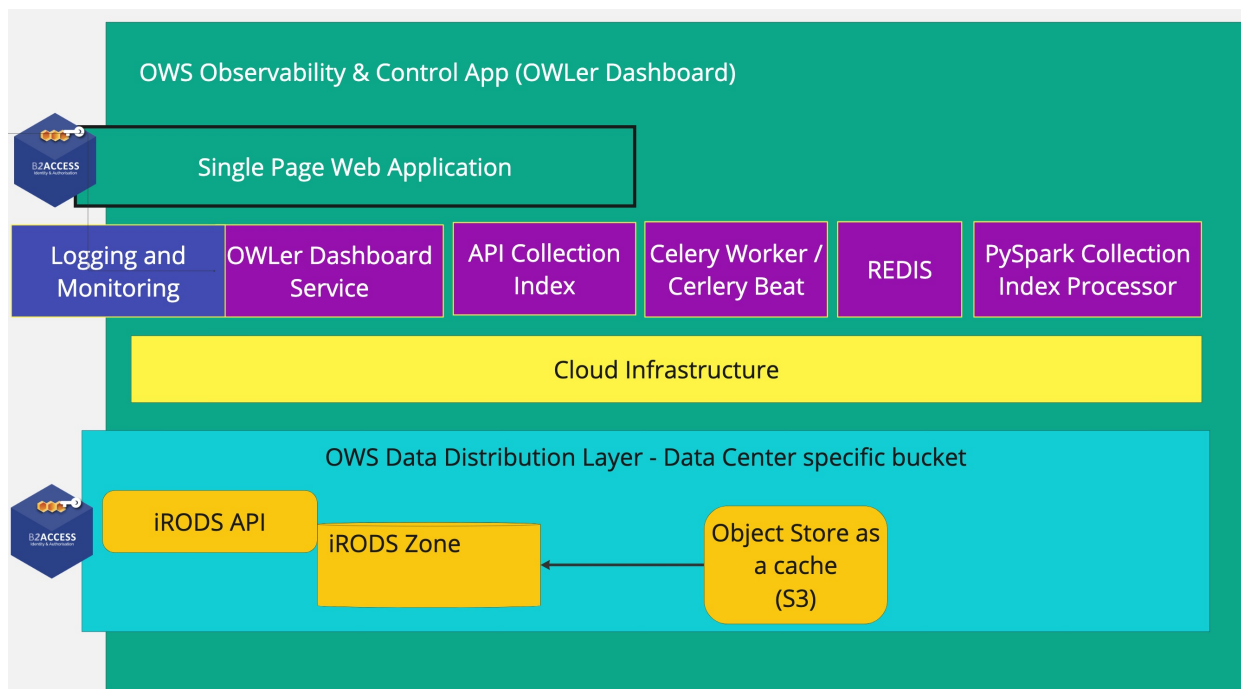


Figure 2: OWS Observability & Control App

¹⁵ <https://www.google.com/>

¹⁶ <https://www.bing.com/>

The single page web application requires B2ACCESS login and covers the following key functionalities:

- **Crawling Metrics & Dashboard:** Observing metrics for the crawling activities (e.g. pages crawled, documents pre-processed, documents indexed) over different temporal resolutions (days, weeks, months) including aggregation statistics for websites and hosts as well as live monitoring of ongoing crawling activities. Data will be analysed from the crawling queue (frontier). For details please refer to the **OWLer Documentation**¹⁷ and “D 1.1 Crawler Coordination Software Stack & Demonstrator”
- **Collection Index API:** Access to statistics for the main index and special sub-indices called *collection indices*. Users can define collection indices as well as “offloading” filters to push data to user-specified storage systems (currently **OpenSearch**¹⁸, **Elasticsearch**¹⁹ and S3).
- **Worker Components** for conducting user-specific indexing and offload tasks utilizing cloud infrastructure at the different data centres. To minimize network costs, the cloud infrastructure should be close to the data, particularly at the different data centres. Thus, every data centre can define a set of *workers* operating on the data. The workers and the API are coordinated via a **(REDIS) Message queue**²⁰, while the functionality is implemented in a **PySpark**²¹ based library called **Collection Index Processor** running at the different workers.

Interfaces to other Components

- Most functionalities require authentication via B2ACCESS
- An API provides programmatic access to the different functionalities and data items
- The App has access to the crawling queue via the metrics server and to the OWS-DDL for gathering statistics and conducting different operations (e.g. offloading of data)

Compute workflow orchestration engine:

This component is the backend of the **HPC Workflow App** as depicted in Figure 1, it streamlines HPC workflows and jobs for the development of the OWI from crawled data, utilizing pre-processing and indexing jobs. Initially, OWS utilized traditional *batch scheduling* systems for resource allocation based on *cluster* priorities. The transition is in place towards fully utilizing the **Compute workflow orchestration engine**, a component of the **LEXIS**²² framework. It handles the challenges of OWS-FDI’s heterogeneity, scale, and geographic distribution. It also interacts efficiently with the batch scheduler, meeting high-level objectives.

This component offers WP2 and WP3 to define workflows, specify execution resources, and ensure seamless integration with **AAI** for security. The management of data, when the crawled data is located differently from computation sites, is also taken care of by interfacing with the OWS-DDL. It abstracts out and manages software provisioning, creation of *virtual machines (VMs)*, *container* uploads, and task assignment across the computing infrastructure (Layer 4). It defines input data, output locations, and processing steps, ensuring efficient task distribution. It collaborates with **HEAppE**²³ *middleware* to provide secure access to HPC/cloud infrastructures.

Public Documentation: <https://opencode.it4i.eu/lexis-platform>

HPC Workflow App:

This GUI facilitates interaction with the **Compute Workflow Orchestration Engine** as depicted in Figure 1, enables efficient management and streamlining of the workflows of WP2 and WP3. They can create a

¹⁷ <https://openwebsearcheu.pages.it4i.eu/wp1/owseu-crawler/owl/er/>

¹⁸ <https://opensearch.org/>

¹⁹ <https://www.elastic.co/>

²⁰ <https://redis.com/>

²¹ <https://spark.apache.org/docs/latest/api/python/index.html>

²² <https://lexis-project.eu/web/>

²³ <https://heappe.eu/>

new workflow execution that calls the corresponding scripts directly, structure these workflows, which are organized and displayed as *Directed Acyclic Graphs (DAGs)*. It also allows users to create, view and modify datasets, including *metadata* handling (e.g. creator or publication year), or file upload, download and deletion within the directory structure of the dataset. Integration of the data-sets with OWS- DDL for data discoverability and replication is also in the same app. The data will be staged to different supercomputing centres. Users are allowed to specify requirements, such as necessary execution resources like GPUs. This component is also from the LEXIS framework.

OWS Download:

This component provides end users with streamlined access to vital datasets produced as the end result of pre-processing and indexing workflows as depicted in Figure 1. The download feature, accessible through the **HPC Workflow App** (LEXIS Portal user interface), enables users to effortlessly download datasets.

The **Dataset Listing**²⁴, a central aspect of OWS Download, offers an overview of available datasets within the OWS-DDL. This includes an extensive range of datasets with their own metadata which are conveniently hosted on the OWS-DDL.

2.3 Layer 2: Authentication and Authorization Infrastructure layer (AAI)

As shown in Figure 1, OWS-FDI constitutes a multifaceted network of services with varied user roles and administrative domains. These services may operate autonomously or in unison, offering user interfaces and backend support for system services, including OWS-DDL. Integration with the **AAI** is crucial for seamless access across different sites by human users and system components. The LEXIS framework's AAI is leveraged for this purpose. Access to all platform services, independent of geographical location, is enabled via SSO through **Keycloak**²⁵, the offered *Identity and Access Management (IAM)* solution. Keycloak operates uniquely yet cooperatively with B2Access, which is included as an IdP within Keycloak. This integration eliminates redundant logins, bolsters security by reducing credential duplication, and simplifies the user experience. It utilizes *OIDC*, *REST interfaces*, and *JWT tokens*²⁶ for service interactions. Access control is managed by *role-based* or *attribute-based* mechanisms (RBAC/ABAC), with Keycloak handling the front-end of authentication, credential management, and session handling. The distributed nature of AAI augments system resilience. HEAppE middleware interfaces with local HPC centre's AAI solutions, providing a crucial integration and separation layer.

2.4 Layer 3: Crawling, Pre-processing, Index Generation

Crawling queue:

This is a crawler component which monitors the status of both crawled and to-be crawled URLs, also managing the order in which web pages are crawled, access statistics and cache digest. This queue termed as "Frontier" is a data structure for storing URLs discovered or visited during a crawl. Figure 3 shows an overview of the individual components, particularly the OpenSearch backend for the several Frontier Apps which then interface via the **URLFrontier API**²⁷ with the peer-to-peer crawling nodes. It can access the OWS-DDL to access different parts of the OWI as well as intermediary files (e.g. transfer logs and public metrics).

²⁴ https://docs.lexis.tech/_pages/architecture/users_view.html#lexis-dataset-listing

²⁵ <https://www.keycloak.org/>

²⁶ <https://jwt.io/>

²⁷ <https://github.com/crawler-commons/url-frontier>

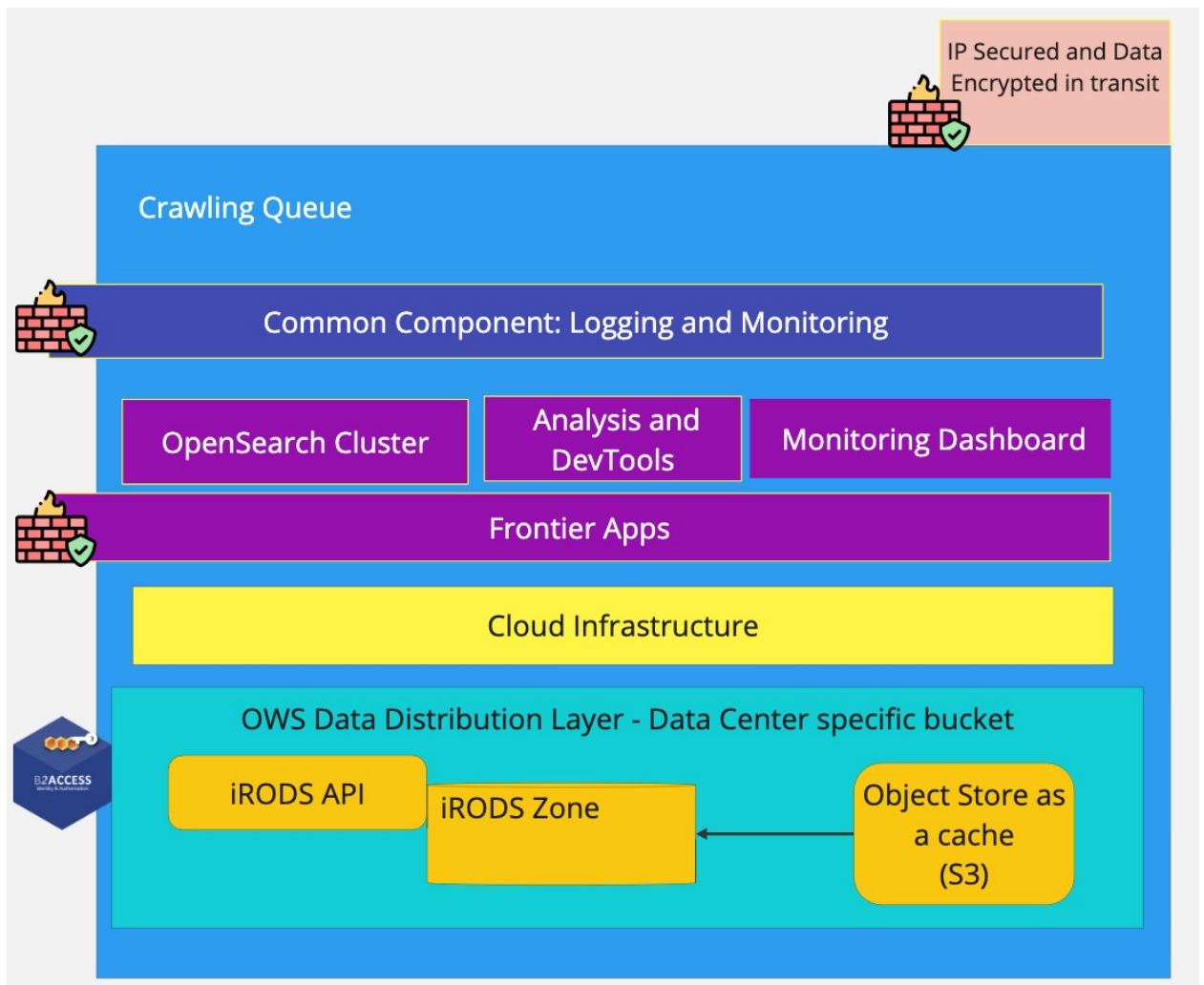


Figure 3: Crawling Queue

The crawling queue consists of the following components as depicted in Figure 3:

- **Frontier Apps** provides an interface to access the frontier queue facilitating a centralized, collaborative crawling mechanism ensuring polite crawling and cross-data centre crawl-priority control. The frontier implementation is based on the **NGI**²⁸-funded Open Source project **URLFrontier**²⁹, which has been upgraded to support dynamic partitioning for diverse crawling needs - all running at one of the data centres. It also connects to the **Logging and Monitoring service**, which aggregates logs from different stages and allows for (limited) statistical analysis, which will be the primary data source for the website registry and other observability and control apps. For more details, please also see the OWLer Documentation³⁰ and “D 1.1 Crawler Coordination Software Stack & Demonstrator”.
- **OpenSearch Cluster** serves as backend for the URLFrontier queues and backs the monitoring of operational metrics.
- **Monitoring Dashboards**: OpenSearch integrates with tools like **Grafana**³¹ which is used to provide real-time visibility into the operational status of the cluster itself

²⁸ <https://www.ngi.no/en/>

²⁹ <https://github.com/crawler-commons/url-frontier>

³⁰ <https://openwebsearcheu.pages.it4i.eu/wp1/owseu-crawler/owler/>

³¹ <https://grafana.com/>

- **Analysis and DevTools: OpenSearch Dashboards**³² is used as the user interface that is set up to visualize OpenSearch data and run and scale the OpenSearch cluster thereby helping development and operations users again.
- All the above services are currently secured via IP address and thus not accessible from outside the participating services / VPN connections to the computing centres.

Crawlers:

Figure 4 shows the components of the **Crawler** services. The crawlers, essentially automated programs or bots, traverse the internet to gather data. This activity generates *WARC files*, which are essentially collections of HTTP data streams derived from the web crawling. These files are then stored in the object store within the OWS-DDL, offering a shared storage solution for other components to access. A sophisticated web crawler system, developed under WP1, is based on the reliable and scalable **StormCrawler**³³ platform, functioning within an **Apache Storm**³⁴ cluster. This system encompasses three main pipelines:

- WARC2WARC, designed for integrating WARC files obtained externally;
- Exploratory Crawling, aimed at regular exploratory web crawling like the name suggests; and
- Sitemap Crawling, which leverages the *Sitemap mechanism*.

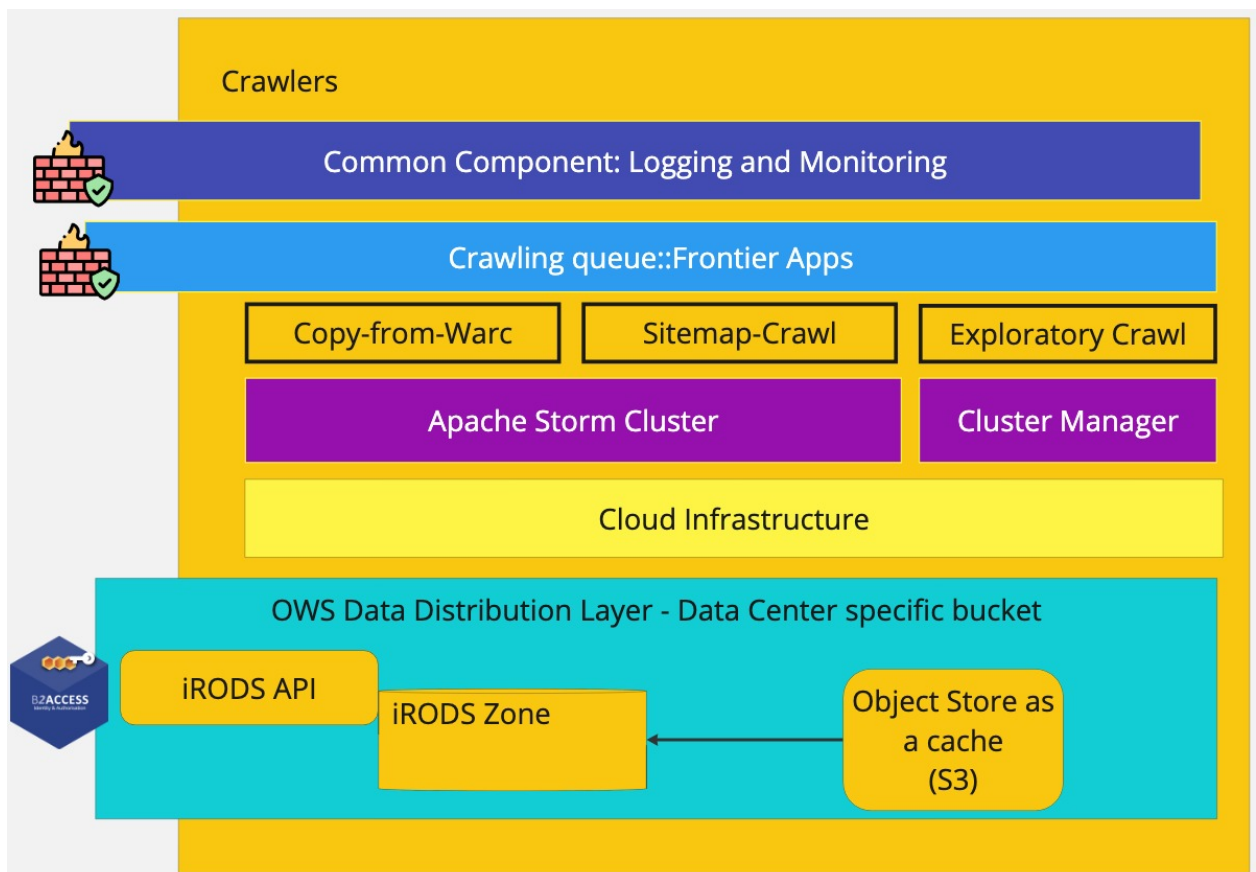


Figure 4: Crawlers

The crawlers run at different data centres and interact with the crawler queue over a Frontier Apps interface and also provide logging and monitoring information to the Logging and Monitoring Component of the crawler queue as depicted in first 4 layers in Figure 4.

³² <https://opensearch.org/docs/latest/dashboards/>

³³ <http://stormcrawler.net/>

³⁴ <https://storm.apache.org/>

For further information, consulting the OWLer Documentation³⁵ and the document “D 1.1 Crawler Coordination Software Stack & Demonstrator” is advised.

Pre-processing & Index generation:

The Pre-processing and Index Generation procedure involves accessing unprocessed data from web crawlers that are stored in OWS-DDL, subjected to a pre-processing stage before being transformed into an indexed format. This process is depicted in Figure 5.

The pre-processing itself is divided into two main activities:

- Pre-processing: This involves enrichment and content analysis. It starts with taking WARC files from the OWS-DDL (providing the shared storage)’ local *object store* which is filled by the **Crawlers**. The step includes extracting cleaned HTML and metadata from these files and storing them in a separate object store under the same OWS-DDL. The metadata from this pre-processing is stored in **Apache Parquet-format**³⁶.
- Pre-processing Plugins Evaluation: This is a unique activity that facilitates the assessment of *plugins* for the content analysis library. These plugins, which can be developed by project members or third parties, enhance the enrichment capabilities during pre-processing.

The pre-processing utilizes **Apache Spark**³⁷ *batch jobs*, employing **Resiliparse**³⁸ for parsing and cleaning HTML content, and applying various metadata enrichments. This work will be elaborated in the upcoming deliverable “Deliverable D2.1 - The OpenWebSearch WARC parsing & content analysis library.” and it is advised to refer it.

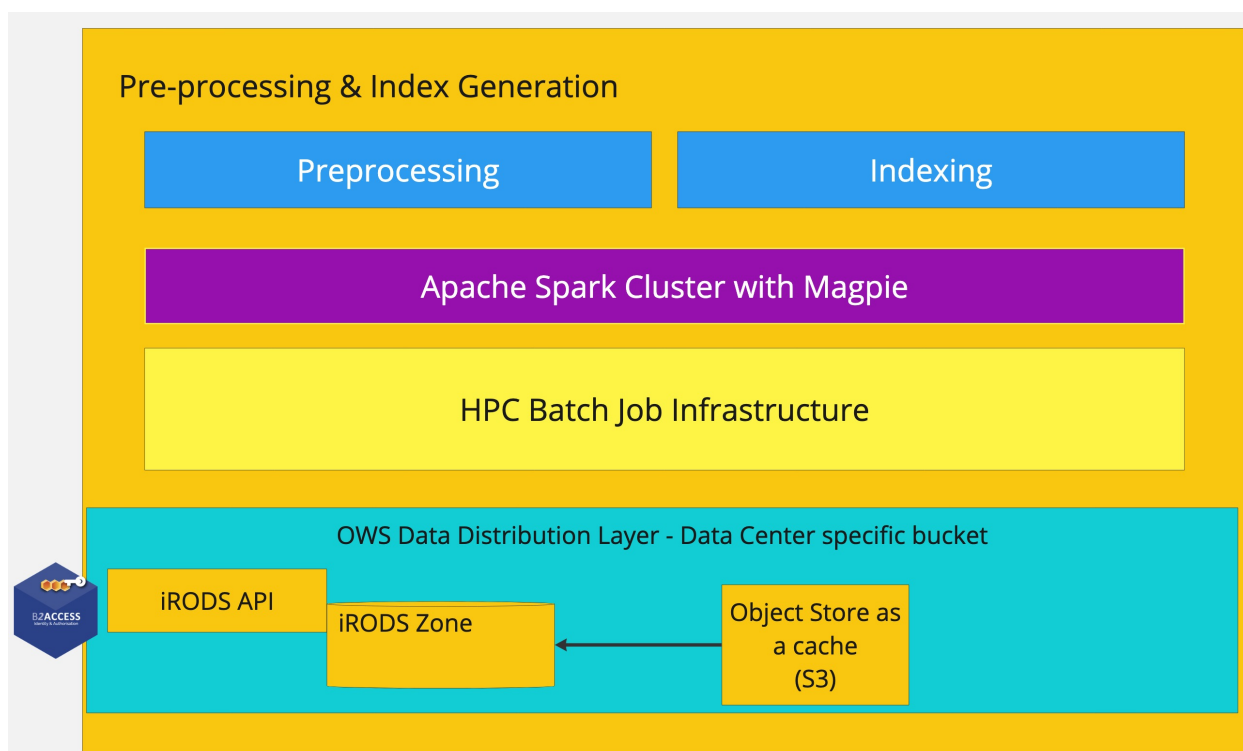


Figure 5: Pre-processing and Index Generation

³⁵ <https://developers.owler.com/>

³⁶ <https://parquet.apache.org/>

³⁷ <https://spark.apache.org/>

³⁸ <https://github.com/chatnoir-eu/chatnoir-resiliparse>

The **Indexing** activity then converts the pre-processed content into a functional index or *inverted file*. These indexes are inverted files, segmented into “*shards*” based on certain metadata types identified from pre-processing, such as topic and language, and distributed as **Common Index File Format (CIFF) files**³⁹.

Like the pre-processing, indexing is executed as an Apache Spark batch job. This approach enables the construction of semantically coherent shards of the entire web index, based on the metadata from pre-processing. These shards facilitate the creation of various search engines, for instance, those targeting specific languages, geographical areas, or topics like news or sports.

Index post-processing (LLMs):

After the index is generated, there is a post-processing step as depicted in Figure 1, which might involve training *Large Language Models (LLMs)* for understanding or categorizing the content. Index post-processing regards products that are built on top of the outputs of the main processing pipeline. This includes, for instance, large language models trained on (a subset of) the cleaned and pre-processed Web content. At the time of writing, no such products exist yet, but we anticipate interesting use cases and ideas to arise when we start publicly sharing the data – both from within the project consortium and from the community.

2.5 Layer 4: Compute infrastructure and Storage

The WP5 partners contribute the vital computing and storage capabilities to the project. Some partners offer traditional High-Performance Computing (HPC) resources, while others supply *Infrastructure-as-a-Service cloud (IaaS-cloud)* solutions.

It leverages the implementation that is based on the federation of large-scale geographically distributed cloud-computing resources and HPCs which are integrated with existing systems including some of the world’s top 20 supercomputers. These are the foundational computing resources that support all operations of OWS-FDI. The HPC/Cloud (compute) infrastructure and the storage layer focuses on the interactions among HPC and Cloud hardware systems to provide the computing power and data storage space to the layers above it in Figure 1. The computing facilities employ *perimeter firewalls* as well as *local firewalls* to implement *defense-in-depth*. The supercomputing centres providing infrastructure for OWS are certified according to *ISO 27001* and are thus managing IT security within a framework compliant to this standard.

The confluence of Cloud and HPC in OWS-FDI:

Some of WP5 partners offer Infrastructure-as-a-Service (IaaS) platforms having their on-premise cloud solutions, enabling on-demand provisioning of high-specification VMs and/or *bare metal machines* (supporting *hypervisors* for VMs) with extensive CPU cores and RAM, which run services of various layers in Figure 1. These are backed by HDD/SSD storage and extensible S3 cloud storage clusters. These platforms are tailored for tasks requiring significant configurability, extended runtimes, or continuous uptime, typical of crawler applications and services in OWS. They can be enhanced by **Kubernetes**⁴⁰ for *container orchestration*. The cloud solutions are based either on **OpenStack**⁴¹ and/or **VMWare**⁴²-based IaaS-cloud that merges the advantages of cloud computing with control over physical infrastructure. This cloud environment ensures scalable application deployment and management. The storage clusters, accessible via VMs, bare metals or containers, are safeguarded by TLS for API communications.

The cloud complements the HPC ecosystem, which some other WP partners offer, thereby providing confluence of cloud and supercomputers, which support the processing of vast crawled data within practical time frames. Security is managed by the respective Cloud/HPC’s internal security departments. All the

³⁹ <https://github.com/osirrc/ciff>

⁴⁰ <https://kubernetes.io/>

⁴¹ <https://www.openstack.org/>

⁴² <https://www.vmware.com/>

services running in the infrastructure and the access to the infrastructure resources itself are currently secured via IP address and thus not accessible from outside the participating services or computing centres.

Logging and Monitoring Service:

This component is the **Log aggregations and Monitoring Service** that is designed to gather logs and metrics from all other components. It interfaces with the **OWS Observability & Control App** as illustrated in Figure 1, ensuring that public log data and metrics are made accessible as intended. Incorporating the concept of publicly displaying logs in the monitoring service, this approach enhances transparency, fosters community engagement, and aids in collaborative troubleshooting. Public logs can be invaluable for open-source projects or public services, as they build trust by showing real-time system performance and updates. Additionally, it serves as an educational resource for users and contributors who wish to understand the system better, potentially leading to more effective community contributions. Furthermore, in compliance with certain regulatory requirements, making logs public is necessary for transparency in operations. Care is taken that these publicly displayed logs are carefully curated to maintain security and privacy.

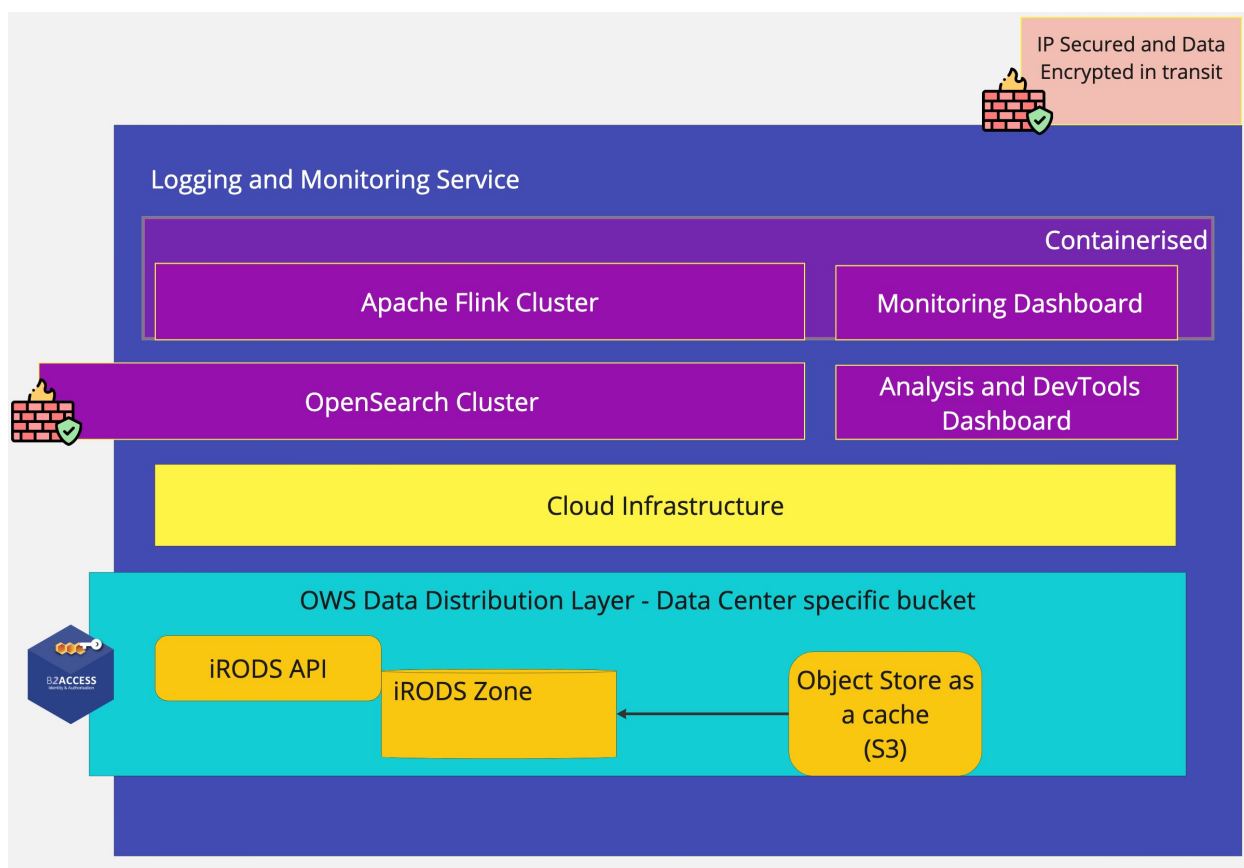


Figure 6: Logging and Monitoring Service

As depicted in Figure 6's upper 2 layers, at the core of this service is an **Apache Flink**⁴³ Cluster, a data streaming platform (which is monitored), operating in conjunction with the OpenSearch Cluster. These layers, in turn can interface with the OWS-DDL from where it can have access to all the logs and metrics from other components. The **OpenSearch Dashboards**⁴⁴, serving as the user interface of **Analysis and DevTools**, facilitates data visualizations of OpenSearch data. It also aids to effectively run and scale the OpenSearch cluster thereby helping development and operations.

⁴³ <https://flink.apache.org/>

⁴⁴ <https://opensearch.org/docs/latest/dashboards/>

OWS Data Distribution Layer (OWS-DDL) :

OWS-FDI utilizes a multi-site architecture for computing and data storage, with the OWS Data Distribution Layer (OWS-DDL) playing a critical role. This layer employs geo-distributed storage and mirroring for data redundancy and safety, though mirroring is not always necessary. It uses the Integrated Rule-Oriented Data System **iRODS** and **EUDAT-B2SAFE**⁴⁵, where each site is represented by an **iRODS zone**⁴⁶ as depicted in Figure 7. This figure depicts that the data storage and management is distributed across three separate data centres, labeled A, B, and C. Each data centre is equipped with S3 compatible local storage, and an associated iRODS Zone, designated as Zone A, Zone B, and Zone C, respectively. The iRODS zones are interconnected, suggesting a networked data management system that allows for data transfer and sharing across the zones.

Further, iRODS manages file-system-like and individual metadata for datasets in an **iCAT**⁴⁷ *metadata catalogue*, with each zone having its own *iCAT server*. This setup allows for a unified view of data across all sites, facilitated by the **AAI** described in Section 3.3 which uses **KeyCloak** for SSO. Access rights on collections and data objects can be controlled via iRODS's built-in mechanisms. Therefore each iRODS zone contributes to the system's availability and unified data view.

The OWS-DDL layer is essential for managing various datasets like raw crawled data, preprocessed data, indices, logs, metrics, indices with specifications etc. It enables secure, location-independent data retrieval and integrates with other OWS-FDI components via REST APIs for data transfer. It aims to provide seamless, cross-site access to aforementioned datasets, making them accessible as if they were in a single file system over a federation of diverse data backend systems.

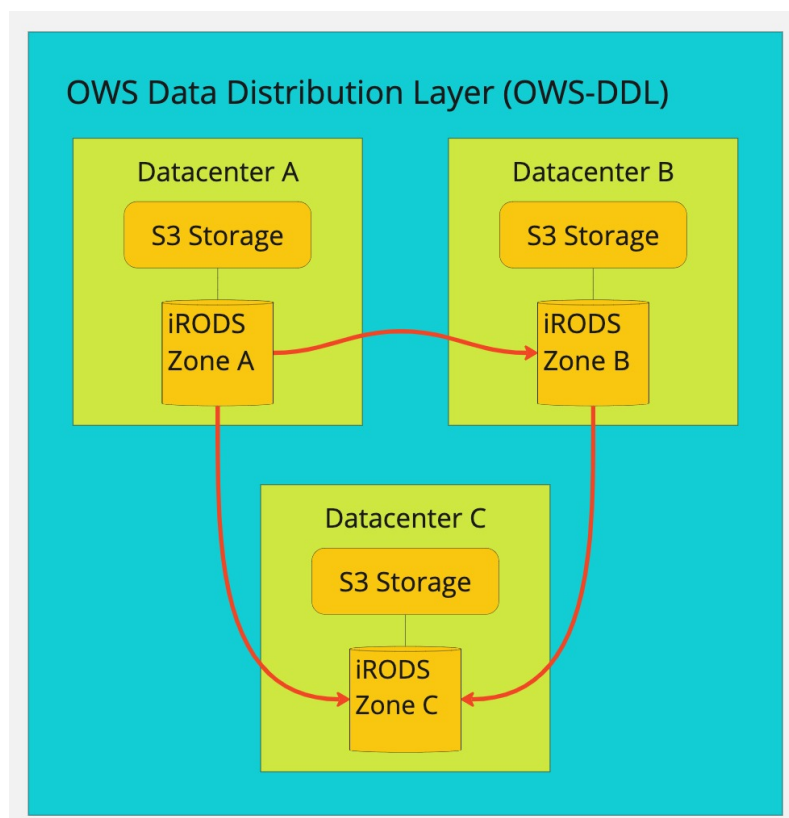


Figure 7: OWS Data Distribution Layer

⁴⁵ <https://www.eudat.eu/b2safe>

⁴⁶ https://docs.lexis.tech/_pages/data_system/irods.html

⁴⁷ <https://icatproject.org/>

3. Implementation Status and Roadmap

WP5 has facilitated the implementation of the loosely coupled distributed design outlined in Figure 1 to encompass operational functions. The approach relies on functional abstraction, where each component provides an interface that conceals its internal workings, allowing for the replacement of components/sub-components and also adaptation without disruption. After the initial implementation and deployment in early months of Y1, a large portion of the service life cycle has included operational tasks like automating, maintenance, updates, and monitoring. Key features enabling these operational tasks include system configuration and backup/restore capabilities, fostering the necessary introspection for debugging, tuning, and maintenance. Recognizing that failure is a common aspect of large-scale systems, the design anticipates and accommodates hardware and software failures, ensuring resilience.

This section provides a detailed update on the implementation and deployment status of each individual component, as well as the strategic roadmap set for the remaining lifetime of the project period. Unlike Section 2, which grouped components by layers, this section will address each component separately for a more granular overview. In general, the overall components are in place and ready to be tested for (i) scaling, (ii) extending it to more data-centres and (iii) to include additional functionalities/features.

On the individual level of individual components we pursue the plans as outlined in the following.

SSO via B2ACCESS:

Currently, B2ACCESS is implemented as follows:

- As the login gateway for the ***OWS Observability & Control App***.
- As one of the IdP within the Keycloak interface of **AAI** which is described in Section 2.3.

At one of the data centre partner sites, the process of integrating B2ACCESS with its iRODS zone is in progress. Over the next year, the remaining sites are expected to adopt the similar approach. Additionally, the upcoming year will see the initiation of efforts to integrate B2ACCESS for authentication across various components within the interface layer, as outlined in Section 2.2 and depicted in Figure 1 signifying the authentication mechanism protecting the services on which it is overlays.

OWSE-HUB:

The initiative for OWSE-HUB, with preliminary steps, such as integrating the indexer and the CIFF standard into **TIRA**⁴⁸ are in place. This early integration provided valuable insights into using the generated indexes from the pipeline in a centralized platform thereby providing a tangible vision for the upcoming year.

In the coming year, WP3 will develop more concrete plans for the OWSE-HUB. Specifically, it will define how search engine specifications are constructed and represented, and WP5 will help how we can set up the platform (or 'hub') for sharing these specifications as depicted in its representation in Figure 1. Our target is to deploy an initial version of the OWSE-HUB by the end of Y2.

Web Search Applications:

The prototype application, along with two use case applications derived from this prototype developed by WP4, are currently hosted on-site at the development partner locations. There are on-going plans to transition these applications to the infrastructure of one or more cloud provider partners. Collaboration with WP4, which focuses on Search Applications and User Experience, is ongoing to devise deployment strategies that utilize the compute and storage infrastructure effectively. This collaboration aims to ensure

⁴⁸ <https://www.tira.io>

that these applications are not only efficiently deployed but also made securely accessible to the public as represented by their position in Figure 1.

OWS Observability & Control App:

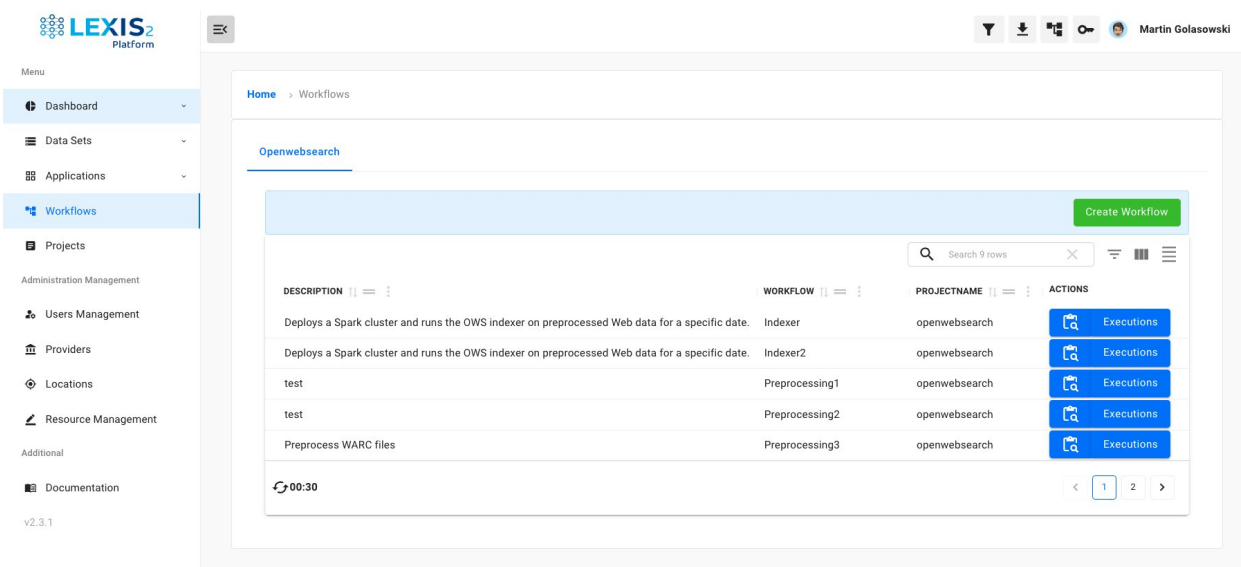
This component comprises of the **OWLER Dashboard** and the **Collection Index API** as depicted in Figure 2, both of which are currently in the early stages of development. WP1 will further improve usability and engage with third parties to utilise the provided data. OWLER Dashboard is expected to be fully deployed and made operational until M 20.

HPC Work-flow App and Compute Workflow Orchestration Engine:

Currently, two key workflows, Pre-processing and Indexing, operate in a sequential manner. Until recently, these workflows involved submitting jobs to a batch scheduler, which then managed the pre-processing and indexing tasks across various compute nodes. Access to these processes was primarily through SSH, connecting to a frontend node of an HPC cluster, with operations conducted via a *Command Line Interface (CLI)*.

This manual approach is being transitioned to a more automated system as depicted in Figure 2, utilizing the **HPC Workflow App** as the frontend interface and the **Compute Workflow Orchestration Engine** as the backend. Therefore, leveraging the advanced execution environment provided by HPC partners within the LEXIS framework, which offers access to supercomputers and cloud resources. This integration significantly enhances the handling and transfer of large data volumes, particularly in connection with OWS-DDL.

In the **HPC Workflow App**, the pre-processing and indexing tasks initiate new workflow executions that trigger the corresponding scripts. [List of the workflows running in the LEXIS Platform is visible in Figure 8.](#) These processes are organized and visualized as DAGs, as seen in Figure 8. For both the workflows, a Apache Spark cluster is configured and deployed within the HPC environment. A Spark job is then submitted to this cluster to index preprocessed data for a specified day (as specified in LEXIS).



The screenshot shows the LEXIS Platform interface. On the left is a navigation menu with items like Dashboard, Data Sets, Applications, Workflows, Projects, and Administration Management. The main content area is titled 'Workflows' and shows a table of workflow executions for 'Openwebsearch'. The table has columns for Description, Workflow, Project Name, and Actions. There are 5 rows of data, each with an 'Executions' button. A 'Create Workflow' button is in the top right. A search bar and pagination controls are also visible.

DESCRIPTION	WORKFLOW	PROJECTNAME	ACTIONS
Deploys a Spark cluster and runs the OWS indexer on preprocessed Web data for a specific date.	Indexer	openwebsearch	Executions
Deploys a Spark cluster and runs the OWS indexer on preprocessed Web data for a specific date.	Indexer2	openwebsearch	Executions
test	Preprocessing1	openwebsearch	Executions
test	Preprocessing2	openwebsearch	Executions
Preprocess WARC files	Preprocessing3	openwebsearch	Executions

Figure 8: List of workflow being executed by OWSeu in the LEXIS Platform

Although these pipelines can be executed from the **HPC Workflow App** and interact with S3 storage currently, integration with the data staging and datasets API provided by LEXIS is still in progress. This integration is a current focus area, aiming to enable WP2 and WP3 to segment their tasks into execution blocks mapped onto the OWS-DDL within the LEXIS framework. Additionally, there are plans to manage

both distributed input/output data and intermediate results through the HPC Workflow App, with B2ACCESS facilitating secure login through **AAI**. The public datasets will therefore be configured for secure download accessibility.

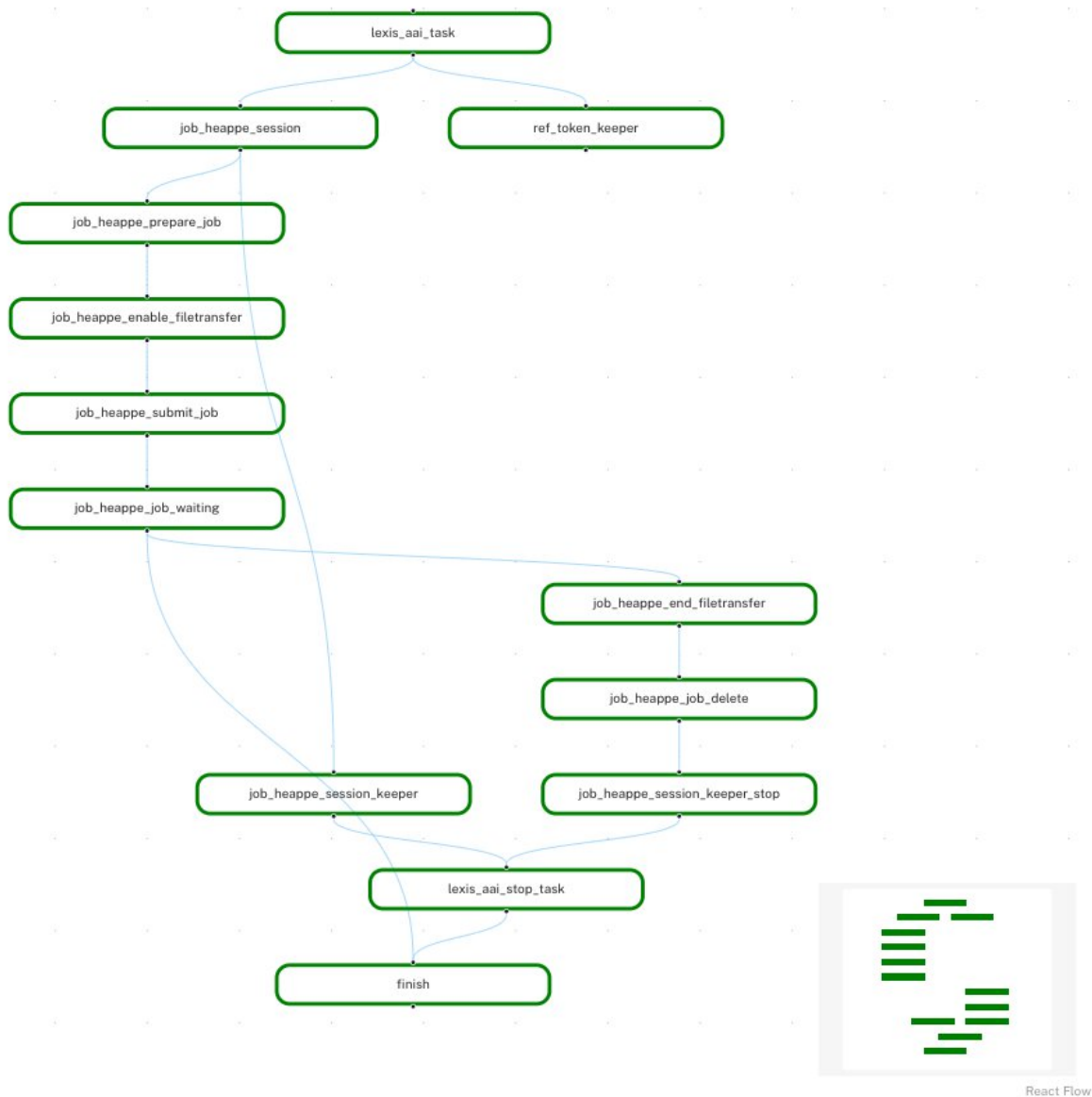


Figure 9: Workflow execution organized and displayed as Directed Acyclic Graphs (DAGs)

Authentication and Authorization Infrastructure (AAI) Layer:

The LEXIS framework integrates the **HPC Workflow App**, **Compute Workflow Orchestration Engine**, and **OWS Download**, optimizing user access through the framework's **AAI** Layer. This integration is depicted within the top 3 layers of Figure 1.

Keycloak, currently functioning as the AAI platform, facilitates the authentication process on the login page. B2Access, integrated as an IdP within Keycloak, enables OWS consortium users to authenticate using their institutional credentials, eliminating the need for multiple service-specific logins. Future enhancements

aim to expand this access to additional OWS-FDI components and OWS-DDL, further streamlining platform integration.

Crawling Queue:

The **Crawling Queue** set-up is implemented according to the description of Section 3.4 and Figure 3, featuring two **Frontier App** instances connected to a backend OpenSearch Cluster on a separate instance. This deployment uses *Logical Volumes* for efficient storage management across physical hard disks. The system's integration with external S3 clusters ensures *data availability* in *alternate zones*, with TLS encryption protecting data in transit. OpenSearch leads the authentication process, maintaining an internal database for user roles and hashed passwords, while partner sites handle *basic-auth credentials*.

The current deployment operates at a partner cloud site, accessible via SSH. Plans include expanding data nodes and implementing a *warm/hot storage mechanism* for better scalability with the Frontier Apps. Memory usage is optimized by adhering to the maximum standard limit per process. Future developments involve connecting to OWS-DDL for frontier and OpenSearch cluster public logs and metrics to be displayed in the main **OWS-O&C App**. Accommodating *horizontal scaling* for additional crawlers and secure WARC file sharing is in the pipeline, along with interfacing with the *open webmaster console* for enhanced management capabilities.

Crawlers:

The crawlers have been implemented in accordance with the description in Section 3.4 referring to Figure 4. They are successfully deployed at 2 partner sites's cloud infrastructure and are in operation at both. The WARC files are stored in individual S3 buckets at both sites for accessibility by other components in the same layer as depicted in Figure 1. At the time of writing, the crawlers crawl peak at roughly 1TB of content per day and in Y2 we aim to scale it up to 2 to 10 TB/day through horizontal scaling and identifying and resolving bottlenecks beyond horizontal scaling.

Pre-processing and Indexing:

These are the pipelines that run currently at the same infrastructure sites as the crawlers but utilizing the HPC infrastructure. The implementation follows the description in Section 3.4 with reference to the Figure 5. For further details on these pipeline themselves it is advised to refer the "Deliverable D2.1 - The OpenWebSearch WARC parsing & content analysis library." and "Deliverable D3.3 The OpenWebSearch Hub and the Open Web Index Y1".

The pre-processing and indexing used to run as **Apache Spark** batch jobs. To deploy a Spark cluster within a HPC batch job allocation and submit Spark applications, the **Magpie**⁴⁹ script collection was being utilised. This has been phased out as described in the implementation section of **HPC Work-flow App** and **Compute Work-flow Orchestration Engine**. An instance of the TIRA platform is currently hosted at one of the data centres.

Current efforts for Y2 are focused on facilitating the scaling up of the pre-processing/enrichment and the indexing so the indexes are built for the content crawled so far and scale in response to the number of WARC files crawled. Functionality will be put in place to estimate the amount of resources required for processing a day's worth of WARC files, such that the pre-processing and indexing can scale along with the amount of content being crawled by the crawlers.

The Log Aggregation and Metrics Service:

The service is implemented according to the illustration in Figure 6 currently running on the cloud infrastructure. Currently it exposes an interface to collect log files from crawler component (Figure 4:

⁴⁹ <https://github.com/LLNL/magpie>

Crawlers) and allows to monitor the different crawler components as described in Section 3.4:Crawlers. Logs and metrics are stored in an OpenSearch index operated at one of the WP5 partners. OpenSearch has been chosen for its robust capabilities in managing large datasets and aggregating log information.

This Service utilises the **Dockerized Flink**⁵⁰ Cluster and Dockerised Grafana is used to monitor this Flink Cluster as depicted in the 1st layer of Figure 6. It currently processes the crawler logs in realtime while reading them from the OpenSearch Cluster (2nd layer of Figure 6), merges them and writes the aggregated logs back into this OpenSearch Cluster. This instance also has the *cron-jobs* for filling the *Blacklist index*. Regular *snapshots* of the OpenSearch clusters are carried out and stored in an external S3 cluster in a different zone.

In Y2 we envision scaling the OpenSearch Cluster and aggregating logs from other components accessible via the OWS-DDL other than only the crawlers. Beyond horizontally scaling the OpenSearch cluster, we aim to identify structural bottlenecks in this setup and identify potential solutions.

OWS Data Distribution Layer (OWS-DDL):

All WP5 partner data centres have successfully deployed iRODS, thereby having the federated OWS-DDL. Each data centre has the capability to push or pull data across the network via iRODS zones.

Each iRODS zone has versatile storage resources available - The Cloud **CEPH**⁵¹ storage cluster available with raw HDD/SDD backed storage via the *POSIX* compatible file system **CephFS**⁵², storage accessible via *NFS*, S3 storage clusters to name a few. At the moment all HPC/data centres in OWS.EU use S3 object storage as a cache, and one of them use it for *disaster recovery* and *business continuity* purposes. Consistent access is ensured via entry points (iRODS servers) at all HPC/Cloud partner sites. Data ingestion, movement, and retrieval within/from the OWS-DDL are exclusively performed via REST APIs in order to ensure sanitized usage patterns and security (restriction of entry points) within the OWS ecosystem.

In the OWS-DDL, data organization follows the structure of *iRODS collections*, which contain data objects and, potentially, sub-collections. These datasets have a universally unique identifier (UUID) and can contain input and output data for a specific workflow. The **py4lexis API**⁵³ is available for download of bulk data.

In Y2, the system will enhance its functionality by leveraging iRODS zones to push or pull data across the network. This approach will for instance enable the centralized collection of logs from all components, streamlining the process by eliminating the need for each component to connect individually to the **Logging and Monitoring Service** (described in Section 3.5:Logging and Monitoring Service)

⁵⁰ <https://flink.apache.org/>

⁵¹ <https://ceph.io/en/>

⁵² <https://docs.ceph.com/en/latest/cephfs/>

⁵³ <https://opencode.it4i.eu/lexis-platform/clients/py4lexis>

4. Conclusion and Outlook

The ambitious goal of crawling 1PB of raw web data in OWS, then pre-processing and indexing it for search applications to be built using these indexes necessitates a sophisticated yet accessible and extensible infrastructure. The combination of Cloud and HPC resources creates a comprehensive infrastructure capable of achieving this goal within the planned timeline. WP5 has managed to design and implement a fully operational Pilot Federated Data infrastructure facilitating the full pipeline — from crawling the Web to pre-processing and indexing — deployed at more than one infrastructure partners. At the end of Feb 2024, the project has indexed approximately 1.23 billion web pages in 185 languages, resulting in a storage of ca. 77 TiB.

For the second half of the project's timeline, WP5 will focus on improving and expanding the current Pilot OWS-FDI. For instance, we will extend deployment of the full pipeline to the other infrastructure partners in the consortium, to make optimal use of the large amount of resources we have available to us. Also, we will measure performance and throughput of the current layers in Figure 1, to evaluate how well they scale and determine where further improvements are necessary.

With the expected adoption of the new governance model being formulated by WP6, substantial resources could be further unlocked depending on the availability to address the interdisciplinary challenges associated with HPC-cloud convergence.

5. Appendix

5.1 List of Acronyms/Abbreviations

Acronym/Abbreviation	Term
AAI	Authentication and Authorization Infrastructure
ABAC	Attribute-Based Access Control
API	Application Programming Interface
CIFF	Common Index File Format
CPU	Central Processing Unit
DAG	Directed Acyclic Graph
DDL	Data Distribution Layer
FDI	Federated Data Infrastructure
FTP	File Transfer Protocol
GUI	Graphical User Interface
HDD	Hard Disk Drive
HPC	High-Performance Computing
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
HTML	HyperText Markup Language
IAM	Identity and Access Management
IdP	Identity Provider
IaaS	Infrastructure as a Service
ISO 27001	International Organization for Standardization 27001
JWT	JSON Web Token
JSON	JavaScript Object Notation
LLM	Large Language Model
LV	Logical Volume
NGI	Next Generation Internet
OIDC	OpenID Connect
OWI	Open Web Index
OWS	Open Web Search
OWS-FDI	Open Web Services - Federated Data Infrastructure
POSIX	Portable Operating System Interface
RAM	Random-Access Memory
RBAC	Role-Based Access Control
REST	REpresentational State Transfer
S3	Simple Storage Service

SQL	Structured Query Language
SSD	Solid State Drive
SSL	Secure Sockets Layer
SSO	Single Sign-On
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UI	User Interface
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
VM	Virtual Machine
WARC	Web ARChive
W3C	World Wide Web Consortium
WP	Work Package
WP1	Work Package 1
WP2	Work Package 2
WP3	Work Package 3
WP5	Work Package 5
WP6	Work Package 6
Y1	Year 1
Y2	Year 2

Table 2 : List of Abbreviations/Acronyms

5.2 Glossary:

Term	Description
Apache Flink Cluster	A framework and distributed processing engine for stateful computations over data streams.
Apache Parquet	A columnar storage file format optimized for use with Apache Hadoop.
Apache Spark	An open-source distributed general-purpose cluster-computing framework.
Application Programming Interface (API)	A set of rules and protocols for building and interacting with software applications.
Attribute-based Access	A security framework where access rights are granted to users through the use of policies that combine attributes, rather than roles, allowing for a more flexible and fine-grained access control.
Authentication	A system that manages user permissions for various services.
Authorization	A system that manages user access for various services.
Bare Metal Machines	Physical computers with no pre-installed operating systems, often used in cloud computing for high performance.
Basic-auth Credentials	A simple way to log in to a website, where you enter a username and password.
Batch Scheduler	A system in computing used to queue jobs for a computer program or system to execute in order.
Batch job	A computer program or set of programs that are executed in sequence without manual intervention, typically processing a large volume of data.
Big data	Extremely large and complex data sets that cannot be effectively managed or processed using traditional data processing applications.
Bottlenecks	Like narrow parts of a bottle that slow down the flow of liquid, these are areas in a process that slow down progress or performance.
Cloud Infrastructure, Cloud Service	The necessary technology equipment and programs, provided by collaborating organizations, located online rather than on personal computers, to help in collecting web data.
Cloud computing	The delivery of computing services—including servers, storage, databases, networking, software, and more—over the internet.
Cluster	A group of servers running a service.
Command Line Interface (CLI)	A text-based user interface used for interacting with software or operating systems, where users issue commands to the program in the form of successive lines of text (command lines).
Compute Infrastructure	The collection of hardware resources like servers and storage devices that power computing tasks.
Compute Workflow Orchestration Engine	A system within the LEXIS framework that coordinates and manages high-performance computing workflows and jobs.
Container	A lightweight, portable, and self-sufficient software package that contains everything needed to run an application.
Container orchestration	The automated arrangement, coordination, and management of computer application containers, ensuring their efficient and reliable deployment, scaling, and operation within a cloud computing environment, often facilitated by systems like Kubernetes.

Collection index	A data structure used in information retrieval systems to optimize and speed up searches by providing quick access to the location of items within a collection or database.
Common Index File Format (CIFF)	A standardized file format used for storing search indexes, where data is organized into 'shards' based on specific metadata, facilitating efficient data retrieval and distribution in large-scale indexing systems.
Crawling, Crawler	The automated process by which a web crawler or spider systematically browses the internet, typically for the purpose of indexing web pages or gathering data.
crawling=>pre-processing=>indexing pipeline	A sequential process in data handling where 'crawling' involves systematically browsing the web to gather data, 'pre-processing' entails preparing and refining this data, and 'indexing' involves organizing the data to facilitate efficient retrieval and analysis.
Crawling Queue	A component in web crawling that manages and prioritizes the sequence of web pages to be crawled.
Credential translation	Information like usernames and passwords that you use to prove who you are when logging into a computer or website.
Dashboard	A visual display of key metrics and data points, often in real-time, providing users with a comprehensive overview of their business or operations.
Data Pre-processing	The process of transforming raw data into an understandable format before it is used for analysis or processing.
Data Streaming Platform	A technology used for processing and analyzing continuous data streams.
Defense in Depth	A multi-layered strategy in cybersecurity involving multiple defensive mechanisms to protect data and information.
Directed Acyclic Graph (DAG)	A mathematical structure consisting of nodes connected by directed edges, where no cycles are formed, commonly used to represent workflows or dependencies in computer science.
EUDAT CDI	European infrastructure for integrated data services and resources.
EUDAT-B2SAFE	A service for storing research data securely in the European Data Infrastructure.
Frontiers Apps	Applications that manage the crawling queue and control web crawling activities.
Geo-Distributed Storage	Storage systems that are distributed across multiple geographic locations.
High Performance Computing (HPC)	Very powerful computers used for solving complex problems.
Horizontal Scaling	Expanding a system's ability to handle more work by adding more computers to the network or more copies of the software online.
Hypervisor	Software, hardware, or firmware that creates and runs virtual machines.
iCAT metadata catalogue	A component of the iRODS system that functions as a centralized database, storing and managing metadata for datasets, and providing a unified view of data across different zones or locations, enhancing data management and accessibility.
iCAT server	A server component of the iRODS system that hosts the iCAT Metadata Catalogue, responsible for managing and providing access to metadata for datasets within an iRODS Zone, facilitating coordinated data management and retrieval across different zones.

Identity and Access Management (IAM)	Systems that manage who can access certain technology and what they can do with it.
Identity Provider (IdP)	A service that keeps and manages digital identities and helps you use these to log into different websites or systems.
Index	A data structure that enables efficient retrieval of information by organizing and storing data based on certain key values or attributes.
Infrastructure-as-a-Service (IaaS)	A form of cloud computing that provides virtualized computing resources over the internet.
Infrastructure-as-a-Service-cloud (IaaS-cloud)	A type of cloud computing service that provides virtualized computing resources over the internet.
Inverted File	A data structure used in a search index that stores a mapping from content, such as words or numbers, to its locations in a database file.
iRODS zones	Distinct, administratively separate instances of iRODS (Integrated Rule-Oriented Data System) that allow for controlled access and management of data within different sections of a larger data grid or system, facilitating secure and compliant data handling.
ISO 27001	An international standard that outlines the best practices for an information security management system (ISMS), providing a systematic approach to managing sensitive company information and ensuring its confidentiality, integrity, and availability.
JSON Web Tokens (JWT)	A secure way to send information between two parties on the internet in a format that's easy to read and verify.
Kubernetes	An open-source platform for automating containerized applications' deployment, scaling, and management.
Large Language Model (LLM)	An advanced artificial intelligence system capable of understanding and generating human-like text across various languages and domains, trained on vast amounts of textual data.
Local firewalls	A local firewall is a software-based security system that controls incoming and outgoing network traffic on a single computer or device based on predetermined security rules.
Log Aggregation and Monitoring Service	A system that collects, aggregates, and monitors logs and metrics from various components.
Logical Volumes (LVs)	A method for dividing a computer's hard disk into flexible, easily-managed sections.
Metadata	Data that provides information about other data, often used to describe the content, quality, condition, origin, and other characteristics of data, especially in the context of large-scale web data collection and analysis.
Middleware	Software that helps different parts of a computer system or application to communicate with each other.
Monitoring, Monitoring Dashboards	Tools used for real-time monitoring and visualization of operational metrics in a system.
Object store	Data storage architecture that manages data as distinct units, or objects, each associated with metadata and a unique identifier, enabling the storage of large volumes of unstructured data.
OpenID Connect (OIDC)	A way to make sure that the person logging into a website is who they say they are, based on checks done by another service.

OpenSearch Cluster	A group of servers running OpenSearch, used for processing and searching large volumes of data.
OpenSearch Dashboards	A visualization tool in the OpenSearch stack for data analysis and visualization.
OpenStack	An open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service.
OWS Data Distribution Layer (OWS-DDL)	A layer in the OWS system for managing data storage and distribution across multiple sites.
Performant	A system whose performance meets or exceeds the design requirements.
Perimeter firewalls	Security systems placed at the boundary of a network that monitor and control incoming and outgoing network traffic based on predetermined security rules, serving as a first line of defense against external threats.
Pipeline	In the context of the Open Web Index (OWI), a pipeline refers to a sequence of data processing steps or stages arranged in a linear workflow for the systematic collection, processing, and indexing of web data.
Platform	In the context of OWS-FDI, a platform refers to a foundational technology or environment that supports and facilitates the integration and operation of various applications, frameworks, and tools in distributed computing and storage.
plugins	Software components that add specific features or functionalities to an existing computer program, allowing for customization and enhancement of the program's capabilities, particularly in content analysis and data processing applications.
Pre-processing	A crucial step in the data pipeline involving the preparation and transformation of raw data, such as web content, into a format suitable for further analysis and processing, like indexing.
Pre-processing Plugins Evaluation	The assessment of plugins for content analysis and enrichment during the data pre-processing stage.
Proxy model	A network architecture component that acts as an intermediary, facilitating credential translation, managing trust and authorization policies, and often implementing security measures like Transport Layer Security (TLS) and client authentication using X.509 certificates.
Py4lexis API	An API used for downloading bulk data within the OWS ecosystem.
PySpark	An interface for Apache Spark in Python, used for large-scale data processing.
Representational State Transfer (REST)	A way of building web services that are easy for computers to use and understand.
Resiliparse	A software tool for parsing and cleaning HTML content.
Role-Based Access Control (RBAC)	A way to decide what someone can do on a computer system or network based on their job or role.
S3	A scalable storage service that offers object storage widely used for storing and retrieving large amounts of data over the internet.
S3 Storage Clusters	Cloud storage resources typically used for storing large amounts of data, accessible via S3.
Sitemap mechanism	A method used in web crawling that utilizes a sitemap, which is an XML file listing the URLs of a site, to inform and guide web crawlers about the structure and content of the website for more efficient indexing.

shards	Segments or portions of a larger database or index, often used in distributed data storage systems, where each shard represents a subset of the data, typically organized based on specific criteria like metadata, to enhance search and retrieval efficiency.
Trust and authorization policies	Guidelines and mechanisms within a network or system that manage and define the level of trust and the permissions granted to users or entities, often involving security protocols such as Transport Layer Security (TLS) and mutual client authentication.
Transport Layer Security (TLS)	A cryptographic protocol designed to provide secure communication over a computer network, ensuring data privacy and integrity between two communicating applications, commonly used in secure web browsing and email.
Virtual machines	Software-based emulations of physical computers that provide the functionality of a physical computer, enabling multiple operating systems and applications to run on a single physical machine, often used in cloud computing for efficient resource management and task distribution.
WARC Files	Web ARChive files, a file format used to store "harvested" web content, including HTML, images, and other media from web crawls.
Worker	In cloud computing, a worker refers to a component or instance responsible for executing specific tasks, such as indexing and data processing, often distributed across various data centres to optimize performance and reduce network costs.
X.509 certificates	Digital certificates that use the X.509 standard to provide a mechanism for mutual authentication, ensuring secure communication by verifying the identity of devices or entities in networked systems, commonly used in TLS/SSL protocols.

Table 3: List of Concepts

5.3 Technology/Framework/Tools/Application Information:

Name	Description
Apache Flink	A stream processing framework for large-scale data processing.
Apache Hadoop	A framework for distributed processing of large data sets across clusters of computers.
Apache Kafka	A distributed event store and stream-processing platform.
Apache Parquet	A columnar storage file format, part of the Apache Hadoop ecosystem.
Apache Spark	A unified analytics engine for large-scale data processing.
Apache Storm	A platform for distributed stream processing.
CEPH	An open-source, distributed storage system that provides interfaces for object, block, and file-level storage, known for its high scalability and reliability, often used in cloud computing environments for versatile data storage solutions like CephFS and S3 storage clusters.
Common Index File Format (CIFF)	A standardized format for storing search indexes, where data is organized into 'shards' based on specific metadata criteria, such as topic and language, facilitating efficient and distributed indexing in large-scale data processing systems.
Docker Hub	A service for sharing and managing Docker container images.
Dockerized Flink	A deployment of Apache Flink, an open-source stream processing framework, contained within Docker containers for improved scalability, isolation, and ease of monitoring, particularly used for real-time processing and management of data like crawler logs.
Dataset Listing	A comprehensive catalog or index within the OWS-DDL that provides an overview and detailed metadata of available datasets, facilitating easy access and exploration of a wide range of data resources hosted on the OWS-DDL platform.
eduGAIN	A service that interconnects identity federations around the world, enabling the secure and seamless sharing of access to services and resources across different institutions and countries, primarily used in the academic and research community for simplified authentication and authorization.
Elasticsearch	A search engine based on the Lucene library, offering distributed, multitenant-capable full-text search engine.
EUDAT	A European initiative that provides integrated data services and resources to support researchers, facilitating secure and efficient data management within the EUDAT Collaborative Data Infrastructure (EUDAT CDI), and employing mechanisms like Single Sign-On and OpenID Connect for streamlined authentication and authorization.
EUDAT-B2SAFE	A data service for replication and safekeeping of research data.
EUDAT B2ACCESS	A service for storing research data securely in the European Data Infrastructure.
EUDAT CDI	European infrastructure for integrated data services and resources.
GitLab	A web-based DevOps lifecycle tool that provides a platform for source code repository management, code reviews, issue tracking, and CI/CD (continuous integration and continuous deployment) pipelines, often used for collaborative software development and version control.
Grafana	Software for real-time monitoring and analytics with dashboard visualization.

HEAppE	Middleware for integration and communication between computing systems.
iCAT	The metadata catalogue.
iRODS	Integrated Rule-Oriented Data System for managing distributed data.
iRODS zone	Basic iRODS deployment is a so-called zone, which has its own set of users and a storage assigned on a local storage array. Each zone maintains its own tree of datasets/collections, their metadata and enforces its access rules.
JWT tokens	Short for JSON Web Tokens, these are compact, URL-safe means of representing claims to be transferred between two parties, used in web authentication to securely transmit information and verify the identity of users in service interactions.
KeyCloak	Identity and access management tool.
Kubernetes	A system for automating deployment, scaling, and management of containerized applications.
LEXIS	Access to High-Performance Computing and cloud-computing infrastructures.
Logstash	A server-side data processing pipeline.
NGI	The Next Generation Internet initiative, a European effort focused on funding and supporting open source projects, research, and innovation to create a more inclusive, transparent, and user-centric internet.
OpenSearch	An open-source search and analytics suite derived from Elasticsearch and Kibana.
OpenSearch Dashboards	The user interface for OpenSearch, used for data visualization and management.
OpenStack	An open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service.
Open Console	An alternative to Google's Search Console.
OpenID Connect (OIDC)	Identity layer on top of the OAuth 2.0 protocol.
ORCID	A non-profit organization that provides a unique and persistent identifier (an ORCID iD) to researchers, enabling them to connect and share their professional information (such as research activities, affiliations, and publications) across various platforms and services.
REDIS	An in-memory data structure store, used as a database, cache, and message broker.
Resiliparse	A tool used for parsing and cleaning HTML content.
StormCrawler	A scalable web crawling framework, designed to be used with Apache Storm.
TIRA Platform	A platform for hosting and managing research software, particularly in the fields of information retrieval and natural language processing.
URLFrontier	Open Source project for managing web crawling queues.
VMWare	A cloud computing and virtualization technology company.

Table 4: List of Technology/Framework/Tools/Application