ENVRI-Hub NEXT

D9.1 Knowledge Base Recommender System Design

Status: Under EC Review

Dissemination Level: Public

Funded by
the European Union

Abstract	
Keywords	ENVRI-KB, Personalized Knowledge Agent, Indexing, Search

The ENVRI-Hub platform is designed to provide seamless access to environmental data and services across multiple European Research Infrastructures (RIs), addressing key challenges in data discovery, access, composition, processing, and usability. The platform enhances collaboration and interoperability by offering tools for automated data processing, federated authentication, and workflow automation while supporting virtual research environments (VREs) to facilitate reproducible research. The ENVRI Knowledge Base is one of the ENVRI Hub key components; it aims to help users manage and retrieve data efficiently, allowing support for advanced search. In this deliverable, we analyse the design requirements for the ENVRI Knowledge Base and its recommendation features based on user stories collected from the stakeholders and propose a microservice-based system architecture, including a User Interface, Knowledge Manager, Knowledge Finder powered by Personalized Knowledge Agent, Knowledge Base API, a Knowledge Storage System, and a Knowledge Base Service Library. The deliverable reviews the technical choices and presents the current development status.

Revision History

Version	**Date**	**Description**	**Author/Reviewer**
V 0.1	04/12/2024	First Draft	Nafis Tanveer Islam (UvA)
V 0.2	10/01/2025	Second Draft	Nafis Tanveer Islam (UvA) Zhiming Zhao (UvA)
V 0.3	13/01/2025	Third Draft	Nafis Tanveer Islam (UvA) Zhiming Zhao (UvA)
V 1.0	**31/01/2025**	**Final**	**Nafis Tanveer Islam (UvA) Zhiming Zhao (UvA)**

Document Description			
D9.1 Design Document			
Work Package Number 9			
Document Type	Deliverable		
Document Status	Under EC Review	**Version**	1.0
Dissemination Level	Public		
Copyright Status	This material by Parties of the ENVRI-Hub NEXT Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead partner	UvA		
Document Link	https://documents.egi.eu/document/4034		
DOI	https://zenodo.org/records/14780043		
Author(s)	• Nafis Tanveer Islam (University of Amsterdam) • Zhiming Zhao (University of Amsterdam)		
Reviewers	• Angeliki Adamaki (ULUND) • Ulrich Bundke (FZJ) • Marta Gutierrez (EGI Foundation) • Alessandro Turco (EPOS ERIC) • Delphine Dobler (Euro-Argo ERIC)		
Moderated by:	• Matteo Agati (EGI Foundation)		
Approved by:	Development Steering Board (DSB)		

Terminology / Acronyms	
Term/Acronym	**Definition**
AAI	Authentication and Authorization Infrastructure
ECV	Essential Climate Variable
ENVRI	Environmental Research Infrastructure
EOSC	European Open Science Cloud
FAIR	Findable, Accessible, Interoperable, Reusable
LLM	Large Language Model
RAG	Retrieval of Augmented Generation
VRE	Virtual Research Environment
KB	Knowledge Base
PKA	Personalized Knowledge Agent
AI	Artificial Intelligence
GDPR	General Data Protection Regulation

Table of Contents

List of Figures

List of Tables

Executive Summary

Researchers face challenges accessing and analyzing environmental data scattered across numerous sources. ENVRI is a collaborative initiative that advances environmental research by integrating and supporting a network of European environmental and Earth system research infrastructures. ENVRI provides a framework for researchers, policymakers, and other stakeholders to efficiently access, share, and use multidisciplinary environmental data and services. The ENVRI-Hub platform is designed to provide seamless access to environmental data and services across multiple European Research Infrastructures (RIs), addressing key challenges in data discovery, access, composition, processing, and usability. Its modular, service-oriented architecture integrates data catalogs, knowledge bases, analytical tools, and training resources, enabling interdisciplinary research and informed policy-making. To enhance ENVRI-Hub to the next level, ENVRI-Hub Next provides improved access to environmental data and services by incorporating next-generation digital solutions and expanding its user base.

The Knowledge Base (KB) is one of the core ENVRI-Hub components, with a structured repository of information designed to store, manage, and retrieve data efficiently. It supports advanced search with recommendation capability, enabling users to find the information stored quickly and efficiently. It is enhanced with AI-based search capabilities, enabling users to efficiently discover and access relevant resources for environmental research and collaboration. We integrate Personalized Knowledge Agents to provide customized AI-enhanced dialogue support for KB users to enhance search and recommendation capabilities.

1. Introduction

ENVRI-Hub is designed to enhance accessibility, interoperability, and scalability across the four environmental domains of the Earth system: Atmosphere, Ocean, Solid Earth, and Biodiversity Ecosystems, which form the ENVRI Community. ENVRI Knowledge Base[1] is one of the key ENVRI-Hub components; it manages and accesses information, enhanced by an intelligent dialogue-based Recommender System. With its modular and scalable design, the recommender system analyses user behavior, search patterns, and contextual cues to deliver relevant insights while enabling seamless integration into various applications.

The ENVRI Knowledge Base (ENVRI-KB) introduces the next generation of search technology by seamlessly integrating Artificial Intelligence (AI) into its core functionalities. It improves upon the monolithic design prototype developed in the previous project ENVRI-FAIR[2], which tightly coupled a user interface, search functionality, and knowledge storage. In the new design, ENVRI-KB offers an innovative dialogue-based search, enabling users to interact dynamically with the system's core intelligence. This conversational approach lets users retrieve information more intuitively and effectively by engaging in natural, question-and-answer-style interactions. Additionally, the platform features a customizable plugin component called Personalized Knowledge Agent, which can be easily integrated into various research infrastructures (RIs). This integration empowers RIs to access higher-quality data and resources while maintaining the flexibility to operate within their existing systems, fostering improved collaboration, data discovery, and research outcomes.

This document outlines the design and development of the new ENVRI-KB, beginning with an analysis of design requirements (Section 2) for various user roles based on user stories to address stakeholders' needs for reproducible research. These user stories are reviewed to derive a comprehensive list of functional and non-functional requirements, which form the foundation for the proposed system architecture. The architecture (Section 3.1) features a microservice-enabled design, including components such as a User Interface, AAI Service, Knowledge Manager, Knowledge Finder with Personalized Knowledge Agent, Knowledge Base API, Knowledge Storage System, and Knowledge Base Service Library, integrating advanced features like a recommender system and dialogue-based search capabilities to support interdisciplinary research. After the architecture, we review the technical choices (Section 4) and present the current implementation status and the development plan (Section 5), and finally conclude in Section 6.

1 "ENVRI Knowledge Base", "Knowledge Base" and ENVRI-KB refer to the same thing.

2 https://zenodo.org/records/10363073

2. Requirement Analysis

Our Requirement Analysis follows an Agile methodology to define the various roles of each stakeholder in the ENVRI-KB. As a key component in ENVRI-Hub, ENVRI-KB may share some stakeholders and requirements from the ENVRI-Hub.

2.1. Methodology

Using Agile requirement engineering practices, we first identified key user roles and conducted interviews with these roles, who were partners in the project. This effort began during the ENVRI-FAIR project, during which we held two workshops with over 30 participants to gather user stories. Subsequently, we refined those user stories and identified both functional and non-functional requirements through consultations with all members of the ENVRI-Hub Next Work Package 9. These user stories may be updated as new features are developed or new technical opportunities arise. For example, our current system supports FR2 and FR3. However, with advancements in technologies like Large Language Models (LLMs), we are currently upgrading our system to incorporate an LLM stack for more personalized recommendations and dialogue-based search. **Table 1** summarizes and provides simplified statistics of the data analysis conducted for the requirement analysis of the ENVRI-KB.

Table 1: Summarization of Requirement Analysis

Requirement Type	Total Count
User Roles	10
User Stories	38
Functional Requirements	39
Non Functional Requirements	18

2.2. User Roles

The user roles for the Knowledge Base largely overlap with those of the ENVRI-Hub platform. The roles define the perspective of stakeholders which were analysed in ENVRI-Hub Next D3.1[3] including various users, researchers, developers, and service providers, and we have identified 10 distinct roles for the Knowledge base as shown in the following list. These roles guide the Knowledge base design and functionality, ensuring that it meets the diverse needs of its user base, including researchers, developers, and service providers.

1. Environmental Researcher;
2. Policy-Making Researcher (focused on environmental management);

3 https://zenodo.org/records/12818775

3. Research Software Engineer (with advanced IT skills);
4. ENVRI Developer (involved in platform development);
5. Service Developer (working on services for the platform);
6. Service Provider (providing datasets and services to ENVRI-Hub);
7. Service Operator (managing catalogue of services and data access);
8. Infrastructure Manager (monitoring infrastructure performance);
9. RI Data Manager (focused on ensuring data compliance with FAIR principles);
10. Knowledge Base (KB) Curator (responsible for managing data quality and indexing).

2.3. User Stories

User stories capture the diverse needs and goals of the key roles interacting with the Knowledge base. Each story is framed from the perspective of different user roles, such as environmental researchers, policy-makers, research software engineers, service providers, and developers. These stories outline specific functionalities and interactions users require to utilize the platform effectively. We have collected 38 user stories in total (Appendix 8.3). During the project development, more user stories might be appended when a new feature needs to be included. The user stories guide the development process, ensuring that the Knowledge base remains flexible, scalable, and responsive together with the other ENVRI-Hub components to the evolving needs of its user base while maintaining adherence to best practices such as Findable, Accessible, Interoperable, and Reusable (FAIR) data principles, security, and interoperability.

Some of those user stories have clear connections with the ENVRI-Hub. For instance, **Story 1** (ENVRI-Hub Platform Access for Environmental Researchers), **Story 4** (Policy-Making Researchers Data Access), **Story 5** (Service Integration for Research Software Engineers), **Story 15** (Authentication and Authorization Management), and **Story 22** (FAIR Data Compliance).
We prioritize a number of ENVRI-KB-specific user stories for requirement analysis, such as **Story 17** (Data Quality and Indexing for Knowledge Base), **Story 25** (Natural Language Search), Story 26 (Personalized Recommendations), and **Story 10** (Data and Metadata Access via APIs). The detailed User Stories can be found in Appendix 8.3.

2.4. Functional Requirements

From the user stories, we identified important ENVRI-KB functional requirements, with consideration of ENVRI-Hub integration. Each functional requirement corresponds to specific user stories, ensuring the ENVRI-KB delivers a user-friendly interface, with advanced search capabilities that can be integrated with the data access mechanisms, workflow automation tools, and compliance with FAIR principles and Open Science practices. Based on the user stories, we identified 39 Functional Requirements for the ENVRI-KB, as shown in **Appendix 8.1**.

Among the 39 functional requirements, we prioritize the following functional requirements in our current KB development stage which is highlighted in Section 5. The system will provide a web-based user interface for discovering ENVRI assets (FR1), Natural Language-based semantic search (FR2), and Personalized Recommendations (FR3). Both will lead us towards the integration of a dialogue-based search system where the end-users can interact using natural language. Seamless integration of ENVRI-Hub data into research workflows will be facilitated by tools like Jupyter Notebooks **(**FR5**,** FR9**,** and FR25**)**. The system will follow the Community (ENVRI and EOSC) best practices of the AAI (FR10**,** FR16**,** and FR24) to manage the users of the Knowledge base. The Knowledge base service will be operated following the CI/CD and DevOps practices (FR4).

2.5. Non-Functional Requirements

The non-functional requirements focus on critical aspects such as scalability, availability, performance, security, interoperability, and maintainability. In addition, they address the platform's compliance with accessibility standards, international best practices, and regulations like GDPR. We identified 18 Non-functional requirements of the ENVRI-KB in **Appendix 8.2**.
Among those non-functional requirements, we highlight the following in the initial development phase. Scalability, ensuring it can handle increasing data volumes and users without performance degradation (NFR1), Availability, guaranteeing continuous access to datasets, services, and tools with minimal downtime (NFR2), Performance, providing fast responses to data queries and service requests, ensuring low latency and efficient workflows (NFR3), and Security, emphasizing robust measures such as encryption, user authentication, role-based authorization to protect data integrity and confidentiality (NFR4), and Interoperability, ensuring seamless cross-domain data access and integration with various Research Infrastructures using standardized protocols like REST, SPARQL, and APIs (NFR5).

2.6. Ethics Requirements

The **ENVRI-KB** is heavily committed to adhering to the highest standards of ethics and data protection, as outlined in EU regulations, specifically Article 14, Annex 5 of the Grant Agreement (GA). The Ethics and Data Protection Board is the advisory body responsible for monitoring compliance with these regulations, ensuring that all processes align with legal and ethical guidelines. These principles are integrated into the development of the ENVRI-KB at every stage of its design and implementation. Built on a foundation of user-centric design, the system analyses the platform's usability through diverse user roles, including researchers, policy-makers, engineers, and developers. These roles guide the creation of user stories, emphasizing seamless data access, collaboration, reproducible research, and skill development while ensuring adherence to the FAIR principles (Findability, Accessibility, Interoperability, and Reusability). These principles are essential for fostering open science and maintaining interoperability across research infrastructures.

Ethical considerations are embedded in defining the functional and non-functional requirements of the Knowledge Base Recommender System. Robust security measures, such as user authentication, role-based authorization, and data encryption, safeguard the integrity and confidentiality of user data. At the same time, the system prioritizes equitable access to information and respects user privacy by implementing stringent data protection protocols. In alignment with EU ethical standards, the Ethics and Data Protection Board continuously monitors the development of the ENVRI-KB to address potential ethical challenges proactively. This oversight ensures that the platform evolves to remain flexible, scalable, and responsive to the needs of its diverse user base, serving as a trusted and ethically sound resource for environmental research and collaboration. Furthermore, a thorough per-service analysis will be performed in agreement with the ethical assessment pending approval by the Ethics Board.

3. Architecture Design

Based on the requirements, we improved the ENVRI-KB architecture using a microservice design pattern. In this section, we will explain the architecture, and map how each component corresponds to the Functional Requirements we defined in Section 2.4.

3.1. Architecture

An important drawback of the original monolithic implementation of the KB system in ENVRI FAIR [R1] is its limited scalability when handling heavy concurrent user access. To overcome this limitation, we propose a new microservice-based architecture, providing modularity, scalability, and ease of maintenance. In this architecture, each microservice is responsible for a specific function, allowing the system to be scaled, extended, and adapted to meet evolving requirements. **Figure 1** depicts the conceptual design of the architecture.

The key microservices include the Knowledge Manager (KB-Mnger), Knowledge Finder (KB-Finder) with Personalized Knowledge Agent for dialogue-based knowledge retrieval, Knowledge Storage, Knowledge Base Services Library (KB-ServiceLib), User Interface (KB-UI), AAI Service and Knowledge Base API (KB-API).

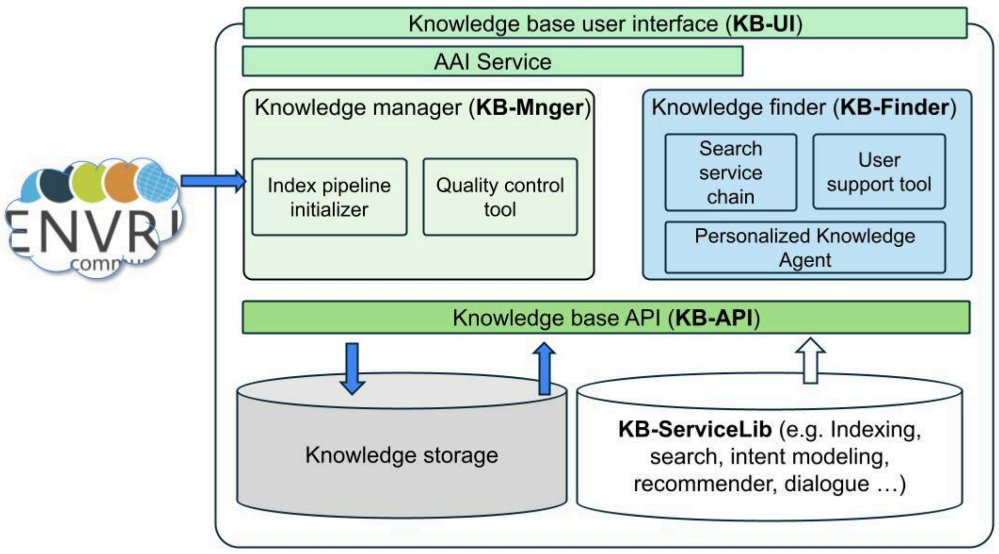

Knowledge base user interface (KB-UI)
AAI Service
Knowledge manager (KB-Mnger)
Index pipeline initializer
Quality control tool
Knowledge finder (KB-Finder)
Search service chain
User support tool
Personalized Knowledge Agent
Knowledge base API (KB-API)
Knowledge storage
KB-ServiceLib (e.g. Indexing, search, intent modeling, recommender, dialogue …)
ENVRI comm

Figure 1: Architecture of different Components of the ENVRI- KB

1. **Knowledge Base User Interfaces (KB-UI)**
   The Knowledge Base (KB) provides flexible user interfaces for various user groups. The KB User Interface (KB-UI) is a web-based platform supporting Platform Access (FR1) and natural language-based dialogue (FR2). The Knowledge Base Access and Management will be implemented using a login system with ENVRI-ID that enables users

to search for information about datasets, notebooks, APIs, and web resources that are indexed from information provided by the RIs or found in the web resources. Furthermore, KB-UI supports recommendations based on the user's search history. While the KB supports searching without any authentication, an authentication is required to become a privileged user of the KB.

2. **AAI Service**
   KB will ensure secure access to the system via the user authentication and authorization services using ENVRI-ID. It handles user registration, login, and role-based access control, ensuring only authorized users can access sensitive data and functionalities. The service integrates with third-party authentication systems like Google OAuth, and the AAI of ENVRI-Hub to streamline login processes and enhance security. Enforcing access control across all microservices, including the User Interface, Knowledge Finder, and KB-API, guarantees that the system's sensitive data is well-protected.

3. **Knowledge Manager (KB-Mnger)**
   The Knowledge Manager (KB-Mnger) is responsible for ensuring that all data processed and indexed in the KB meets the highest standards of quality, thus supporting Data Catalogue indexing Management (FR17) and index quality control (FR32). The KB-Mnger is composed of several indexing pipelines, each designed to handle contents including dataset metadata and API provided by RIs, notebooks, and web contents related to the RIs, ensuring comprehensive data integration and accessibility. It consists of two parts namely Index Pipeline Initializer and Quality Control Tool.

   Index Pipeline Initializer: The Index Pipeline Initializer is responsible for managing the KB's indexing pipelines, which include:
   a. **Dataset Indexer:** This component processes structured metadata from datasets (provided by the ENVRIs at this stage), extracting and structuring it into our standard schema by storing relevant information to make it searchable. While each metadata has its own format (key, value pair), we formalized a standardized and structured set of keys that would support the RIs metadata. Currently, our system supports 23 key pair values when we index dataset metadata;
   b. **API Indexer:** The API Indexer focuses on acquiring and indexing data from external (to the Hub) APIs, converting dynamic and real-time data into structured indexes. The API indexing is done based on the set of API endpoints provided by the RIs. The API endpoints resolve to either a JSON or a CSV file which we index in our KB;
   c. **Notebook Indexer:** This component extracts and indexes content from computational notebooks, such as Jupyter Notebooks, ensuring that both code and research outputs are captured for reproducibility. Currently, we index information including code and functional descriptions for each cell in a notebook from GitHub and Kaggle Repositories;
   d. **Web Indexer:** This component extracts components from the websites of all the RI's ensuring all the contents from their website are indexed.

Quality Control Tool: A critical aspect of the Knowledge Manager is the Quality Control Tool, which ensures that only high-quality data is indexed in the KB. The tool evaluates data based on several quality parameters, including Timeliness, Popularity, Accuracy and Completeness, Consistency and Reproducibility, and Executability. For example, when we index content from the API endpoints or contents from the web, some of them produce broken links or provide incomplete information. We do not index these into our KB, ensuring data correctness. However, the contents from the API endpoints may get updated in the future. Therefore, when a new indexing is requested by an RI, the new contents are always indexed or updated, ensuring timeliness, accuracy, and consistency of the contents. By maintaining these quality standards, the Quality Control Tool ensures the integrity of the KB's indexed assets, making it a reliable resource for researchers and service providers. The details on the quality parameters are explained below:

- *Timeliness*: Ensures data is up-to-date by evaluating publication dates, release dates, and modification dates. This aligns with FR13, ensuring data is findable and accessible;
- *Popularity*: Measures data relevance through metrics such as download frequency and citation count, ensuring that widely used data is prioritized;
- *Accuracy and Completeness:* Ensures that data is correct and all necessary data elements are present. This functionality supports FR5, ensuring that only high-quality and complete data is made available to users;
- *Consistency and Reproducibility:* Verifies data reliability and ensures that it can be consistently reproduced, directly supporting FR9;
- *Executability:* Confirms that data can be successfully run in its intended environment, ensuring compatibility and performance.

4. Knowledge Finder (KB-Finder)

The Knowledge Finder (KB-Finder) is one of the core components responsible for enabling discovery within the KB. It is designed to help users find relevant datasets, APIs, and research outputs by leveraging advanced search techniques. The KB-Finder directly supports Web-based search (FR1) and Personalized Recommendations (FR3) by offering a powerful search and recommendation system. Furthermore, the KB-Finder has one key component, the Personalized Knowledge Agent, that supports dialogue-based knowledge retrieval with advanced recommendation techniques. The User Support Tool provides personalized recommendations based on the user's search history and behavior, and the Search Service Chain, a key sub-component of the KB-Finder, employs semantic search and intent modeling to enhance the accuracy and relevance of search results.

Personalized Knowledge Agent (PKA): The Personalized Knowledge Agent is an extendable part of the system that supports dialogue-based natural language search with an advanced recommendation system based on user preference. This system allows users to use the ENVRI-KB to search the system using dialogue-based queries. The system will respond with appropriate responses, and the user can investigate the returned response further with more natural language-based queries. Furthermore, the

RIs can use the Personalized Knowledge Agent Frontend Plugin in their system for pop-up recommendations and dialogue-based queries. Some of the subcomponents of the personalized knowledge agent are:

- **Dialogue-Based Search:** The dialogue-based search system architecture combines continuous learning and fact-verified retrieval to deliver accurate and reliable user interactions. At its core, the system utilizes Large Language Models (LLMs) to interpret queries and generate responses, supplemented by a Continuous Learning module with a Low-Rank Adaptation (LoRA) [R2] technique. This goal is to fine-tune a small subset of parameters while freezing the original weights, allowing efficient incorporation of new knowledge without full retraining of the model, thereby reducing computational costs. To address hallucination generated by LLMs, our implementation includes a Retrieval-Augmented Generation (RAG) [R3] module that integrates a vector database to store domain-specific documents and their vector representations. During query processing, the RAG module retrieves verified context from the vector database, enabling the LLM to ground its responses in factual, trusted data;

- **Recommender System:** The dialogue-based recommendation is an advanced feature of the recommender system that leverages conversational interfaces to guide users through complex queries. The recommender system allows general recommendations on search results, prioritizing frequently accessed or highly relevant data based on search history. Moreover, if the configuration is turned on, the recommender system can show a popup window with recommendations based on a user's query;

- **Personalized Knowledge Agent Frontend Plugin:** The plugin system is a pluggable system designed to easily integrate into the systems of various RIs using plugin code for the user interface. The user interface is an HTML-based frontend plugin with JavaScript services that connects to the dialogue-based search and recommender system using APIs for smooth integration into the KB system. The lightweight HTML frontend plugin ensures user-friendly access to popup recommendations and dialogue-based interactions with minimalistic setups.

5. Knowledge Storage

The Knowledge Storage component serves as the primary repository for the indexed data in the KB. It employs data storage using indexing for efficient search capabilities, ensuring that large KB data can be quickly searched and retrieved. This component supports FR1: Platform Access and FR9: API Access by ensuring fast and reliable KB data retrieval. In addition, the KB uses a relational database to store user interactions, search histories, and feedback from the community. Therefore, "Terms of Usage" will be provided during production to comply with GDPR. This data is used to improve search capabilities and provide more personalized recommendations, supporting FR3.

6. **KB-API**
   The Knowledge Base API (KB-API) enables internal (KB-UI) and external (to the KB) applications and developers to interact with the KB programmatically. It exposes core functions such as KB data retrieval, indexing, and quality control through API calls, making it easy for external systems to the KB to integrate with the Knowledge Base. Since indexing is a privileged access, it has to be authenticated using AAI service before starting the indexing process (Section 3.3, Fig 2, and 3). The KB-API is key to the extensibility of the system, allowing other platforms and applications to leverage KB functionalities for broader adoption and collaboration within the research community.

7. **Knowledge Base Services Library (KB-ServiceLib)**
   The KB-ServiceLib is a service library with modules for indexing, AI models, and machine-learning techniques that power the KB's search, recommendation, and dialogue capabilities. This library service will be used as a building block for the search system, the dialogue-based system, or the recommendation system, which is used to build the different components of the KB-Finder, including the Search Service Chain and the Personalized Knowledge Agent**.** This library is designed to be modular, allowing new models to be added or existing models to be replaced as more accurate and high-performing alternatives become available. This flexibility aligns with FR2 and FR3, ensuring that the KB remains up-to-date with the latest advancements in AI and ML.

3.2. Requirement Mapping

The following section presents the mapping between the components of our architecture and the functional requirements. The mapping links KB components to functional requirements and illustrates how the KB system will deliver key features. Table 2 demonstrates the mapping between each component of the KB architecture and which key Functional Requirements (FR) they fulfill.

Table 2: Requirement Validation: Mapping Functional Requirements to the respective components from the KB architecture described in section 3.1

Component Name	Key Functional Requirements (FRs)
KB-UI	FR1, FR5, FR6, FR9
AAI Service	FR13, FR19, FR 27
KB-Mnger	FR1, FR6, FR8, FR9, FR12, FR16.
KB-Finder (including PKA)	FR2, FR10, FR19, FR20, FR3
Knowledge Storage	FR1, FR9, FR16
KB-ServiceLib	FR6

3.3. How does the System Work?

Summarizing 3.1, the ENVRI Knowledge Base system is designed as a modular and integrated architecture where each component collaborates with the others to deliver advanced data discovery, management, and interaction functionalities. The **Knowledge Base User Interface (KB-UI)** provides an accessible web platform with an authentication service using AAI that caters to diverse user groups, enabling them to search for datasets, retrieve recommendations, and manage metadata. This interface is supported by the **Knowledge Finder (KB-Finder)**, which uses semantic search and intent modeling to interpret and refine user queries, ensuring accurate and context-aware search results. Additionally, the **Personalized Knowledge Agent** extends the **KB-Finder**'s ability by enabling dialogue-based natural language queries and advanced recommendation systems, enhancing user interaction with conversational capabilities and personalized suggestions.

Behind the scenes, the **Knowledge Manager (KB-Mnger)** ensures that data ingestion and indexing are standardized and quality-controlled through its Index Pipeline Initializer and Quality Control Tool. The indexed data is stored in the **Knowledge Storage**, for efficient retrieval and a relational database for managing user interactions and search histories, thereby improving search personalization. The **Knowledge Base API (KB-API)** acts as a bridge for external (to the KB) applications and systems, allowing programmatic access to core functionalities like KB data retrieval and indexing using AAI Service.

The workflow diagram in Figures 2 and 3 illustrates the interaction and data flow among the various components of the ENVRI-KB system, covering user and content management and support of other search services. Figure 2 demonstrates the login procedure for the three user roles: basic user, superuser, and content manager. The details on the different levels of user types are discussed in Section 5.2. In the current implementation, the content manager needs to be approved by the superuser to run the index pipeline. However, we are working with the RIs to decide the integration and automation of this process with AAI Service. Figure 3 shows the use cases of the KB-Finder with the Personalized Knowledge Agent (PKA). This includes a configurable popup dialogue initiative that presents recommended knowledge to the user. Moreover, the user can also initiate a conversation with the PKA, which would retrieve and rank information from the KB and present it to the user.

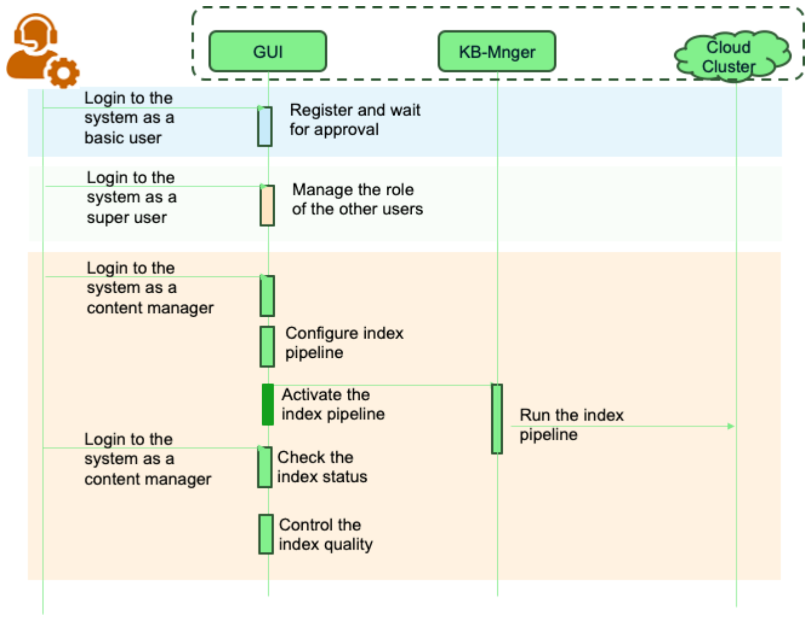

GUI
KB-Mnger
Cloud Cluster
Login to the system as a basic user
Register and wait for approval
Login to the system as a super user
Manage the role of the other users
Login to the system as a content manager
Configure index pipeline
Activate the index pipeline
Run the index pipeline
Login to the system as a content manager
Check the index status
Control the index quality

Figure 2: User and Content Management in ENVRI-KB

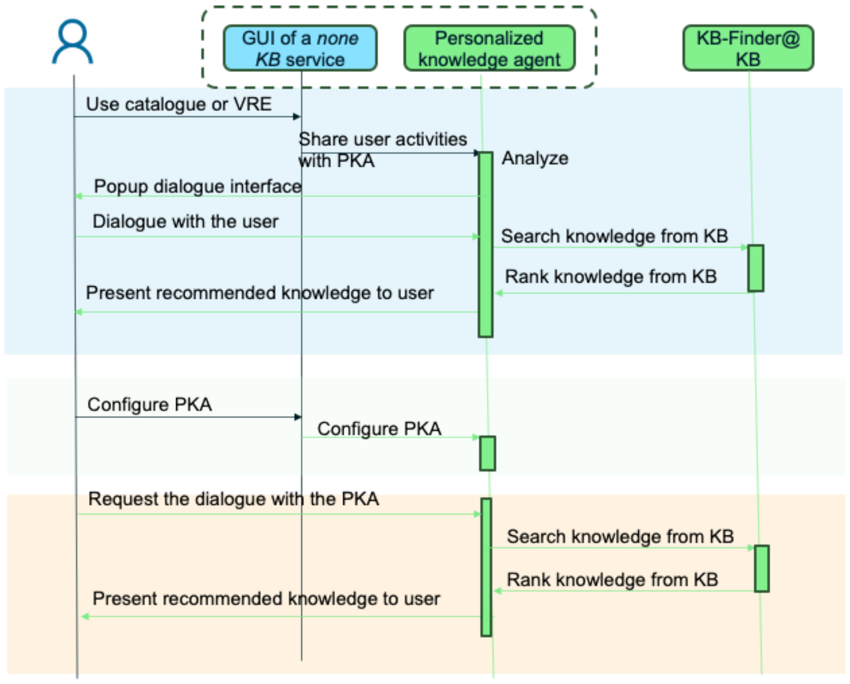


Figure 3: Support other Search Services Using the Personalized Knowledge Agent

4. Technology Choices

We reviewed different technology choices for each of the components of ENVRI-KB and selected the suitable technology. Table 3 shows the technological candidates and the technology chosen for each component.

Table 3: This table provides clear references for the technological options considered for each component, ensuring transparency in the decision-making process for selecting the most appropriate technologies.

Component Name	List of Technological Candidates	Selected Technology
Knowledge Manager (KB-Mnger)	1. **Apache Airflow**: Widely used for data workflows (Apache Airflow Documentation) 2. **Celery**: Distributed task queue (Celery Documentation) 3. **Prefect**: Modern orchestration (Prefect Docs) 4. **Kubernetes CronJobs**: Native K8s scheduling (Kubernetes CronJobs Docs)	**Celery**: Chosen for its lightweight task scheduling and distributed execution, ideal for managing indexing pipelines.
Knowledge Finder (KB-Finder) with PKA	1. **Elasticsearch**: Search engine (Elastic Docs) 2. **Solr**: Open-source search (Solr Docs) **OpenSearch**: AWS-supported search (OpenSearch Docs) 3. **LLM-based tools**: Natural language processing (OpenAI GPT Docs)	**Elasticsearch + LLM-based tools**: Elasticsearch ensures efficient search indexing, while GPT-based tools enable semantic query understanding. These will be used as a package from the KB-ServiceLib.
Knowledge Base API (KB-API)	1. **Django:** Highly scalable framework (Django Documentation) 2. **Flask**: Lightweight API framework (Flask Documentation) 3. **FastAPI**: High-performance framework (FastAPI Docs) 4. **Node.js**: Event-driven APIs (Node.js Docs)	**Django**: Selected for a full-stack framework and robust data management with web application alongside API.

	5. **Spring Boot**: Enterprise-grade APIs (Spring Boot Docs)	
Knowledge Base Service Library (KB-ServiceLib)	1. **Scikit-learn**: Machine learning (Scikit-learn Docs) 2. **TensorFlow**: Deep learning (TensorFlow Docs) 3. **PyTorch**: Deep learning (PyTorch Docs) 4. **Hugging Face Models**: NLP models (Hugging Face Docs) 5. **OpenAI GPT APIs**: Natural language tasks (OpenAI GPT Docs) 6. **Retrieval of Augmented Generation** (RAG Docs) 7. **OpenAI LLM** (OpenAI Docs)	**RAG with Hugging Face Models**: Selected for their pre-trained models and APIs, enabling efficient implementation of dialogue and recommendation systems. Also, Huggingface models are fully customizable, trainable, and free to run on any GPU system, while OpenAI charges for each inference.

The current system is being developed in the Python Django framework. When indexing the contents, to run the pipeline managed by KB-Mnger periodically based on a schedule, we select Celery which allows lightweight task scheduling and distributed execution. The KB-Finder uses a combination of ElasticSearch and LLM-based tools for basic search and dialogue-based search with recommendations. The KB-Finder is designed to retrieve content from the web, metadata from ENVRI dataset resources, APIs provided by the RIs, and notebooks from GitHub and Kaggle. KB-API enables external applications and developers to interact with the KB programmatically. Therefore, a full-stack framework must be integrated with the rest of the system. Therefore, we use Django for this implementation. KB-ServiceLib is a complete library package of the dialogue system, recommender system, indexing system, and intent modeling. The KB-ServiceLib aims to provide a package of the individual sub-systems so they can be used as building blocks to build the KB system. For KB-ServiceLib, we use HuggingFace LLMs for dialogue capabilities. HuggingFace LLMs have easy access and training capabilities, and further, we used RAG to ensure the contents generated by the LLM follow facts.

5. Current Status and Implementation Plan

In the following subsections, we provide the details of each implemented system component, including KB-UI, KB-Mnger, and KB-Finder, with implementational details on the Personalized Knowledge Agent. We also demonstrate some of the results of the indexing of KB-Mnger. Then we discuss the CI/CD Pipeline and the implementation plan.

5.1. KB-UI

The current prototype of KB-UI is deployed on the local infrastructure of UvA. KB-UI in **Figure 1** shows the user interface of the system. The **KB-UI** is responsible for managing user interactions and providing the interface through which content managers can configure workflows, schedule indexing tasks, and monitor the system. It is built using Bootstrap for responsive design and Django for backend integration, following the Model-Template-View (MTV)[4] pattern. The user interface of the PKA is currently a work in progress. In the following section, we will explain various components of the KB-UI of our system:

Via the **KB-UI,** the **ENVRI-KB** super user can manage different users**.** In **Section 2.2** different user roles are defined for each of the key stakeholders of the platform. To simplify the management of different user roles in the current KB system, we group them into three categories for the ENVRI-KB backend. However, after the completion of the discussion with the RIs, this grouping of the users may be updated. Furthermore, during registration of each type of user group, they need to agree to the "Terms of Agreement", as discussed in **3.1** under Knowledge Storage. The three categories of users are explained below:

1. **Superuser:** The Superuser has the highest priority among all of the users. The superuser has the ability to approve or deny the request of a to-be content manager and also manage content. Furthermore, a superuser can view all the content managers and downgrade a content manager to a basic user. The number of superusers is currently one and may be increased based on the request or feedback from the RIs.
2. **Content Manager:** A content manager is a privileged user who can request daily, weekly, monthly, and yearly content management updates. The update request will be placed in a queue and will be executed based on the given frequency. The content manager typically refers to the Knowledge Base (KB) Curator who is responsible for managing KB data quality and indexing, ensuring compliance with FAIR principles, and providing datasets and services to any of the ENVRI-Hub sub-systems.
3. **Basic User:** A basic user has no privileges except to search and view content from the knowledge base. A basic user can request from the superuser to become a content manager. Without login, they can only search content from the KB.

Each user group has a set of features available to them, some available once the user is logged in, authenticated, and authorised to get access to the respective (and more advanced) features. The **Profile** feature lets authorised users change their profile information, including email

[4] https://www.javatpoint.com/django-mvt

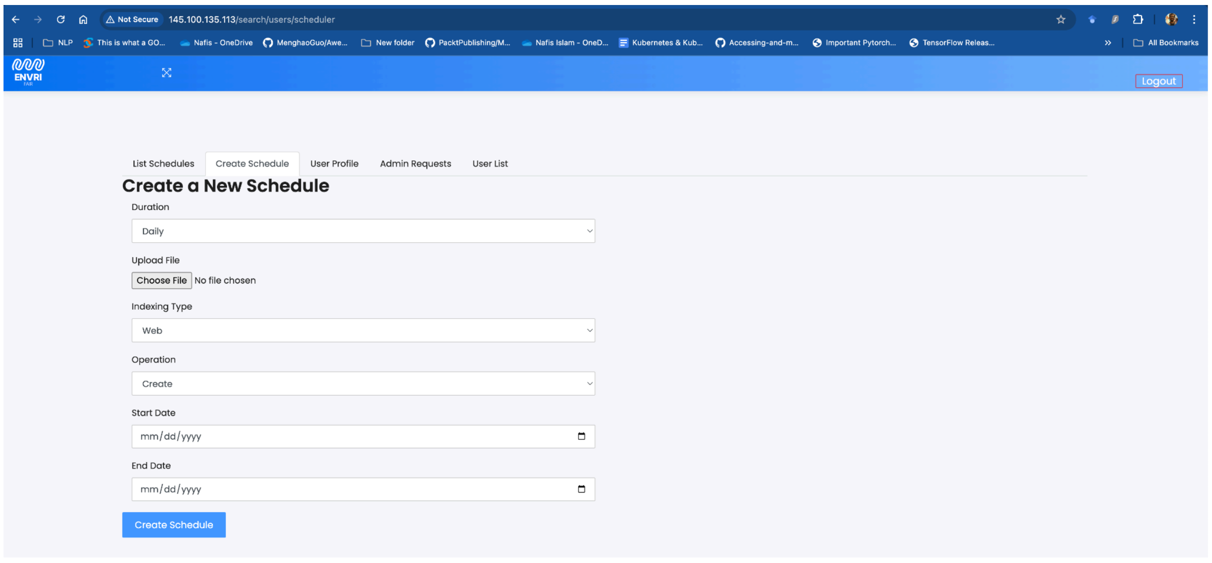
address, password, etc. The **Content Schedule** allows the content managers and superusers to request and schedule the indexing of new content, and while indexing new content, the search service remains fully operational. The **Schedule Modify** allows the content managers to modify or delete any existing schedule. **Approve CM** feature enables the superuser to approve or deny a user's request to become a Content Manager (CM). Finally, the **Decommission CM** feature allows a superuser to remove a content manager and revert to being a basic user. Table 4 shows the feature availability list of the 3 user groups we implemented (Superuser, Content Manager, and Basic User).

Table 4: Feature List Comparison of Different levels of Users

Features	Basic User	Content Manager(CM)	Superuser
Search	☑	☑	☑
View	☑	☑	☑
Profile	☑	☑	☑
Content Schedule	☐	☑	☑
Schedule Modify	☐	☑	☑
Approve CM	☐	☐	☑
View Users	☐	☐	☑
Decommission CM	☐	☐	☑

5.2. KB-Mnger

KB-Mnger allows the content managers to configure scheduling indexing workflows for scheduled indexing through a dedicated UI, updating parameters and defining workflows for different asset types. Figure 4 shows that content managers and superusers can use it to ingest content for a particular RI. Configurations are stored in a MySQL database for easy access. Currently, the backbone structure of the scheduled indexing is developed. Since this system is connected to the AAI service implementation, this will be fully developed when we have a complete understanding of the RIs for authentication.



List Schedules | Create Schedule | User Profile | Admin Requests | User List

Create a New Schedule

Duration	Daily
Upload File	Choose File No file chosen
Indexing Type	Web
Operation	Create
Start Date	mm/dd/yyyy
End Date	mm/dd/yyyy

Create Schedule

Figure 4: Scheduling Interface for the Content Managers

New Content Ingestion: In addition to ingesting content primarily from RI resources that provide datasets, notebooks, APIs, and relevant websites, we have refined our ingestion and indexing procedures. One key improvement addresses a limitation in our previous notebook indexing process in ENVRI FAIR, which only included metadata but did not include the source code itself—a critical component for Research Infrastructures (RIs). We now extract and index the source code directly from notebooks to address this. However, processing notebook code presents unique challenges due to extracting and interpreting various code blocks. For Jupiter notebooks, we extract code from each cell along with its corresponding header, which typically describes the purpose of the code block or cell. We intentionally omit outputs and other non-essential cells, such as those containing images or supplementary information. For standalone Python code files, we separate the code from the comments and index code and comment individually, ensuring a more comprehensive and structured representation of the source code for indexing and retrieval. These updates significantly enhance the usability and reproducibility of code for RIs.

5.3. KB-Finder

The KB-Finder is responsible for retrieving information from the Knowledge Base. The KB-Finder currently uses the indexed content from the knowledge base implemented using ElasticSearch and then retrieves it. We use the BM25 algorithm in order to rank the contents retrieved.

The **Personalized Knowledge Agent (PKA)** is an intelligent system that enhances responses using the Large Language Model (LLM) by understanding user intent and verifying facts. It retrieves past conversations to provide more context-aware responses. To ensure a factual response, PKA uses Retrieval Augmented Generation (RAG) to pull fact-verified information from a vector database we build, ensuring responses do not deviate from the information we indexed in the Knowledge Storage from the RIs. It leverages historical user search data stored in our MySQL database for recommendation-based queries and ranks results based on relevance and user preferences. To stay updated with new facts, PKA continuously fine-tunes itself using Low-Rank Adaptation (LoRA), efficiently integrating new knowledge without requiring complete model retraining.

5.4. Service: CI/CD Pipeline

The Knowledge Base is deployed in the Local UvA/LifeWatch (prototype development) and LIP infrastructure (staging, production environment) following modern DevOps practices for the CI/CD pipeline. The entire Knowledge Base is deployed on a 3-node cluster managed by Kubernetes. The components that make up the Knowledge Base are containerized using Docker and deployed on the cluster cited below:

- Elasticsearch stores indexed documents. We deployed one instance of this service.
- The KB-UI serves web pages and gets results by calling the KB-Finder. We deployed one instance of this service.
- The KB-Finder searches the content from ElasticSearch and returns to the KB-UI. We deployed two instances of this service.
- The KB-Mnger indexes the contents into ElasticSearch. We also deployed two instances of this service to the Kubernetes cluster.

For the KB-UI, KB-Mnger, and KB-Finder, new images are built from the source and published in Docker Hub using GitHub actions. This cloud deployment provides better flexibility in handling variable loads. For instance, the Elasticsearch database and search interface can be scaled dynamically to respond to peaks in demand. It also improves data resilience through duplication across two geographic locations and service availability. In failure, traffic can be redirected to the backup cluster through the load balancer. Because data and services on the backup cluster are kept ready, downtime is limited to a few minutes. Finally, the whole service infrastructure and configuration can easily be recreated in a new infrastructure by deploying the Helm Charts.

5.5. Implementation Plan

The KB system components KB-UI, KB-Mnger, and KB-Finder are implemented and will be delivered on month 12, and the Personalized Knowledge Agent of KB-Finder AAI Service are work in progress and will be delivered based on the plan we described in **Table 5**. Here, WIP refers to Work In Progress and Comp. refers to complete. Our internal development is accessible online (**https://dev.envri-search.lab.uvalight.net/search/**).

Table 5: Implementation Status of our System

Component Name	Completed Parts	Remaining Parts	Status	Month XX
LIP Site Integration	Site Link: **https://search.staging.envri.eu/search/**		Comp.	Month 12

KB-UI	KB-UI currently handles search, display results, user profiles of different levels of users, and interaction between different user levels.	Working on creating the User interface for the PKA.	WIP	Month 14
KB-Mnger	Separate API, implemented as a POST service in Django Python such that the content managers can update or reindex their content.	Ingestion of New Content given by the RIs.	Comp.	Month 14
KB-Finder	Search API to retrieve content from the Knowledge storage (Elastic Search), with PKA.	PKA Implementation ongoing.	WIP	Month 16
Knowledge Storage	Containerized implementation of the Elasticsearch.		Comp.	
AAI Service	Initial meeting with RIs.	Waiting to hear from RIs to start the development of the authentication System.	WIP	Month 16
KB-Service Lib	Library version of dialogue-based agent and recommender system.	Library version of dialogue-based agent and recommender system.	WIP	Month 16
Version 1 Release		Deliver the version 1 Knowledge base software and service.		Month 18

6. Summary

In this deliverable, we analyse the system requirements from user stories collected from different stakeholders and improve the previous monolithic design as a microservice. In the new design, we propose a Personalized Knowledge Agent to deliver customized support to different user needs, and the PKA can be integrated into different systems or RIs to allow their users to get dialogue-based searches and recommendations. We reviewed the State-of-the-Art AI technologies for developing search recommendations and personalized knowledge agents. Development is currently in progress. We plan to deliver the final version of the software by **Month 18** (July 31, 2025).

7. References

Reference	
No	**Description/Link**
R1	**Knowledge sharing and discovery across heterogeneous research infrastructures. Open Research Europe**, 2023 Jun 6;1:68 Farshidi S, Liao X, Li N, Goldfarb D, Magagna B, Stocker M, Jeffery K, Thijsse P, Pichot C, Petzold A, Zhao Z.
R2	**Low-rank adaptation of large language models**, arXiv preprint arXiv:2106.09685, 2021 Jun 17 Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. Lora
R3	**Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems**, 2020;33:9459-74. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S.

8. Appendix

8.1. Functional Requirements

FR1: Web Interface for Data Discovery

The system shall provide a user-friendly web interface for environmental researchers to search and discover relevant data and services.

FR2: Natural Language Semantic Search

The system shall support natural language search capabilities, enabling researchers to find relevant datasets, APIs, or computational notebooks quickly.

FR3: Personalized Recommendations

The system shall provide personalized recommendations to researchers based on their past searches and activity.

FR4: Continuous operation of the Knowledge base service

The system shall provide tools to workflows, streamlining repetitive tasks focusing on the continuous operation of the Knowledge base, and maintaining the quality of the service and user experiences.

FR5: Jupyter Notebooks Integration

The system shall provide Jupyter Notebooks and other tools for integrating ENVRI-Hub data into research workflows.

FR6: Knowledge Base and Training Resources Access

The system shall offer access to a knowledge base containing training resources and documentation for skill development.

FR7: Data Customization and Integration Tools

The system shall allow researchers to customize and adapt available software tools and data services to fit specific research needs.

FR8: Policy-Maker Data Access

The system shall enable policy-making researchers to access relevant environmental data (e.g., water quality and species distribution) for decision-making.

FR9: API Access for Service Integration

The system shall offer APIs, SPARQL endpoints, and other programming interfaces for research software engineers to integrate ENVRI data into workflows and experiments.

FR10: Training and Capacity Building for Developers

The system shall provide training materials and capacity-building tools for ENVRI developers to improve skills relevant to platform development.

FR11: Cross-Domain Data Search

The system shall enable cross-domain searches for researchers and service developers through the ENVRI-Hub Catalogue of Services.

FR12: Seamless VRE Integration

The system shall allow researchers using VREs like Jupyter or Galaxy to seamlessly import data assets from the ENVRI-Hub Catalogue into their VRE systems.

FR13: Single-Sign-On (SSO) Access

The system shall support Single-Sign-On (SSO) and external identity providers (e.g., ORCID) to enable users to access multiple ENVRI services with one login.

FR14: API and Metadata Access for Service Providers

The system shall allow service providers to make their data and services accessible via well-defined APIs and metadata protocols.

FR15: Contribution to Platform Development

The system shall enable ENVRI developers to author, edit, or validate ENVRI-Hub artifacts for platform maintenance and improvement.

FR16: Service Provider Integration

The system shall support service providers in exposing datasets and services through APIs, including SPARQL endpoints, for easy discovery and access.

FR17: Data Catalogue Indexing Management

The system shall provide a backend for service operators to manage and update the Catalogue of Services, ensuring it is up-to-date and consistent.

FR18: Programmatic Data Access via APIs

The system shall offer programmatic access to ENVRI-Hub datasets and metadata via REST APIs and SPARQL endpoints for researchers.

FR19: Authentication and Authorization Mechanisms

The system shall implement authentication and authorization mechanisms (e.g., OAuth, SAML) to manage user access based on roles and permissions.

FR20: Search and Filter Training Content

The system shall allow users to search and filter training materials by topics or predefined learning paths in the Training Gateway.

FR21: Data Quality and Indexing Management

The system shall allow Knowledge Base curators to manage quality control and indexing pipelines for the Knowledge Base content.

FR22: Reproducible Research Support with Jupyter Notebooks

The system shall provide ENVRI-Hub analytical tools and workflow templates within Jupyter for researchers to create reproducible workflows.

FR23: Cross-Infrastructure Data Access

The system shall enable researchers to access and integrate data from various ENVRI research infrastructures without switching systems or re-authenticating.

FR24: Access to Science Demonstrators
The system shall provide Science Demonstrators to help researchers learn how to use specific datasets for research projects.

FR25: Monitoring Service for Infrastructure Performance
The system shall allow infrastructure managers to monitor the performance and availability of ENVRI-Hub services and components.

FR26: FAIR Data Compliance
The system shall ensure that datasets provided by RIs to ENVRI-Hub comply with FAIR principles.

FR27: Integration with External Identity Providers
The system shall allow users to log in using external identity providers (e.g., ORCID, eduGAIN) through ENVRI-ID, without needing new credentials.

FR28: Workflow Automation for Data Processing
The system shall provide tools to automate data processing workflows, streamlining repetitive tasks and focusing on analysis and interpretation.

FR29: Notifications on New Data
The system shall allow researchers to receive notifications about new or updated datasets that match their interests.

FR30: Indexing Pipeline Configuration
The system shall allow Knowledge Base operators to configure the indexing pipeline for adding new datasets, APIs, and computational notebooks.

FR31: Quality Control Configuration
The system shall allow Knowledge Base operators to set quality control criteria to ensure that users have access to only high-quality, relevant data.

FR32: System Health Monitoring
The system shall allow Knowledge Base operators to monitor the health and performance of the Knowledge Base system.

FR33: API-Based Programmatic Access
The system shall provide data scientists with programmatic access to the Knowledge Base via APIs for integration with external applications or automation of data retrieval.

FR34: Feedback Mechanism for Datasets and APIs
The system shall allow data scientists to submit feedback on datasets and APIs to help improve recommendations and data quality.

FR35: User Roles and Permissions Management

The system shall enable system administrators to manage user roles and permissions to control access to specific parts of the Knowledge Base.

FR36: Secure Access Management

The system shall ensure secure access to the Knowledge Base, maintaining data integrity and protecting user privacy.

FR37: Impact Measurement on Contributions

The system shall allow research community members to track how often their datasets are accessed or cited to measure the impact of their contributions.

FR38: AI Model Management for Search and Recommendation

The system shall allow machine learning engineers to update or swap AI models used for search and recommendation in the Knowledge Base to improve accuracy.

FR39: Exposing New Data Endpoints

The system shall allow API developers to expose new data endpoints for external applications to access the Knowledge Base.

8.2. Non-Functional Requirements

NFR1: Scalability

The platform must be scalable to handle increasing data volumes and users across multiple RIs without performance degradation.

NFR2: Availability

The platform must ensure high availability to allow continuous access to datasets, services, and tools without significant downtime, ensuring minimal disruption to research activities.

NFR3: Performance

The platform must respond quickly to data queries and service requests, ensuring smooth user interaction and efficient data processing workflows. Fast data retrieval and low latency are essential for user satisfaction.

NFR4: Security

The platform must implement robust security measures, including data encryption, user authentication, and role-based authorization, to ensure the integrity, confidentiality, and availability of user data and services.

NFR5: Interoperability

The platform must be interoperable with different Research Infrastructures and support cross-domain data access and integration through standardized protocols such as REST, SPARQL, and APIs. This will allow seamless data exchange and collaboration across different systems.

NFR6: Usability

The platform must offer an intuitive and user-friendly interface, ensuring that both technical and non-technical users can easily navigate, search for, and access data and services. It should accommodate a diverse user base with varying levels of expertise.

NFR7: Maintainability
The platform's codebase and infrastructure must be designed for easy maintenance, with clear documentation and modularity to facilitate updates, bug fixes, and system improvements.

NFR8: Extensibility
The platform must be easily extensible to accommodate new datasets, services, and functionalities as the needs of users and research infrastructures evolve. New features should be integrated without requiring major rework.

NFR9: Compliance with FAIR Principles
The platform must ensure that all data follows FAIR principles, making it easy for users to discover, access, and reuse datasets.

NFR10: Accessibility
The platform must be accessible to users with diverse needs, adhering to accessibility standards such as WCAG 2.1 to ensure that all users, including those with disabilities, can use the platform effectively.

NFR11: Reliability
The platform must ensure reliable data access and service delivery, minimizing the risk of failure or data loss, and providing redundancy and fault tolerance for critical services.

NFR12: Compliance with International Standards
The platform must adhere to international standards and best practices for software development, security, and data management, ensuring compatibility and compliance with existing systems and legal frameworks.

NFR13: Privacy and GDPR Compliance
The platform must comply with data privacy regulations, including the General Data Protection Regulation (GDPR), to ensure the protection of user data and respect for privacy.

NFR14: Responsiveness
The platform must have fast response times, ensuring users can quickly access search results, datasets, and services, providing a seamless user experience.

NFR15: Reusability of Components
The platform must be designed to ensure that its components (e.g., APIs, tools) are reusable across different services and research infrastructures, promoting efficiency in development and usage.

NFR16: Auditability and Traceability

The platform must provide audit logs and tracking mechanisms to monitor user activity and changes to datasets or services, ensuring transparency and accountability.

NFR17: Internationalization

The platform must support multiple languages and regional settings to accommodate a diverse, global user base, ensuring inclusivity.

NFR18: Data Quality Assurance

The platform must include mechanisms to ensure the accuracy, consistency, and reliability of the data and services provided and maintain high data quality standards for users.

8.3. User Stories

The following user stories capture the diverse needs and goals of the key stakeholders interacting with the ENVRI-Hub platform. Each story is framed from the perspective of different user roles, such as environmental researchers, policy-makers, research software engineers, service providers, and developers. These stories outline specific functionalities and interactions users require to utilize the platform effectively. By focusing on user-driven development, these stories emphasize the platform's role in enabling seamless access to environmental data, improving collaboration, supporting reproducible research, and fostering skill development through training resources. The user stories guide the development process, ensuring that the ENVRI-Hub remains flexible, scalable, and responsive to the evolving needs of its user base while maintaining adherence to best practices such as FAIR data principles, security, and interoperability.

Story 1: ENVRI-Hub Platform Access for Environmental Researchers
As an environmental researcher,

I want to access the ENVRI-Hub platform via its website to search and discover relevant data and services for my research project,
So that I can integrate these data into my analysis and workflows and utilize Jupyter Notebooks and other provided tools.

Story 2: Knowledge Base and Skill Development for Researchers
As an environmental researcher,

I want to access the Knowledge Base and training resources through the ENVRI-Hub platform,
So that I can explore additional resources and develop my skills using the training gateway.

Story 3: Data Integration and Customization
As a researcher using ENVRI-Hub,

I want to customize and adapt the available software tools and data services, So that I can tailor the data integration and processing to my specific research needs.

Story 4: Policy-Making Data Access
As a policy-making researcher working in environmental management,
I want to access relevant environmental data through ENVRI-Hub, such as water quality, species distribution, and greenhouse gas emissions,
So that I can make informed decisions and develop sustainable management strategies for coastal and agricultural regions.

Story 5: Service Integration for Research Software Engineers
As a research software engineer with advanced IT skills,
I want to access ENVRI-Hub services via APIs, SPARQL endpoints, and programming inter- faces like Python or R,
So that I can integrate ENVRI data into my experiments and workflows within Jupyter or other VREs.

Story 6: Training and Capacity Building for ENVRI Developers
As an ENVRI developer,
I want to access training materials and capacity-building tools through the Training Gateway,
So that I can improve my knowledge and skills relevant to developing the ENVRI-Hub.

Story 7: Cross-Domain Data Search
As an ENVRI researcher or service developer,
I want to perform cross-domain searches through the ENVRI-Hub Catalogue of Services,
So that I can find and utilize relevant datasets efficiently across multiple research infrastructures.

Story 8: Integration with VREs
As a researcher using a VRE like Jupyter or Galaxy,
I want to seamlessly import data assets from the ENVRI-Hub Catalogue into my VRE system,
So that I can integrate and process ENVRI data directly in my experiments.

Story 9: Single-Sign-OnAccess
As an ENVRI-Hub user,
I want to log in with a single account (Single-Sign-On) and use external identities (e.g., ORCID),
So that I can access multiple ENVRI services without logging in separately each time.

Story 10: Data and Metadata Access via APIs
As an ENVRI-Hub service provider,

I want to make my data and services accessible via well-defined APIs and metadata protocols,
So that researchers can easily find and integrate them into their workflows.

Story 11: Contribution to ENVRI-Hub Development
As an ENVRI-Hub Developer,
I want to contribute by authoring, editing, or validating ENVRI-Hub artifacts,
So that I can help build, improve, and maintain the platform in a collaborative environment.

Story 12: Service Provider Integration
As an ENVRI Research Infrastructure (RI) Service Provider,
I want to expose my datasets and services to the ENVRI-Hub through APIs, including SPARQL endpoints,
So that ENVRI-Hub users can easily discover and access my resources.

Story 13: Data Catalogue Management
As an ENVRI-Hub service operator,
I want to manage and update the Catalogue of Services via an administrative backend,
So that I can ensure the catalogue remains up-to-date, consistent, and functional for users.

Story 14: Data Search and Processing via APIs
As a researcher using API access,
I want to access ENVRI-Hub datasets and metadata programmatically via REST API and SPARQL endpoints,
So that I can incorporate these datasets directly into my computational workflows without using the web interface.

Story 15: Authentication and Authorization Management
As a service operator managing user access,
I want to implement authentication and authorization mechanisms within the ENVRI-Hub (e.g., OAuth, SAML),
So that I can manage who has access to specific data and services based on user roles and permissions.

Story 16: Training Content Search and Filtering
As an ENVRI-Hub user,
I want to search and filter training materials in the Training Gateway by topics or predefined learning paths,
So that I can find the most relevant resources to enhance my skills and knowledge efficiently.

Story 17: Data Quality and Indexing for Knowledge Base
As a Knowledge Base curator,

I want to manage the quality control and indexing pipelines for the Knowledge Base content,
So that the indexed data remains accurate, relevant, and up-to-date for ENVRI-Hub users.

Story 18: Reproducible Research with Jupyter Notebooks
As an ENVRI researcher using Jupyter Notebooks,
I want to use ENVRI-Hub's analytical tools and workflow templates within a Jupyter environment,
So that I can create reproducible research workflows using real-time ENVRI data.

Story 19: Cross-Infrastructure Data Access and Integration
As a researcher working across multiple RIs,
I want to access and integrate data from various ENVRI RIs without switching between systems or re-authenticating,
So that I can conduct cross-domain analysis seamlessly.

Story 20: Access to Science Demonstrators
As an ENVRI researcher,
I want to use Science Demonstrators provided by the ENVRI-Hub,
So that I can learn how to use specific datasets and apply them in research projects effectively.

Story 21: Monitoring Service for Infrastructure Performance
As an ENVRI-Hub infrastructure manager,
I want to monitor the performance and availability of the services and infrastructure components,
So that I can ensure the platform functions optimally and address any issues proactively.

Story 22: FAIR Data Compliance
As an ENVRI RI Data Manager,
I want to ensure that the datasets I provide to ENVRI-Hub are compliant with FAIR (Findable, Accessible, Interoperable, Reusable) principles,
So that researchers can easily discover and reuse the data in their research workflows.

Story 23: Integration with External Identity Providers As an ENVRI-Hub user,
I want to log in using external identity providers (e.g., ORCID, eduGAIN) through ENVRI-ID,
So that I can use my existing credentials without creating new ones for ENVRI-Hub.

Story 24: Workflow Automation for Data Processing
As an ENVRI researcher,
I want to automate data processing workflows using ENVRI-Hub's workflow tools and VRE integration,

So that I can streamline repetitive tasks and focus on analysis and interpretation.

Story 25: Natural Language Search
As a researcher,
I want to search the knowledge base using natural language queries,
So that I can quickly find relevant datasets, APIs, or computational notebooks for my research.

Story 26: Personalized Recommendations
As a researcher,
I want to get personalized recommendations based on my past searches,
So that I can discover related assets that I may have missed.

Story 27: Notifications on New Data
As a researcher,
I want to receive notifications about new datasets or updates to data that match my interests,
So that I can stay up to date with the latest findings.

Story 28: Configure Indexing Pipeline
As a KB operator,
I want to configure the indexing pipeline,
So that new datasets, APIs, and computational notebooks can be added to the knowledge base.

Story 29: Quality Control Configuration
As a KB operator,
I want to set quality control criteria for data being indexed,
So that only high-quality and relevant data is available to users.

Story 30: System Health Monitoring
As a KB operator,
I want to monitor the health and performance of the KB system,
So that I can ensure it is running smoothly.

Story 31: Programmatic Access via API
As a data scientist,
I want to access the KB programmatically via API,
So that I can integrate it with my own applications or automate data retrieval.

Story 32: Feedback on Datasets and APIs
As a data scientist,
I want to submit feedback on datasets and APIs,
So that I can help improve the quality of recommendations.

Story 33: User Roles and Permissions Management
As a system administrator,
I want to manage user roles and permissions,

So that only authorized users can access certain parts of the KB.

Story 34: Secure Access Management
As a system administrator,
I want to ensure secure access to the knowledge base,
So that data integrity and user privacy are maintained.

Story 35: Contribution of Datasets and Research Notes
As a research community member,
I want to contribute datasets or research notes to the knowledge base,
So that other researchers can benefit from my work.

Story 36: Impact Measurement on Contributions
As a research community member,
I want to see how many times my datasets have been accessed or cited,
So that I can measure the impact of my contributions.

Story 37: AI Model Management
As a machine learning engineer,
I want to update or swap AI models used for search and recommendation,
So that the KB continues to improve its accuracy and performance.

Story 38: Expose New Data Endpoints
As an API developer,
I want to expose new data endpoints for external applications to access the KB,
So that more systems can integrate with the knowledge base.