

FAIRiCUBE – F.A.I.R. INFORMATION CUBES

WP3 Processing

D3.4 Processing knowledge base services

Deliverable Lead: Stefania Morrone (EPS)

Deliverable due date: 30/04/2025

Version: 2.4

2025-05-02

Document Control Page

Document Control Page	
Title	D3.4 Processing knowledge base services
Creator	EPS
Description	D3.4 Processing knowledge base services
Publisher	"FAIRICUBE – F.A.I.R. information cubes" Consortium
Contributors	EPS
Date of delivery	05/05/2025
Type	R — Document, report
Language	EN-GB
Rights	Copyright "FAIRICUBE – F.A.I.R. information cubes"
Audience	<input checked="" type="checkbox"/> Public <input type="checkbox"/> Confidential <input type="checkbox"/> Classified
Status	<input type="checkbox"/> In Progress <input type="checkbox"/> For Review <input checked="" type="checkbox"/> For Approval <input type="checkbox"/> Approved

Revision History			
Version	Date	Modified by	Comments
0.1	08/02/2024	Stefania Morrone, Antonio Cozzolino, Liliana Martirano, Giacomo Martirano	First draft
0.2	23/02/2024	Mirko Gregor	Internal review
1.0	23/02/2024	Stefania Morrone	Final version
2.0	09/12/2024	Antonio Cozzolino	First draft
2.1	15/04/2025	Stefania Morrone, Antonio Cozzolino, Giacomo Martirano	Final Draft
2.2	30/04/2025	Jaume Targa, Lorena Banyuls	Review
2.3	30/04/2025	Giacomo Martirano	Minor updates
2.4	02/05/2025	Jaume Targa & Lorena Banyuls	Final review, ready for submission

Disclaimer

This document is issued within the frame and for the purpose of the FAIRICUBE project. This project has received funding from the European Union's Horizon research and innovation programme under grant agreement No. 101059238. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

This document and its content are the property of the FAIRICUBE Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the FAIRICUBE Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the FAIRICUBE Partners. Each FAIRICUBE Partner may use this document in conformity with the FAIRICUBE Consortium Grant Agreement provisions.

Table of Contents

Document Control Page	2
Disclaimer	3
Table of Contents	4
List of Figures.....	5
List of Tables.....	6
1 Introduction	7
2 Knowledge Base services design	9
2.1 Content Management.....	10
3 Digital Library	11
3.1 Digital Library sections.....	12
4 The Query Tool.....	13
4.1 The Query tool infrastructure.....	13
4.2 The Query Tool interface	14
4.2.1 Simplified Queries	15
4.2.2 Custom Queries	16
5 The Chatbot.....	18
5.1 Components and architecture.....	19
5.2 Workflow	20
5.3 Chatbot User Interface	20
6 Summary and outlook.....	22

List of Figures

Figure 1: KB landing page.	8
Figure 2: Digital Library landing page	11
Figure 3: KB components and structure	13
Figure 4: Query Tool landing page	14
Figure 5: Query result example	15
Figure 6: Simplified Queries interface	16
Figure 7: Custom Queries interface	17
Figure 8: RAG architecture	18
Figure 9: Chatbot query workflow	20
Figure 10 Chatbot interface	21



List of Tables

Table 1: List of KB services requirements	9
Table 2: KB Digital Library sections	12

1 Introduction

This document describes the [FAIRiCUBE Processing Knowledge Base services](#) (hereafter the **"Knowledge Base"** or **"KB"**). The Knowledge Base (KB) is a user-centric ecosystem designed to support data analysis and machine learning on large, complex datasets. Grounded in real-world FAIRiCUBE use cases, it provides expert guidance, practical tools, and actionable insights to help users develop data-driven ML models and extract meaningful results.

As a core component of FAIRiCUBE, and fully aligned with its mission, the Knowledge Base plays a key role in helping users, particularly those outside conventional Earth Observation (EO) domains, to discover, access, process and share gridded data and algorithms in accordance with the FAIR (Findable, Accessible, Interoperable, Reusable) and TRUST (Transparency, Responsibility, User focus, Sustainability, Technology) principles.

The KB consists of three key services:

- the [Digital Library](#)
- the [Query tool](#)
- the [Chatbot](#)

Together, these services provide a robust framework to guide use cases teams and end users through data analysis and processing tasks.

While the initial users of the KB are the project use cases teams, the potential end users are diverse and have varying levels of technical expertise- from scientists wishing to apply the models developed by the use cases to different areas, to urban policy makers interested in extracting insights from large and heterogeneous data collections, for instance to assess their city's resilience to climate change. The KB is structured to accommodate these varying levels of technical expertise, offering both high-level guidance and advanced tools.

The Digital Library offers streamlined access to key FAIRiCUBE documentation and use case resources. It includes onboarding material, examples, a self-training library focused on machine learning and data cubes and a practical collection of "Tips & Tricks" based on real challenges, solutions and lessons learned from use cases. The content shared is largely based on existing documentation in the project's GitHub repositories and on information in the analysis/processing (a/p) metadata, but it also includes input from selected external web resources. The exchange of information between experts/users also takes place through the FAIRiCUBE dedicated GitHub repositories.

The Query Tool offers an intuitive interactive gateway to explore FAIRiCUBE processing resources, ranging from datasets and pre-trained models to code libraries. As detailed in Section 4, the Query Tool leverages the metadata from analysis/processing resources described in the use cases. It is designed to accommodate users of all experience levels and supports basic keyword searches, pre-defined queries and advanced custom queries, allowing users to construct complex searches by specifying parameter values. This flexibility is also valuable for users who want to take a more granular approach to their research.

The Chatbot acts as an intelligent assistant, helping users navigate the FAIRiCUBE ecosystem. It answers questions about the project's goals, methodologies and tools, and directs users to relevant documentation, GitHub repositories and other resources. With its intuitive interface, the chatbot enhances accessibility and eases the learning curve for both new and expert users.

The main entry point to the Knowledge Base is the [FAIRiCUBE Hub](#), giving access to the services through the landing page shown in Figure 1: KB landing page.

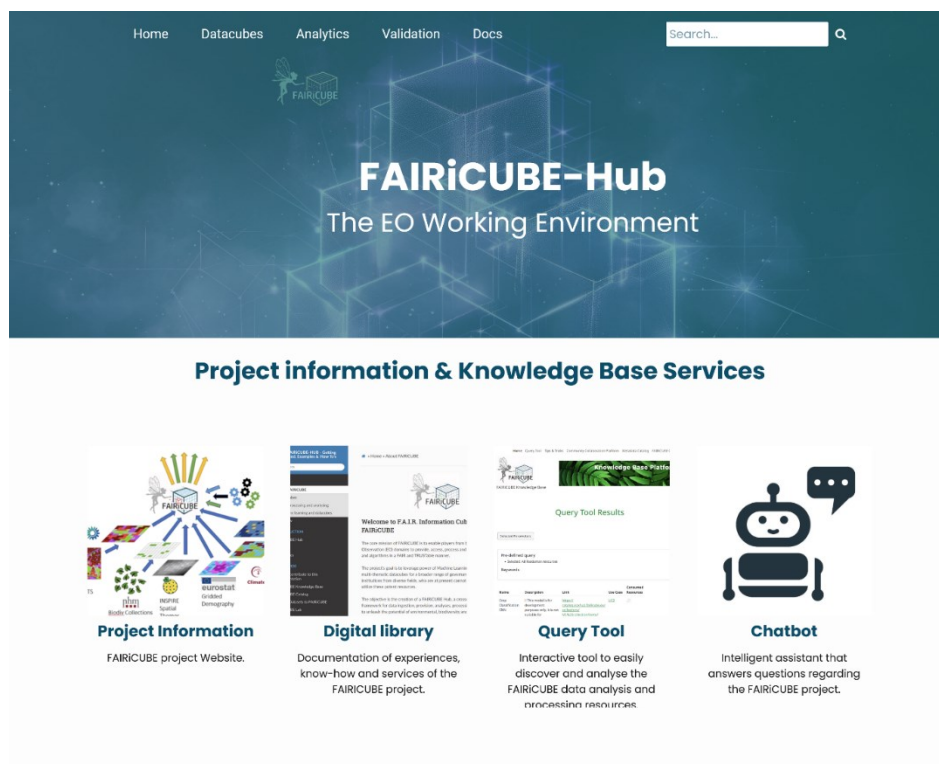


Figure 1: KB landing page.

The development of the KB is part of the activities related to Task 3.5 and should be considered in the general context of WP3, i.e. to provide guidance, recommendations, technical and implementation expertise to the FAIRiCUBE use cases in terms of data analysis and processing. This document represents the deliverable D3.4 "Processing knowledge base services" due (considering the project extension) at M34 i.e. by 30.04.2025.

2 Knowledge Base services design

The design of the Knowledge Base considered the following key principles:

- User inclusivity: the KB should serve a diverse audience, ranging from users with limited technical knowledge to specialists seeking in-depth solutions.
- Ease of use: the KB should support full access to and understanding of the KB content, especially supporting the non-expert users.
- Efficient access to information: the KB should allow users to quickly find relevant information, whether general or highly specific, through a searchable interface.
- Flexibility: the KB should support both structured browsing and direct searching, adapting to different user preferences and needs.
- Scalability: the KB should handle growing volumes of content while maintaining ease of use.

To better define the guided paths, three basic operational scenarios were considered:

- A biologist, with technical skills but no AI training, looking for a working example to analyse complex data collections and understand what factors make species grow happily.
- A researcher with an AI background, looking for details of applications and algorithms applicable in similar contexts/scenarios as FAIRiCUBE.
- The head of an environmental department, a data policy expert who is not interested in technical details but wants to understand whether the department's technical staff could reuse FAIRiCUBE results to solve similar problems.

Based on these potential scenarios and key design principles abovementioned, the following functional and non-functional requirements for the KB were formalised.

List of KB requirements

Req. ID	Req. description
R1	<ul style="list-style-type: none"> • GitHub is used as a platform to store technical artefacts and documentation (code, models, etc) as well as to have discussions and document issues. • An interface allows non-technical users to avoid interacting directly with GitHub.
R2	<ul style="list-style-type: none"> • The KB is accessible from the Hub
R3	The KB landing page contains: <ul style="list-style-type: none"> ◦ description of the KB scope, structure and functionalities ◦ links to different KB services
R4	The specific services landing pages contain: <ul style="list-style-type: none"> ◦ descriptive text ◦ links to related GitHub repositories ◦ links to specific artefacts
R5	<ul style="list-style-type: none"> • The KB contains use cases sections, with descriptive text as well as links to the metadata of related datasets and a/p resources. • It must complement without overlapping the website uc description page as well as uc-specific repositories descriptions.
R6	<ul style="list-style-type: none"> • A user interface (query tool) allows “queries” to resources metadata. <ul style="list-style-type: none"> • Basic (e.g. search by keywords) as well as complex customised queries must be enabled.
R7	<ul style="list-style-type: none"> • An intelligent assistant answers questions regarding the FAIRiCUBE project in human language.

Table 1: List of KB services requirements

2.1 Content Management

The FAIRiCUBE Knowledge Base (KB) draws primarily from content in project [GitHub repositories](#) and from metadata documentation of processing and analysis resources, but it also integrates input from selected external sources.

Knowledge exchange also takes place through FAIRiCUBE's dedicated GitHub repositories, enabling collaboration between users and experts.

The GitHub repositories relevant to the KB are listed below:

- [use case repositories](#). These repositories capture practical knowledge and artifacts from specific project implementations. They reflect real-world challenges, solutions and workflows unique to each use case.
- [common code](#) repository. This repository provides general-purpose tools, scripts and step-by-step guides that are not tied to any single use case. These resources could be reused across different projects and user types (e.g. scripts for generating Cloud-Optimised GeoTIFFs or guide on using QGIS for raster data processing).
- [lessons-learnt](#) repository collects use cases challenges and related successes, failures, solutions and workarounds.

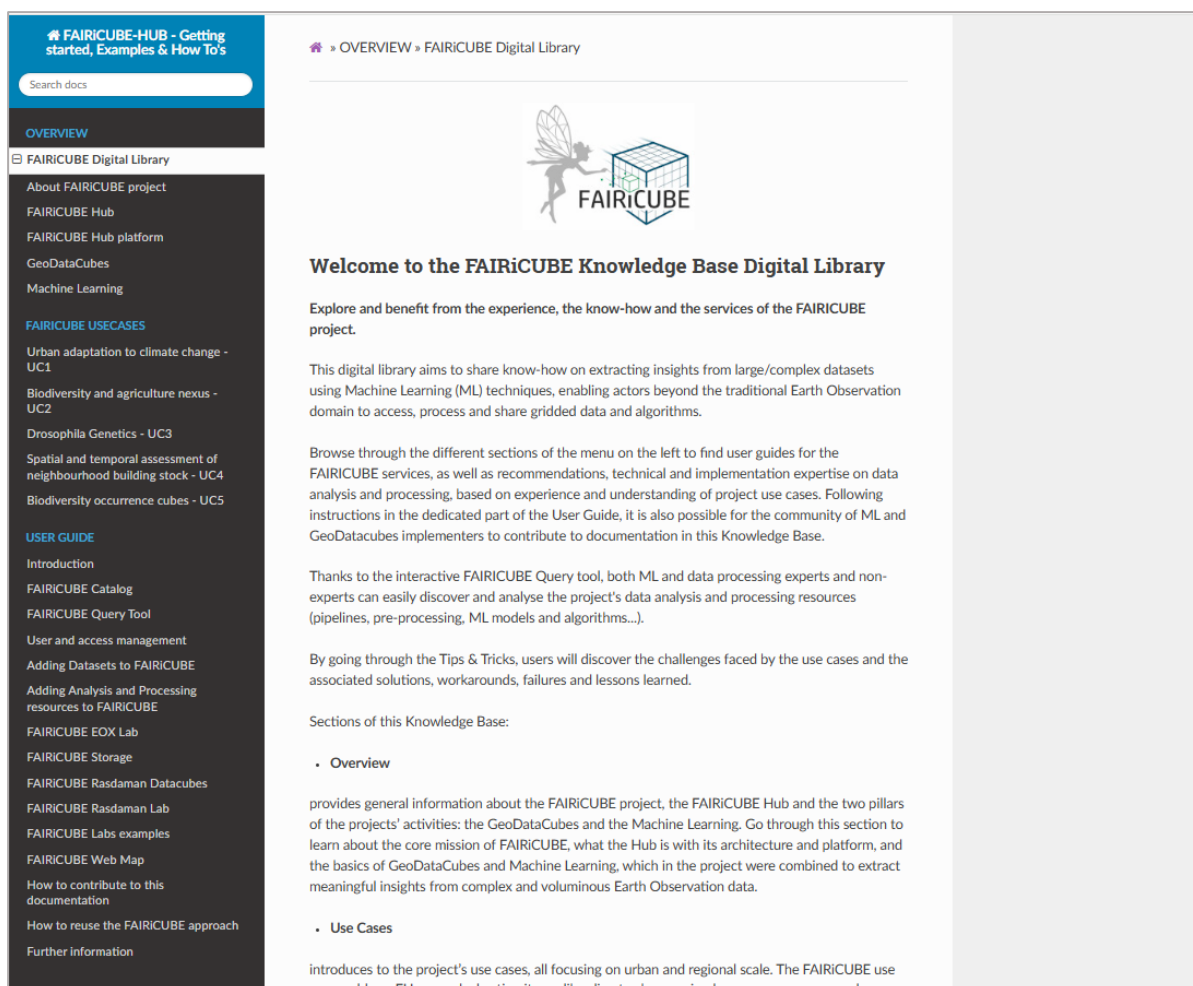
Partners can contribute to content shared through the Knowledge Base in several ways:

- documenting processing and analysis resources (such as algorithms, pipelines, preprocessing steps) developed within their use cases. This is done by creating associated metadata that are then accessible to all KB users via the Query Tool.
- sharing practical experiences and insights, including successes, failures and workarounds. These contributions can be made by posting issues in the [lessons-learnt](#) repository using a [dedicated template](#). This information becomes available in the lessons learnt section of the Digital Library.
- contributing to the different sections of the Digital Library. This is done by editing or adding content through GitHub pull requests in the [collaboration-platform](#) repository. A [contribution guide](#) is available in the Digital Library to support this process.

3 Digital Library

The [Digital Library](#) service provides easy and structured access to key FAIRiCUBE documentation and use case materials. It includes onboarding guidelines, practical examples, a self-training section focused on machine learning and data cubes, and a collection of practical “Tips & Tricks” drawn from the challenges, solutions and lessons learned from the use cases. As described in section 2.1, the content is primarily sourced from existing documentation in the project’s GitHub repositories and from the metadata of the analysis/processing resources. It also includes curated input from relevant external web resources. It is possible to contribute to Knowledge Base Digital Library using GitHub pull request following the guidance provided in the User Guides section.

The content managed via the GitHub repositories is connected to [ReadTheDocs](#), the platform that hosts, builds and version-controls documentation. This platform integrates directly with the FAIRiCUBE GitHub repositories resulting in automatic deployments of contents to the Digital Library. Users can explore the Digital Library via the menu on the left, where they’ll find user guides, technical recommendations and implementation insights based on the project’s practical experience. The home page of the Digital Library is illustrated in Figure 2: Digital Library landing page.



FAIRiCUBE-HUB - Getting started, Examples & How To's

Search docs

OVERVIEW

FAIRiCUBE Digital Library

- About FAIRiCUBE project
- FAIRiCUBE Hub
- FAIRiCUBE Hub platform
- GeoDataCubes
- Machine Learning


FAIRiCUBE USECASES

- Urban adaptation to climate change - UC1
- Biodiversity and agriculture nexus - UC2
- Drosophila Genetics - UC3
- Spatial and temporal assessment of neighbourhood building stock - UC4
- Biodiversity occurrence cubes - UC5

USER GUIDE

- Introduction
- FAIRiCUBE Catalog
- FAIRiCUBE Query Tool
- User and access management
- Adding Datasets to FAIRiCUBE
- Adding Analysis and Processing resources to FAIRiCUBE
- FAIRiCUBE EOX Lab
- FAIRiCUBE Storage
- FAIRiCUBE Rasdaman Datacubes
- FAIRiCUBE Rasdaman Lab
- FAIRiCUBE Labs examples
- FAIRiCUBE Web Map
- How to contribute to this documentation
- How to reuse the FAIRiCUBE approach
- Further information

OVERVIEW » FAIRiCUBE Digital Library



Welcome to the FAIRiCUBE Knowledge Base Digital Library

Explore and benefit from the experience, the know-how and the services of the FAIRiCUBE project.

This digital library aims to share know-how on extracting insights from large/complex datasets using Machine Learning (ML) techniques, enabling actors beyond the traditional Earth Observation domain to access, process and share gridded data and algorithms.

Browse through the different sections of the menu on the left to find user guides for the FAIRiCUBE services, as well as recommendations, technical and implementation expertise on data analysis and processing, based on experience and understanding of project use cases. Following instructions in the dedicated part of the User Guide, it is also possible for the community of ML and GeoDataCubes implementers to contribute to documentation in this Knowledge Base.

Thanks to the interactive FAIRiCUBE Query tool, both ML and data processing experts and non-experts can easily discover and analyse the project's data analysis and processing resources (pipelines, pre-processing, ML models and algorithms...).

By going through the Tips & Tricks, users will discover the challenges faced by the use cases and the associated solutions, workarounds, failures and lessons learned.

Sections of this Knowledge Base:

- Overview**

provides general information about the FAIRiCUBE project, the FAIRiCUBE Hub and the two pillars of the projects' activities: the GeoDataCubes and the Machine Learning. Go through this section to learn about the core mission of FAIRiCUBE, what the Hub is with its architecture and platform, and the basics of GeoDataCubes and Machine Learning, which in the project were combined to extract meaningful insights from complex and voluminous Earth Observation data.

- Use Cases**

introduces to the project's use cases, all focusing on urban and regional scale. The FAIRiCUBE use cases address EU green deal action items like climate change, circular economy, energy and

Figure 2: Digital Library landing page

3.1 Digital Library sections

The Digital Library content is grouped into the below six sections:

KB section	Brief description
OVERVIEW	Offers a general overview to the FAIRiCUBE Project, FAIRiCUBE Hub and the two fundamental entities: Datacubes and Machine Learning
FAIRiCUBE USECASES	Introduces to the project use cases, all focusing on urban and regional scale. The FAIRiCUBE use cases address EU green deal action items, like climate change, circular economy, energy and biodiversity. Specifically, they investigate the adaptation to climate change, the nexus between biodiversity and agriculture, the environmental adaptation genomics in drosophila, the spatial and temporal assessment of neighbourhood building stock, the validation of Phytosociological Methods through Occurrence Cubes.
USER GUIDE	Provides instruction for use and access to the FAIRiCUBE Services: the FAIRiCUBE Catalog describing the project's datasets (used and produced by the use cases) and processing resources (pipelines, pre-processing, ML models and algorithms), the Query Tool to query over the resources, the EOX Lab , the rasdaman Lab and the FAIRiCUBE storage to let the user try the environments and the algorithms. For each service detailed descriptions, examples and instructions for use are available.
SELF-TRAINING	Provides insights into the basic concepts of data science as applied/applicable to FAIRiCUBE, together with a list of useful links to external resources, instructions for use and examples. Those new to ML and geodatabases can use the topics in this section to acquire the basic concepts/skills needed to understand and benefit from the work done in the use cases.
AI TOOLKIT	Introduces the most used tools and frameworks in the AI field, with description and examples of use. In the documents of this section one can find link to practical resources and code snippets that demonstrate how to apply AI methods across a range of domains, from machine learning and deep learning to natural language processing and computer vision.
GEODATACUBES TOOLKIT	Introduces the most used tools and frameworks in the GeoDatacubes field, with links to description page and examples of use. In the documents of this section one can find links to tools enabling use of the GeoDatacubes in a wide range of applications that require large-scale geospatial analysis, like the environmental monitoring, climate change research, agriculture, biodiversity and nature conservation, urban planning, disaster management and much more.
LESSONS LEARNT AND TIPS&TRICKS	Documents the use cases challenges, their successes & failures, solutions & workarounds. Reading through the content of this section will prevent you from wasting time trying to solve problems that have already been faced by use cases and for which they have already found a solution. You will also learn which approaches have been successful, which have failed and which points are still open.
EXTERNAL RESOURCES	Contains useful links to external resources used in FAIRiCUBE, like the Sentinel Hub , the Copernicus Hub providing comprehensive and accessible access to a wealth of Earth observation data gathered by the Sentinel satellites, the EOxHub / Euro Data Cube providing the FAIRiCUBE EOx Lab , the rasdaman platform the integrated solution for managing and providing access via standardised API to spatio-temporal datacubes.

Table 2: KB Digital Library sections

4 The Query Tool

The Query Tool service has been specifically designed to facilitate the understanding, access and reuse of FAIRICUBE processing resources, including data processing pipelines, preprocessing techniques, ML models, algorithms, pre-trained models and code libraries. It allows users to interactively search for processing resources in different ways, according to their level of expertise and the desired level of granularity. It is possible to carry out keyword searches, pre-configured queries and more advanced custom queries, thus offering the flexibility needed for simple as well as more in-depth/specialised research. For each resource that meets the criteria, key properties are shown to clarify its function and how it was used in the use case, along with a link to the full details in the [metadata Catalog](#).

4.1 The Query tool infrastructure

The Query Tool is powered by a PostgreSQL database. All SQL statements issued by the Query Tool are executed against this database, which contains information derived from the metadata of the analysis/processing resources. Specifically, each time a processing resource metadata is ingested into the metadata catalog, a web application automatically populates the database tables with the related information (for details see D4.3). Specifically, the database tables do not store all the information contained in the metadata records, but only those pieces of information required to filter resources according to the user's request. The Query Tool relies on a web-application, providing:

- a user interface, through which the users can both access static content web pages and interact dynamically with information on the processing resources of the project (via Query Tool).
- a query engine
- links to multiple external sources: first and foremost, the project's metadata catalog and GitHub repositories.

The Query Tool web-application is coded in Python and using the Django web-framework. The components and structure of the Query Tool are illustrated in Figure 3: KB components and structure.

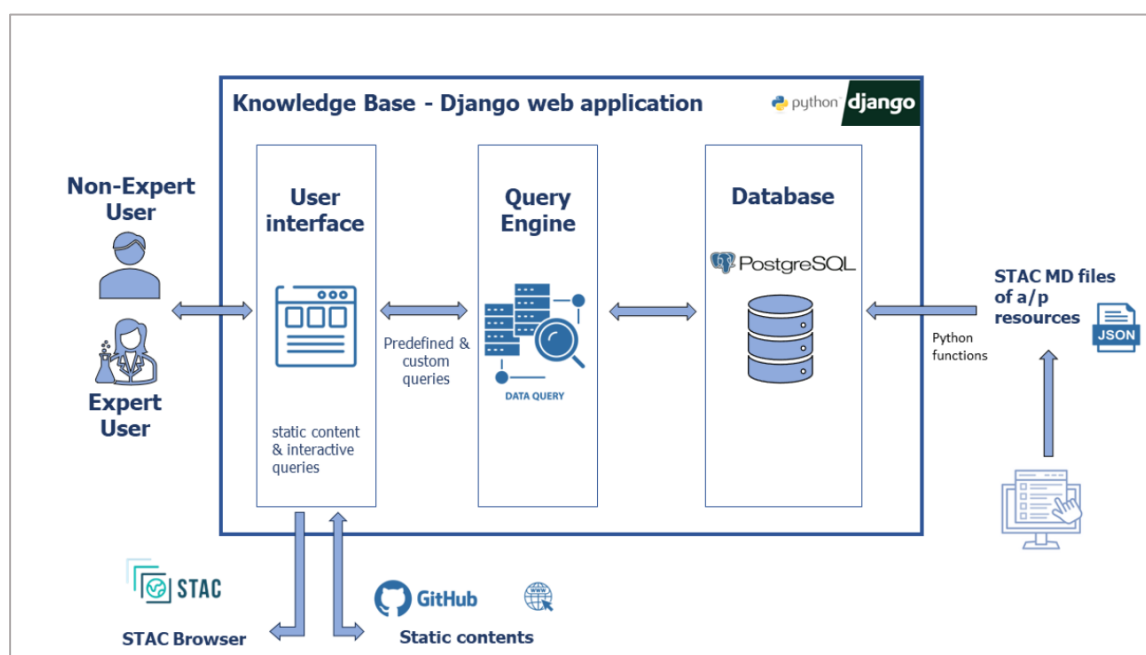


Figure 3: KB components and structure

4.2 The Query Tool interface

The Query Tool interface enables filtering a/p resources based on:

- simplified queries, that filter resources by keyword or pre-defined query.
- custom queries created by expert users.

The query results return a list of a/p resources that meet all the query criteria specified. For each result entry, the interface displays the resource name, a concise description, a direct link to the corresponding metadata record in the metadata catalog, a link to the associated use case page(s) within the Digital Library and (where available) the link to the detailed description of the hardware/software resources consumed during the execution of the specific a/p resource. The Query Tool landing page is shown in Figure 4: Query Tool landing page, while an example of a typical results page is shown in Figure 5: Query result exampleFigure 5.

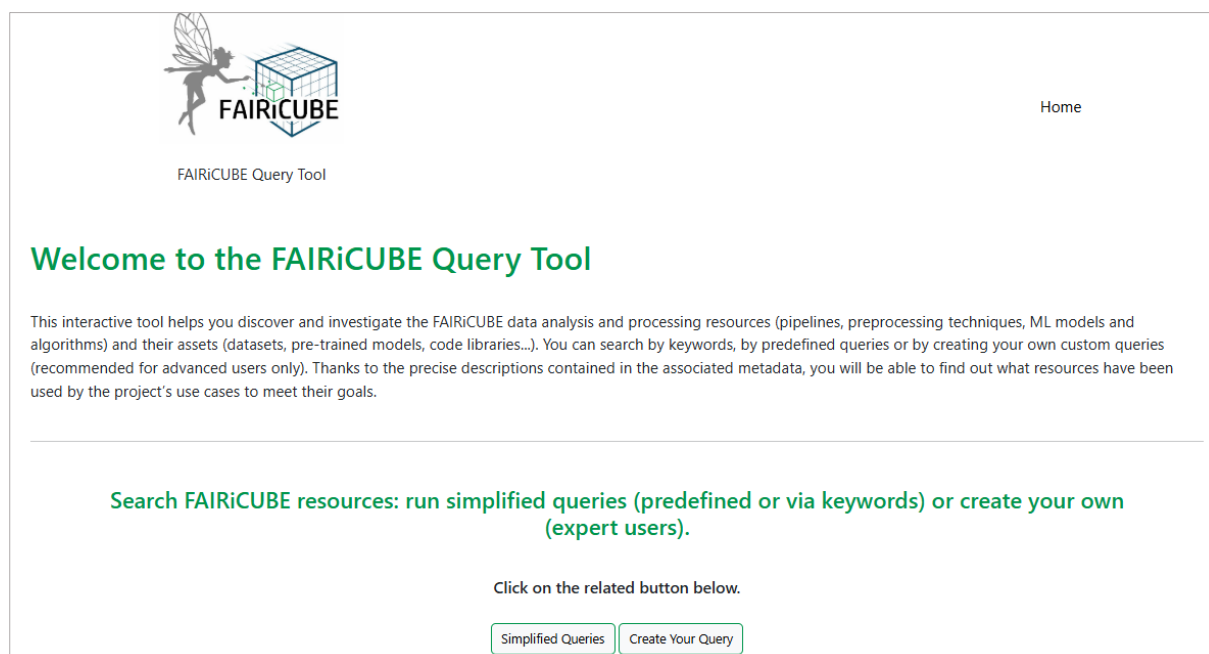


Figure 4: Query Tool landing page

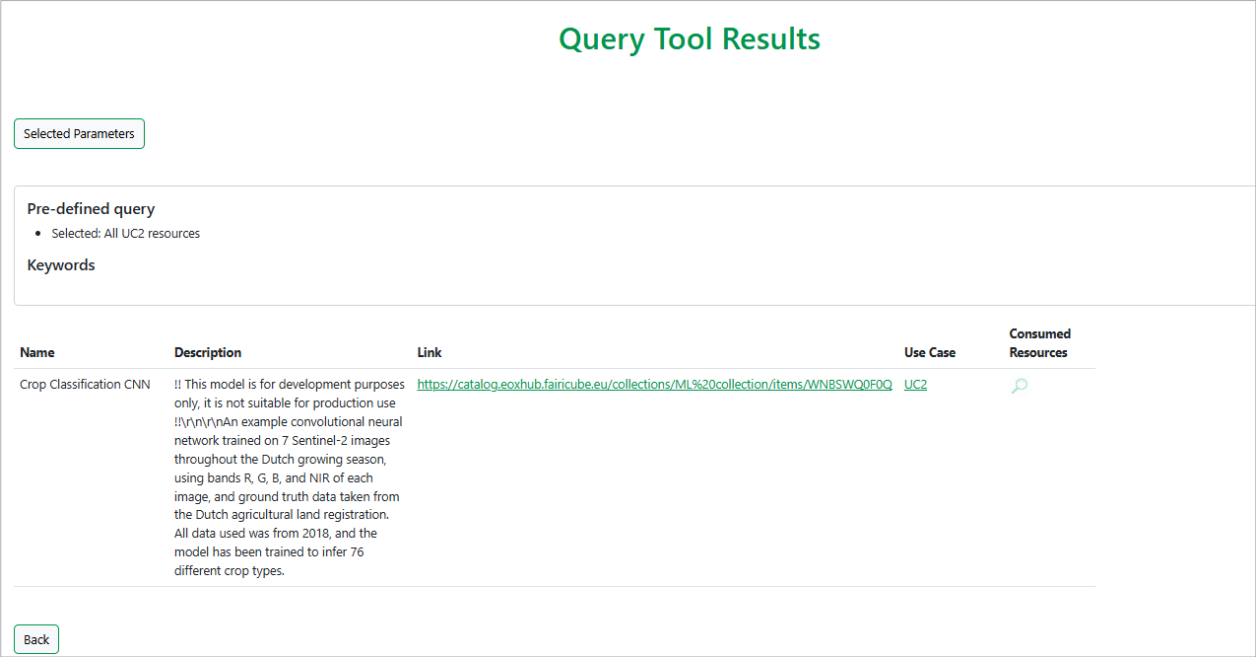


Figure 5: Query result example

4.2.1 Simplified Queries

The Simplified Query interface enables users to perform targeted searches by leveraging both keyword-based input and predefined query parameters. As users type into the keyword field, the system dynamically suggests matching terms from the metadata, displaying only those keywords that are already present in the catalog. This facilitates the selection of relevant, standardised keywords and ensures consistency across searches. Users can add one or more keywords to construct compound queries, enabling precise filtering of resources tagged with the selected terms.

Additionally, the interface provides a set of predefined queries - available via a drop-down menu- that allow users to quickly retrieve resources based on commonly used criteria. These include, for example, retrieving all machine learning resources, all resources associated with a specific use case, or those related to a particular platform. By supporting both free-text keyword search and controlled vocabulary through predefined filters, the Simplified Query interface streamlines the discovery of relevant resources and promotes metadata-driven exploration. The Simplified Queries interface is illustrated in Figure 5: Query result example.

Simplified Queries

Filter resources by keyword or pre-defined query.

Keywords

Start typing your keyword here, select an option from the keyword drop-down list that appears, click 'Add keyword' and finally click 'Run query' to retrieve the associated resources.

Add keyword

Run Query

Predefined Query

Select a predefined query to quickly access resources based on common search criteria.

Query list:

All ML resources

Run Query

Figure 6: Simplified Queries interface

4.2.2 Custom Queries

Custom Queries enable users to define values for one or more query parameters by selecting from dynamically populated drop-down menus. Each drop-down presents the complete set of distinct values currently available for the corresponding parameter in the resource database, ensuring that all selections align with existing metadata entries. The Custom Queries interface is illustrated in Figure 7: Custom Queries interface.



Custom Queries

Filter resources by selecting a keyword and /or one or more parameters (click on related "Add" button).

Start typing your keyword here, select an option from the keyword drop-down list that appears and then click 'Add keyword'.

Keyword

Add keyword

Parameter	Value	
Choose Algorithm	-	Add to query
Choose Approach	-	Add to query
Choose Architecture	-	Add to query
Choose Conditions for Access and Use	-	Add to query
Choose Framework	-	Add to query
Choose Main Category	-	Add to query
Choose Objective	-	Add to query
Choose Os	-	Add to query
Choose Platform	-	Add to query
Choose Processor	-	Add to query
Choose Use Case	-	Add to query

Run Query

Figure 7: Custom Queries interface

5 The Chatbot

The FAIRiCUBE Chatbot is an intelligent assistant that answers questions regarding the FAIRiCUBE project. This chatbot acts as a guide to FAIRiCUBE resources, answering questions about its objectives, methodologies and knowledge base. With an intuitive interface, it efficiently retrieves project information, clarifies technical aspects and directs users to relevant documents and data sources within the FAIRiCUBE knowledge base and GitHub repositories.

The chatbot allows users to find and navigate the Digital Library and the GitHub artifacts. It integrates Retrieval-Augmented Generation (RAG) technique (shown in Figure 8: RAG architecture) with state-of-the-art technology to ensure that responses are accurate, context-sensitive and reliable. The integration of retrieval-based searching and generative AI allows the chatbot to provide a smooth way of exploring and working with both structured and unstructured information. More details about RAG can be found in Section 5.1 below.

The chatbot is designed not only to answer user queries with speed and accuracy but also to serve as an important assistant in navigating the wide and complex resources of the FAIRiCUBE project. Its ability to provide contextually relevant responses, coupled with a strong retrieval mechanism, places it as a central tool for those looking to interact with the project.

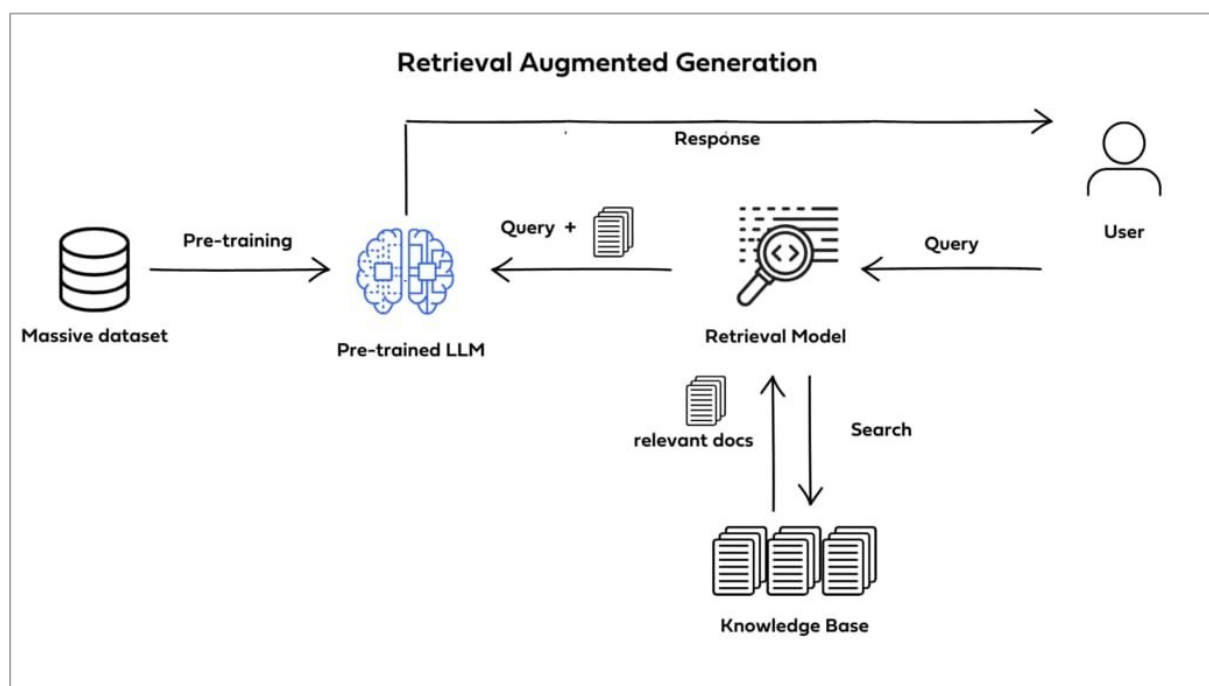


Figure 8: RAG architecture

5.1 Components and architecture

Retrieval-Augmented Generation (RAG) is a technique that enhances the capabilities of generative artificial intelligence models by allowing them to retrieve and utilize information from external sources. It is a hybrid AI model framework that combines the strengths of information retrieval and generative language models to provide accurate and contextually relevant responses.

The RAG-based architecture of the chatbot integrates three significant components - OpenAI's text-embedding-3-large model, chromaDB and GPT-4o - into a unified system. This combination enables the chatbot to provide real-time access to pertinent information while augmenting it with capabilities for natural language generation. The use of chromaDB for the information retrieval guarantees that the system is scalable with new data being incorporated, thereby allowing adaptation to the changing and enrichment of the FAIRiCUBE project information. In particular:

- the embedding model (text-embedding-3-large) plays a vital role by enabling the system to retrieve relevant information efficiently and meaningfully from a knowledge base or document corpus. The model functions by encoding both the query (the input question) and the documents (or chunks of text) into dense vector representations in a shared embedding space. These embeddings capture the semantic meaning of the text, allowing the system to retrieve relevant information even in the absence of exact keyword matches.
This semantic encoding facilitates similarity matching, where the system compares the vector representation of the query with precomputed document embeddings to identify the most relevant texts. Using similarity metrics like cosine similarity, the RAG framework can prioritize documents based on semantic relevance rather than relying solely on exact word matches. This approach is especially important for unstructured datasets, where retrieving contextually meaningful information is key.
Once the relevant documents are retrieved, they provide essential context for the generative model (such as GPT) to produce coherent and accurate responses. The embedding model ensures that the retrieval process brings back the most useful data, which directly impacts the quality of the generative model's output.
- ChromaDB plays a pivotal role by managing the document embeddings and providing the infrastructure to retrieve them using similarity metrics like cosine similarity or Euclidean distance. When a query is encoded into an embedding, ChromaDB quickly compares it with the pre-stored document embeddings to return the most relevant results. This process ensures that the RAG system retrieves contextually appropriate data, even from large and complex datasets. Another key function of ChromaDB is its support for scalability. It is optimised for handling vast amounts of embeddings efficiently, which is essential for large-scale knowledge bases used in RAG systems. Its design allows for real-time updates, so new data can be added and indexed dynamically without interrupting the retrieval process.
- GPT-4o (or similar large language model) is the generative part responsible for producing coherent and contextually relevant responses. Having identified the most appropriate pieces of information from its knowledge base using a vector database like ChromaDB and embeddings produced by a model like text-embedding-3-large, GPT-4o takes that retrieved information as input to generate its output.
The primary function of GPT-4o in RAG is to synthesize the retrieved context with the user's query to create responses that are accurate, fluent and grounded in the provided data. It processes both the user query and the retrieved documents, understanding the relationship between them to deliver a meaningful answer. By grounding its response in the retrieved context, GPT-4o mitigates the risks of hallucination (producing unsupported information), which can occur when the model generates answers based only on its pre-trained knowledge. In addition, GPT-4o has ability to handle nuanced natural language allowing it to interpret ambiguous or complex queries effectively and leverage the retrieved data to provide precise, context-aware responses.

5.2 Workflow

The Figure 9: Chatbot query workflow below illustrates the workflow of interaction between user and chatbot. The workflow starts with a user submitting a query through the interface of the chatbot.

1. The input is processed by the OpenAI text-embedding-3-large model, which embeds the query in a high-dimensional vector space. These embeddings function as quantitative representations of the input's semantic content, enabling the system to perform efficient semantic comparisons.
2. This query vector is then passed on to ChromaDB, a very performant open-source vector database serving as the chatbot's retrieval engine. ChromaDB with pre-computed embeddings for relevant knowledge base documents identifies the most contextually appropriate content in the shortest time. The vectors in ChromaDB were obtained from the GitHub artifacts and all Knowledge Base Digital Library documents and information using OpenAI's text-embedding-3-large model. This allows the chatbot to be provided with the information that is specific to the FAIRiCUBE project and therefore allows the answers to be contextualised.
3. The obtained information is combined with the user's query and sent to the large language model, GPT-4o, provided by OpenAI. GPT-4o conditions the input query with the obtained context to generate a response coherent, precise and similar in tone and structure to human communication. By anchoring responses in the data provided through ChromaDB, GPT-4o minimizes the risk of hallucinated or factually incorrect responses. This combination of active information retrieval and generative reasoning ensures that the chatbot provides answers that are both conversational and grounded on real data.

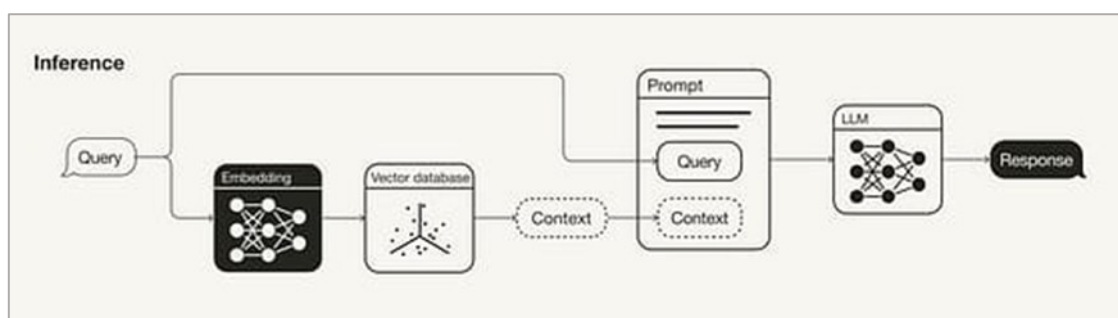


Figure 9: Chatbot query workflow

5.3 Chatbot User Interface

Figure 10 Chatbot interface below illustrates the chatbot user interface. An intuitive interface developed with Gradio makes interaction easy. Gradio provides a transparent, reactive and accessible front-end for the user to ask questions and get answers. This is achieved with the design so that the users, regardless of their level of technical proficiency, will be able to engage with the chatbot naturally and effortlessly. At the time of writing, the chatbot is still in an experimental stage and the users may experience latency in the responses.

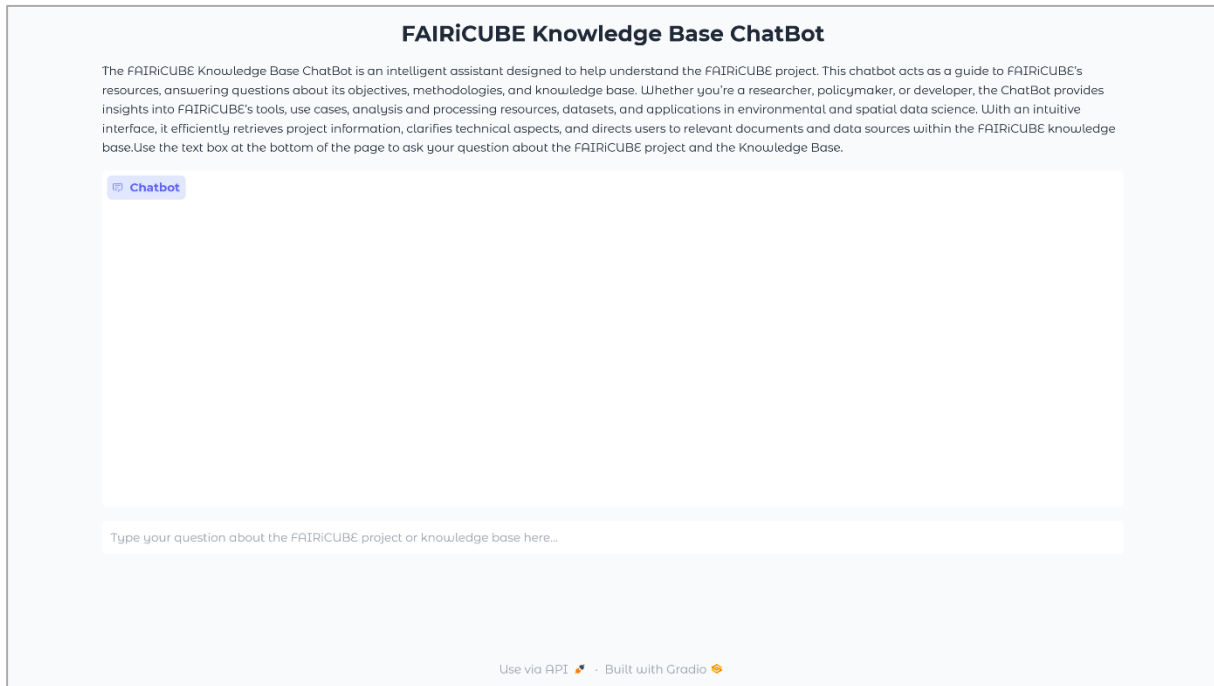


Figure 10 Chatbot interface

6 Summary and outlook

The Knowledge Base (KB) is a central FAIRiCUBE component in fulfilling project core mission: enabling users, particularly those outside traditional Earth Observation (EO) domains, to discover, access, process and share gridded data and algorithms in accordance with the FAIR (Findable, Accessible, Interoperable, and Reusable) and TRUST (Transparency, Responsibility, User focus, Sustainability, and Technology) principles.

Aligned with the project's overarching goal of democratizing access to multi-thematic datacubes and machine learning (ML) capabilities, the KB acts as an integrative support environment, lowering technical barriers and expanding usability for governance bodies, domain experts and research institutions from diverse disciplines. The KB supports this objective through the following three integrated services:

1. Digital Library: Centralised access to FAIRiCUBE documentation, onboarding materials, machine learning resources and practical "Tips & Tricks" based on use case experiences. It enables users to explore and understand practical applications of EO and ML on datacubes.
2. Query Tool: An interactive interface that allows users to perform both simplified and custom metadata-driven searches over the project's resources. By enabling precise filtering based on standardised parameters and existing metadata values, the tool facilitates efficient resource discovery.
3. Chatbot: An intelligent support agent that assists users in navigating the KB, clarifying concepts and accessing relevant resources. It serves as an entry point for users with varying levels of technical expertise.

Rooted in the concrete requirements and workflows of FAIRiCUBE use cases, the KB delivers expert guidance, ready-to-use tools and actionable insights, supporting users to develop and deploy data-driven ML models. By fostering cross-domain knowledge transfer and reducing the entry threshold for working with EO data, the KB directly contributes to the FAIRiCUBE vision of broadening participation and impact across the data and research ecosystem.

The FAIRiCUBE Knowledge Base (KB) is strategically positioned to become a widely used resource, well beyond the initial project scope. By transforming technical evidence into actionable knowledge, it has great potential to support evidence-based decision making in a variety of areas, including policy development, urban planning, environmental monitoring and societal resilience. Its modular and scalable architecture allows it to be adapted to a wide range of domains and user communities. As user engagement grows, the KB content and tools can evolve through ongoing contributions, improving both usability and overall impact.

Future enhancements could include greater personalisation - such as customised user interfaces, adaptive user journeys and multilingual capabilities - and the integration of AI-driven features such as advanced natural language processing, semantic search and intelligent recommendations. Improved interoperability with external data systems via APIs could further strengthen its role within the broader data and knowledge ecosystem. In summary, the Knowledge Base has a potential to become a strategic asset that will continue to grow in functionality and usability and contribute to Open Science and Policy Impact.