SEcure Decentralised Intelligent Data MARKetplace

D4.6 Data sharing platform and incentives - Final version

Document Identification	
Contractual delivery date:	31/07/2025
Actual delivery date:	30/09/2025
Responsible beneficiary:	ATOS
Contributing beneficiaries:	FV, NUID UCD, SIE
Dissemination level:	PU
Version:	1.0
Status:	Final

Keywords:
Marketplace, Data Sharing, User Interface, User Experience, Data Processing

This document is issued within the frame and for the purpose of the SEDIMARK project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No.101070074. and is also partly funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission or UKRI.

Document Information

Document Identification			
Related WP	WP4	**Related Deliverables(s):**	SEDIMARK_D4.5
Document reference:	SEDIMARK_D4.6	**Total number of pages:**	66

List of Contributors	
Name	**Partner**
Maxime Costalonga Arturo Medela	ATOS
Eero Jalo	FV
Elias Tragos Aonghus Lawlor Diarmuid O'Reilly Morgan Erika Duriakova Honghui Du Qinqin Wang Neil Hurley	NUID UCD
Gabriel Danciu Stefan Jarcau	SIE

Document History			
Version	**Date**	**Change editors**	**Change**
0.1	18/03/2025	ATOS	First version of document based on previous iteration (SEDIMARK_D4.5)
0.2	30/06/2025	NUID UCD, SIE, FV	Updated sections 6, 7 (data processing and AI dashboards), section 8 (Recommender), section 10 (data sharing incentives)
0.3	22/09/2025	ATOS	Updated intro, section 2 (overview), 3 (onboarding), 4 (catalogue), 5 (offerings), 9 (open data enabler) and conclusions. Filled section 9 (open data enabler)

Document History			
Version	**Date**	**Change editors**	**Change**
0.4	29/09/2025	ATOS	Reformatting of section 6, 7 and 8 to harmonize with the rest of the document
0.5	29/09/2025	ATOS	Apply review comments
0.9	30/09/2025	ATOS	Quality Review Form
1.0	30/09/2025	ATOS	FINAL VERSION TO BE SUBMITTED

Quality Control		
Role	**Who (Partner short name)**	**Approval date**
Reviewer 1	Tarek Elsaleh (SURREY)	29.09.2025
Reviewer 2	Grigorios Koutantos (WINGS)	25.09.2025
Quality manager	María Guadalupe Rodríguez (ATOS)	30.09.2025
Project Coordinator	Miguel Angel Esbrí (ATOS)	30.09.2025

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
BM25	Best Matching 25
DCAT	Data Catalog
DID	Decentralised Identifier
DLT	Distributed Ledger Technology
DNN	Deep Neural Network
DPO	Data Processing Orchestration
Dx.y	Deliverable number y belonging to WP x
EDC	Eclipse Dataspace Components
ETL	Extract Transform Load
FL	Federated Learning
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
LLM	Large Language Model
LSI	Latent Semantic Indexing
LSTM	Long Short-Term Memory
ML	Machine Learning
NGSI-LD	Next Generation Service Interface – Linked Data
NLP	Natural Language Processing
ODRL	Open Digital Rights Language
RAG	Retrieval Augmented Generation
RM3	Relevance Model 3
RS	Recommender System
SVD	Singular Value Decomposition
UI	User Interface

Abbreviation / acronym	Description
VC	Verifiable Credential
URL	Uniform Resource Locator
WPx	Work Package x
YAML	Yet Another Markup Language

Executive Summary

This document corresponds to the Deliverable D4.6 of the SEDIMARK project, named “Data sharing platform and incentives – Final version”. Its goal is to describe the SEDIMARK marketplace, a web frontend application constituting the entry point for users to the SEDIMARK ecosystem and all the functionalities it offers. It updates a previous Deliverable, SEDIMARK_D4.5 “Data sharing platform and incentives – First version” [1], submitted in December 2023, which described the Marketplace in an early development stage. This final version provides an up-to-date description of it, focusing on how users can navigate the web interface to access all features of SEDIMARK.

After a brief introduction of the scope of the marketplace and the expected users’ persona in **Chapter 2**, the subsequent chapters of this deliverable describe the graphical user interfaces of the marketplace, grouped by functionality:

- **Chapter 3**: onboarding of new users, guiding them through the creation of their identity.
- **Chapter 4**: browsing the Offering catalogue.
- **Chapter 5**: how users can manage their provided or consumed offerings, as well as how to publish an offering.
- **Chapter 6**: accessing the data processing toolbox.
- **Chapter 7**: accessing the AI toolbox.
- **Chapter 8**: how the system to provide offering recommendations to users works.
- **Chapter 9**: how open datasets are made accessible in the Marketplace.
- **Chapter 10**: perspectives to further incentivise data sharing in the Marketplace.
- **Chapter 11**: conclusions on the Marketplace and recommendations for its improvement beyond the project.

This document revolving around the graphical user interfaces of the SEDIMARK platform, it complements Deliverables SEDIMARK_D4.2 “Decentralized Infrastructure and Access Management - Final version” [2] and SEDIMARK_D4.4 “Edge data processing and service certification – Final version” [3], which explain in greater details how the functionalities exposed in these interfaces actually work internally.

1 Introduction

1.1 Purpose of the document

This report represents the second version of the SEDIMARK's approach on what will be its data sharing platform, as the main entry point to the system from the outside world. Hence, it describes not only on how users will interact with the marketplace, but also how added features such as the Recommender system and the Open Data enabler enhance user experience. As a final version of the description of the SEDIMARK data sharing platform, it iterates over Deliverable SEDIMARK_D4.5 (submitted in December 2023) [1] to provide updated descriptions of SEDIMARK components' user interfaces.

Therefore, this document does not offer a fully functional depiction of this platform: it focuses on a high-level presentation of its constitutive components instead. More precisely, it describes the graphical user interfaces of SEDIMARK components accessible to its end users, except for the Recommender system and the open data enabler which are also described from a backend perspective.

1.2 Relation to another project work

This deliverable represents the main output that emanates from the work carried out in Task 4.5 during the second period of the project execution (from M16, January 2024, to M34, July 2025). Figure 1 depicts the interaction of the activities within WP4 and the relationships with other work packages. As may be inferred from the graph, the work presented in this report relates to the outputs of the work done in Tasks 4.1 (decentralised infrastructure and APIs for Data Spaces), 4.2 (edge data processing and sharing), 4.3 (digital identities and data confidence) and 4.4 (ethical data sharing platform and services), which output appears in Deliverables 4.2 and 4.4 (both of them issued as well in M34, July 2025). The latter deliverables complement the present report, detailing how SEDIMARK components work internally, from a backend perspective.

As a last iteration on the data sharing platform, the work presented in this deliverable represents the final output of the aforementioned WP4 tasks, built on the specifications of the SEDIMARK architecture (Deliverable SEDIMARK_D2.3 [4]) and enriched by the testing done as part of WP5 (whose procedures are described in Deliverables SEDIMARK_D5.4 [5] and SEDIMARK_D5.6 [6]).

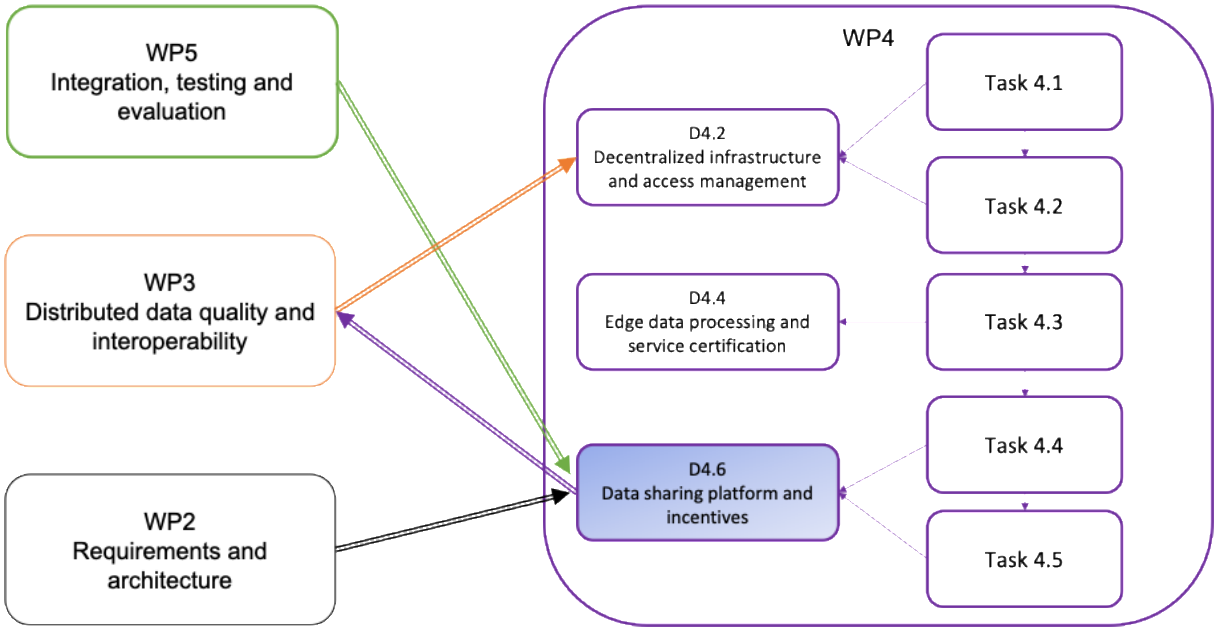

Figure 1 - Relationship between SEDIMARK_D4.3 and other deliverables, tasks, and work packages (©SEDIMARK).

1.3 Structure of the document

This document is structured in 11 major chapters:

- **Chapter 1** is the current chapter and presents the introduction to the report.
- **Chapter 2** presents an overview of the SEDIMARK Marketplace, its constitutive parts and what kind of users it may have.
- **Chapter 3** introduces the way to proceed with the user onboarding and authentication into the SEDIMARK data sharing platform.
- **Chapter 4** includes an initial view on the Catalogue that will be offered by the platform and the options the user may find upon browsing its contents.
- **Chapter 5** depicts the appearance of the Offerings that will be shared by the data sharing platform, including an overview of its registration and further management.
- **Chapter 6** delves into the data processing dashboard incorporated into the data sharing platform to make it work as desired.
- **Chapter 7** discusses the Artificial Intelligence (AI) and Machine Learning (ML) dashboard incorporated into the operational flux of SEDIMARK's data sharing platform and how it will be used.
- **Chapter 8** devotes itself to analyse the Recommender system that is a constitutive part of the data sharing platform and will help users to find information of their interest within the catalogue.
- **Chapter 9** presents an overview of the Open Data enabler, its projected architecture and the kind of data offerings it contributes to the SEDIMARK ecosystem.
- **Chapter 10** establishes how the sharing process will take place as well as its implications.
- **Chapter 11** presents the conclusions of the report debating the main outcomes and the perspectives offered by the platform.

2 Marketplace overview

2.1 Scope and personas

The conception of SEDIMARK's Marketplace implies for it to act as the entry point to SEDIMARK functionalities and resources. Its architecture joins together a vast series of cooperating services and tools that provide various functionalities. This means the developments provided by technical work packages can be deployed independently and expose their functionalities through the marketplace itself.

To do so, the SEDIMARK Marketplace relies on a graphical user interface (GUI) that displays such set of services and offers users an easy access to log in or out of the application, perform the registration of both Participants and/or Offerings, carry out Contract negotiations, receive recommendations based on their profiles and much more actions that will be covered in further sections of this report. Marketplace users are expected to present diverse profiles, corresponding to various usage of the Sedimark platform resources (catalogue, toolbox, …). They fall in two groups:

- **Participants**: corresponding to users who registered to SEDIMARK and have been approved by the administrators. Participants can be *Providers* or *Consumers* of Offerings in the Marketplace Catalogue.
- **Visitors**: referring to non-registered users. They can only view the public Offerings of the Catalogue, and request to register in the SEDIMARK ecosystem.

Table 1 below captures a brief description of what each one of those kinds of users could do in the platform through the depiction of basic user stories.

Table 1 - SEDIMARK marketplace user stories

Ref. No.	As a (stakeholder)	I want to	So that	And is considered 'done' when
SM-US1	Visitor	Register an account in SEDIMARK Marketplace.	I may access the SEDIMARK Marketplace resources (provide/consume Offerings).	I receive a DID and a Verifiable Credential from the Identity Manager component.
SM-US2	Participant	Check my user account information role and permissions.	I can manage my user account and check the permissions granted.	I can check my DID document and my participant Verifiable Credential.
SM-US3	Participant	Decide what personal data I am exposing to other participants.	I keep sovereignty over my personal data.	I can view and edit my personal data, hosted on my premises only.

Ref. No.	As a (stakeholder)	I want to	So that	And is considered 'done' when
SM-US4	Participant, Visitor	Browse the Catalogue of Offerings	I can see datasets or services corresponding to my permissions.	I can access / search / filter the Offerings in the Catalogue via the marketplace GUI.
SM-US5	Participant (Provider)	Register a dataset Offering	I can put make my Offering accessible to other Participants, with the access and usage policies of my choice.	My Offering is accessible in the Catalogue, and its usage policies are enforced.
SM-US6	Participant (Consumer)	Purchase a dataset Offering	I can access the content of the Offering.	I can transfer the data of the Offering to a destination of my choice.
SM-US7	Participant (Consumer)	Get recommendations of Offerings	I can quickly get informed of Offerings matching my interests.	The Marketplace provides me with such recommendations upon browsing the Catalogue.
SM-US8	Participant (Provider)	Monitor the status of my provided Offerings	I can keep track of my contracts, active or expired, and get statistics for all of them.	I can access a dashboard listing all the Contracts corresponding to Offerings I provide.
SM-US9	Participant (Consumer)	Monitor the status of my consumed Offerings	I can keep track of my Contracts, active or expired, and get statistics for all of them.	I can access a dashboard listing all the Contracts corresponding to Offerings I consume.
SM-US10	Participant	Browse the transfer history of my active Contracts	I can monitor the transfers from/to my data sources/sinks.	I can view the transfer history of my contracts in the Marketplace GUI.

Ref. No.	As a (stakeholder)	I want to	So that	And is considered 'done' when
SM-US11	Participant (Consumer)	Download data of my purchased Offerings.	I can use the datasets for any purpose I want according to the Contracts' policies.	I can view my consumed Offerings and download the data from the GUI.
SM-US12	Participant	Access the data processing dashboard	I can use the SEDIMARK data processing toolbox.	I can access the data processing toolbox from the Marketplace GUI.
SM-US13	Participant	Access the AI dashboard	I can use the SEDIMARK AI toolbox.	I can access the AI toolbox from the Marketplace GUI.

2.2 Architecture

In here readers will find a succinct description of the components which comprises the SEDIMARK Marketplace, constituting the backbone of its data sharing platform. As described in Deliverable SEDIMARK_D2.3 "Architecture and Interfaces. Final version" [7], this is one of the constitutive parts of the overall SEDIMARK architecture and relates to the items presented in the green box in Figure 2.

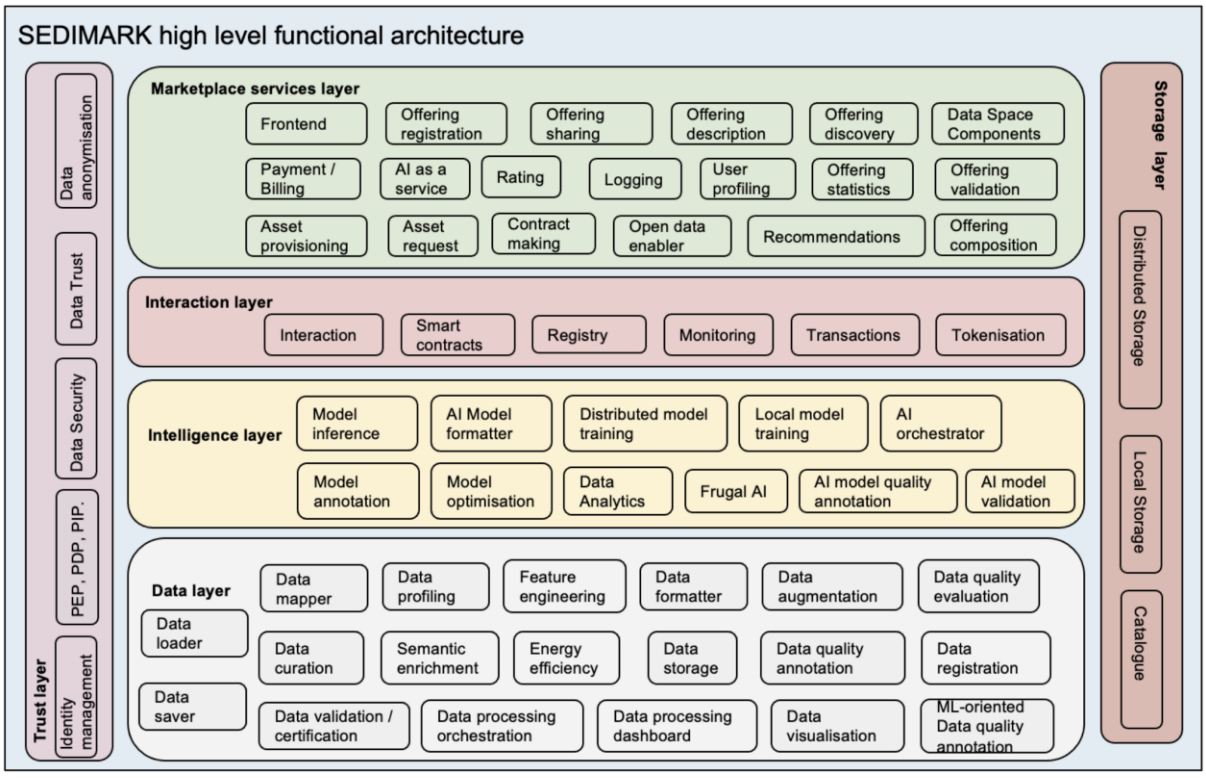

Figure 2 - High level view of the SEDIMARK marketplace architecture (from Figure 9 in SEDIMARK_D2.3 [7] (©SEDIMARK))

Even though the picture does not delve into the fine details on how each component in such Marketplace services layer interfaces with the rest of them, neither goes into the close definition of connections, the descriptions coming in following chapters of the report make explicit how they all fit together, as well as how they interact with the other layers of the overall SEDIMARK platform. In fact, and starting from the premises established in SEDIMARK_D2.3, it is possible to divide the services layer into two parts, namely: Offering management and Marketplace.

Hence, within the former group the modules included go by the naming:

- Offering description to create and/or edit descriptions of data and services apt to be exchanged.
- Offering registration to perform the embarkment of data/services into the local Catalogue and confirm they comply with the SEDIMARK rules.
- Offering discovery to let users navigate through the Marketplace Catalogue and find the data and services closest to their interests.
- Offering sharing to ease the way data and services will be distributed.
- Global Catalogue to compile the complete collection of Offerings.
- Open Data enabler to make available in the SEDIMARK Marketplace datasets coming from diverse open data portals.

While as the latter cluster embarks services such as:

- Marketplace GUI offers the window into SEDIMARK from the outside world.
- Logging establishes the protocol to let registered users (Providers, Consumers, Administrators) get into the platform.
- Contracting sets up the policies to proceed with the acquisition of a certain Offering by an interested user.
- User profiling keeps a log of users' activity within the Marketplace to provide relevant input to the Recommendation service.
- Service provisioning to make resources available to conduct a specific activity.
- Service request eases the procedure to present a formal request for certain service to be provided.
- AI as a Service interacts with the Intelligence layer to interface with the AI orchestrator with the aim to let users perform machine learning tasks on their Offerings or as a service.
- Payment connects with the corresponding gateway to complete the transaction once a contracting takes place.
- Recommendations suggest users particular Offerings that may be of their interest.

As anticipated, the rest of sections in this report cover the main areas of the SEDIMARK Marketplace and offer detailed views on them all.

3 Onboarding and authentication

In this chapter, we describe the home page of the Marketplace, as well as how Participants can log in their account and the onboarding procedure for new users.

3.1 Home page

As a showcase of SEDIMARK's Marketplace, the home page's targeted audience is primarily prospective users and early adopters. Consequently, its main role is to provide an overview of the activities in the Marketplace and encourage visitors to browse the Offering Catalogue. It is also the perfect place to display some news about SEDIMARK's use cases or more broadly about recent developments or outreach of the project.

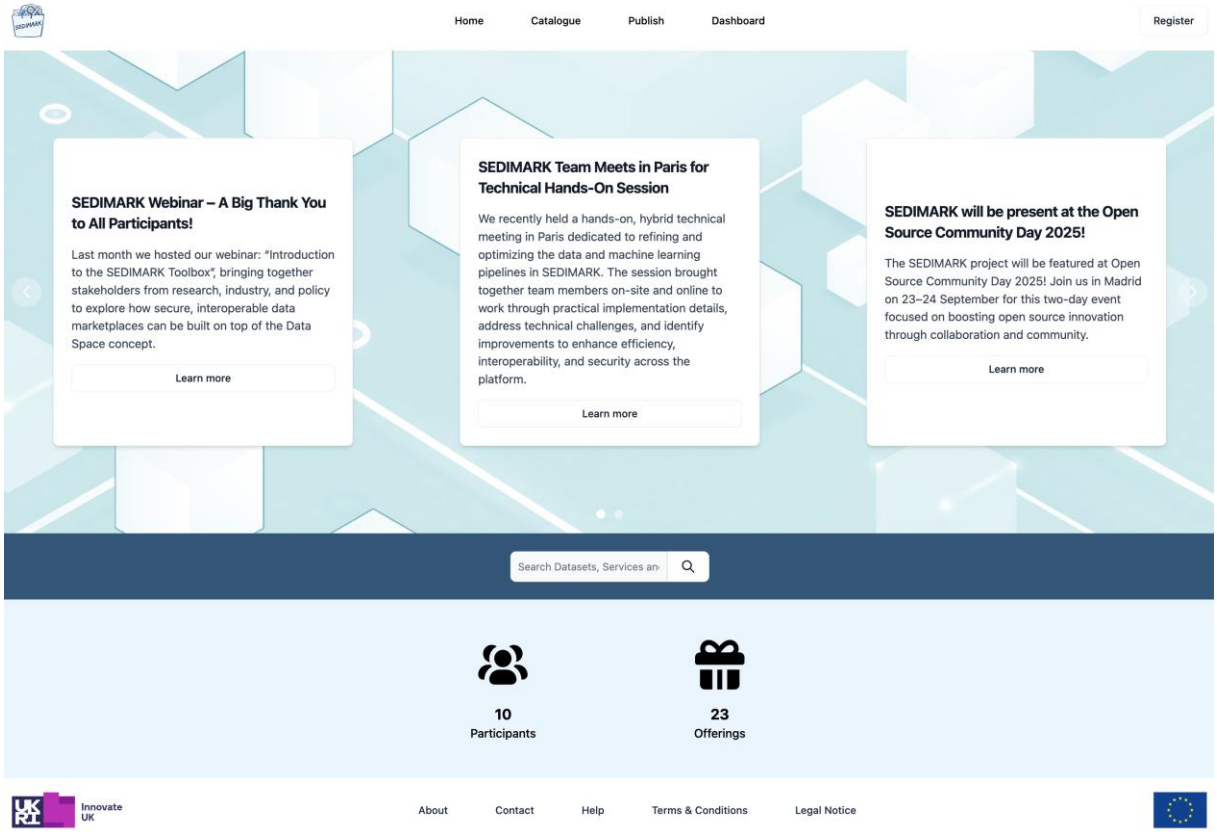

Figure 3 - Marketplace home page (©SEDIMARK)

The Marketplace home page shown in the figure above answers all these calls by showcasing SEDIMARK's latest insights through a carousel. Each of its entries, represented by cards, point towards an article posted in the official SEDIMARK website [8]: these entries can refer to any relevant resources about SEDIMARK, such as blogposts of consortium partners, social media article or scientific publication.

In addition to its possible access via the top navigation bar, a quick search input has been placed in the centre of the page to foster Catalogue browsing by prospective users. This search is kept as simple as possible: it just requires a text input to match with Offering names or descriptions. This search can further be refined by filtering based on Offering providers or keywords.

Finally, the dynamism of the Marketplace is emphasized using a set of quantitative facts: number of Participants in the SEDIMARK ecosystem, as well as the total number of Offerings in the catalogue.

3.2 Participant registration

New users wishing to join the SEDIMARK ecosystem can do so in three simple steps:

1. Creating an identity
2. Providing details to create their corresponding decentralized identifier (DID).
3. Receiving their DID and verifiable credentials.

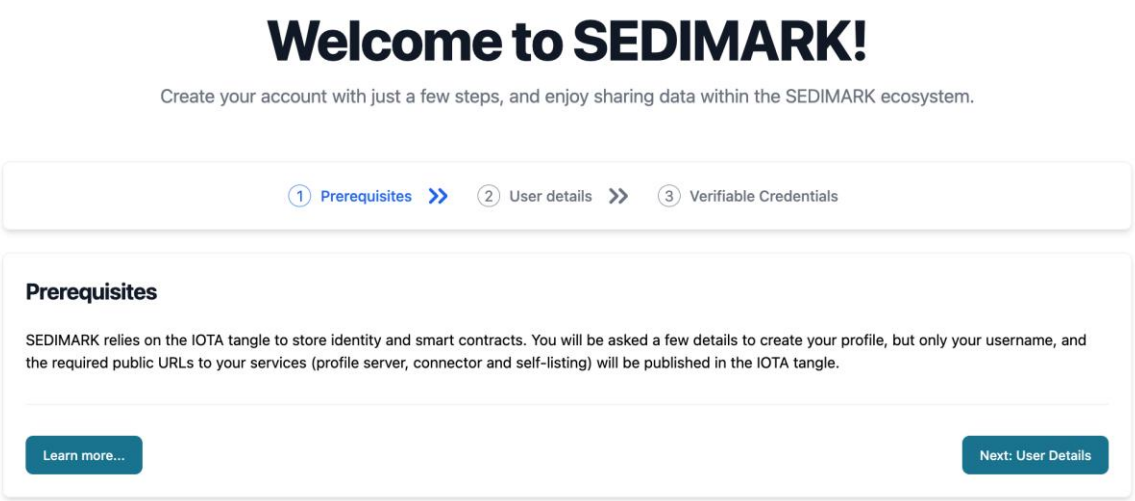

Figure 4 - Marketplace registration form, step 1: prerequisite (©SEDIMARK)

Becoming a SEDIMARK participant only requires the creation of a decentralized identifier (DID) in the IOTA Distributed Ledger Technology (DLT): more information about the identity creation can be found in Deliverable SEDIMARK_D4.2 [2]. However, the onboarding form not only guides the user through the necessary steps to create this DID, but it also allows them to provide more data to customize their profile visible to other users. Therefore, the first step of the onboarding form simply informs users about what the data they provide will be used for. More precisely, user inputs will be stored in two locations:

- **In the DID document and verifiable credentials**: since these resources are public, no personal data are stored in them. Only the username chosen by the new participant, as well as the URLs to the resources she/he must expose publicly to be able to interact with other participants.
- **In the profile server**: it contains all personal data users may want to provide to customize their profile. This server is a component deployed by SEDIMARK participants to host such data on their premises, therefore keeping a total control on their exposure.

Once the user acknowledged the prerequisites, the next step asks him/her to provide a few more details for two purposes: the DID creation and the profile customisation. Since a DID is created for all new users, these data are mandatory and subsequently marked by an asterisk (*): only the username and the URL to the resources the new user must expose are required. This URLs corresponds to:

- **The user's self-listing**: i.e. where all her/his provided offerings can be acquired.

- **The user's connector**: to enable contract negotiation and offering consumption.
- **The user's profile server**: to expose some data he/she is willing to share with other partners. It can be seen as a business card in the ecosystem.

All other data, corresponding to the profile of the user, are optional.

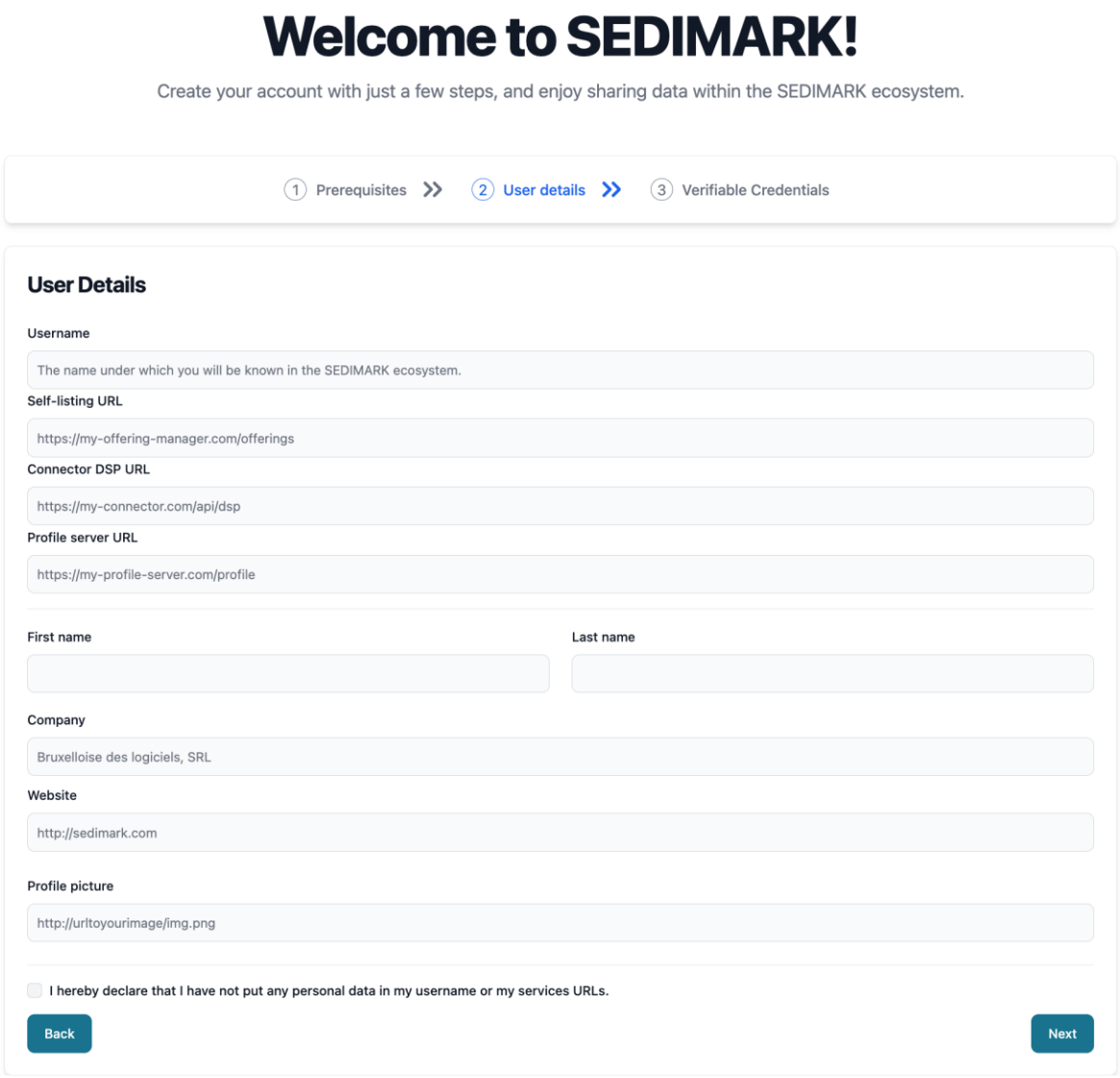

Figure 5 - Marketplace registration form: user details (©SEDIMARK)

Once the user has completed this step, her/his DID gets created and his verifiable credentials as a participant issued and stored in DLT booth instance. For the sake of transparency, users can review their verifiable credentials. Upon hitting the *Go to marketplace* button, users get redirected to the home page.

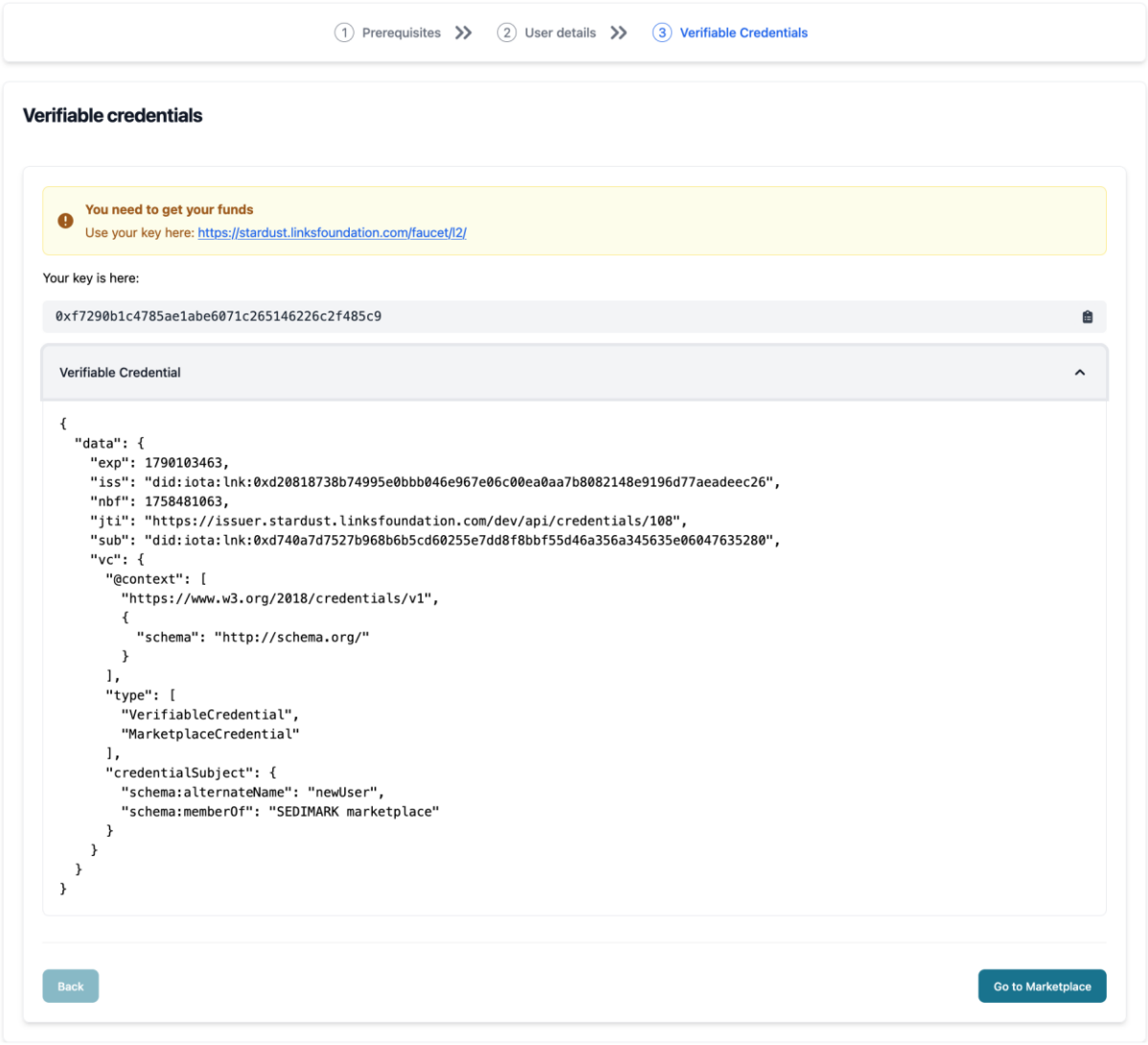

Figure 6 - Marketplace registration form: getting verifiable credentials (©SEDIMARK)

After subsequent connections to the marketplace, onboarded participants cannot see a *Register* button anymore in the navigation bar. Instead, they see an avatar button showing their username. The marketplace automatically checks their identity in their DLT booth instance, therefore removing the need for users to sign in or out. Their DID document and verifiable credential can be accessed at any time upon clicking the avatar icon.

4 Catalogue browsing

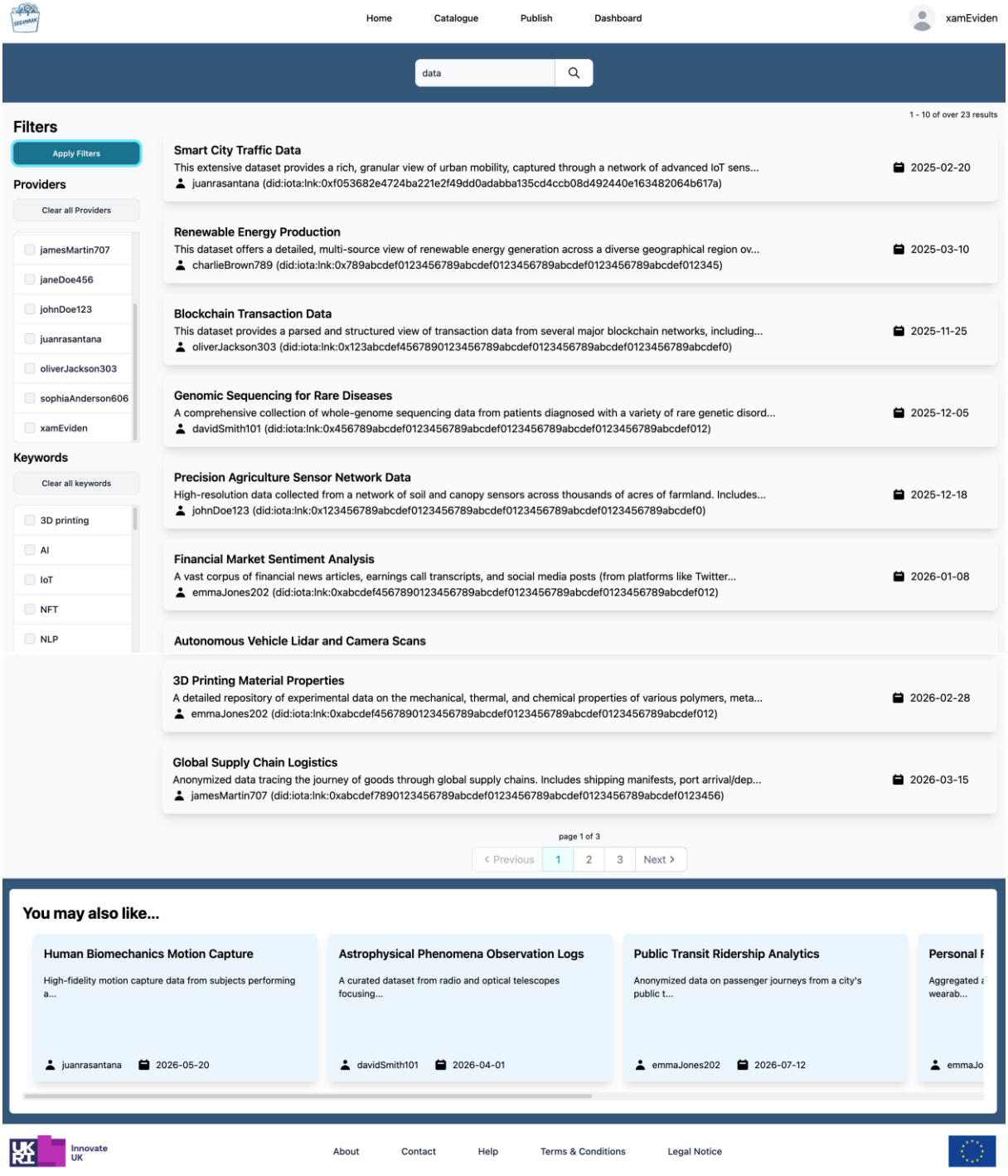

Figure 7 - Marketplace Catalogue browsing page (©SEDIMARK)

This chapter introduces the Catalogue of Offerings which can be browsed in the Marketplace. It gathers all Offerings provided by all SEDIMARK participants. The reader willing to know more about how the Catalogue gets populated can refer to Section 4.4.2 of Deliverable SEDIMARK_D4.2 [2]. Any user, being an authenticated Participant or a simple visitor, can

access the Catalogue, either by performing a quick search in the home page, or by hitting the *Catalogue* button in the navigation bar.

As shown in Figure 7, the Catalogue browsing page consists of a simple list of Offerings, displayed as one Offering per row using the top left button bar. The number of Offerings displayed per page can be customized in the Marketplace's settings (10 by default). As for the onboarding, we strive to keep the interface as simple as possible, aiming at providing a similar user experience as the one of other online Marketplaces such as eBay or Amazon.

The search bar is the same as the one in the home page: simply requiring some words to be matched with the Offering title or description. The side bar enables users to further filter or sort the results of the search, by default sorted by creation date (most recent first). Users can shrink the results list by selecting only the providers or keywords relevant to them.

At the bottom of the result list, a horizontal list of recommended Offerings is displayed. It originates from the Recommender system, based on the user's search query (text to match to title or description, as well as selected filters). The number of recommendations can be adjusted in the Marketplace's settings, defaulting to 5.

Each Offering in the result list is concisely described by its title, a short description phrase and a set of facts associated with icons, such as its provider and creation date. A more detailed description can be accessed by clicking on the Offering entry.

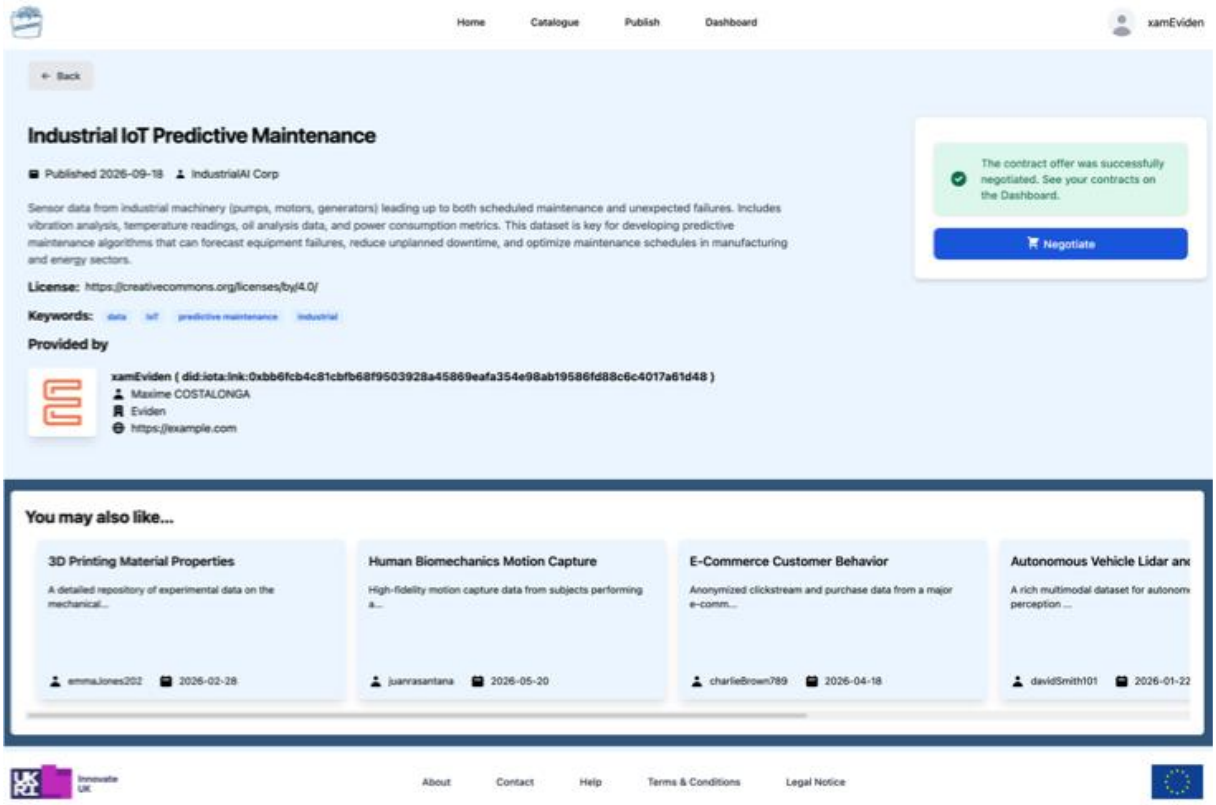

Figure 8 - Example of an Offering description page (©SEDIMARK)

Upon selecting an Offering from the results, the user is redirected towards a page detailing it, an example of which is shown on Figure 8. It expands the description of the Offering, and provides its exhaustive list of keywords, as well as the license of its dataset (if provided). A separate card on the left side, following the user as she/he scrolls through the description of

the Offering, contains a button triggering a contract negotiation between the user and the Offering provider. At the end of this process, a message appears to inform the user of the result of the negotiation (more details about how the negotiation works can be found in section 4.4.1 of Deliverable SEDIMARK_D4.2 [2]). Below the Offering description, more information about its provider is displayed. At a minimum, only the username and the DID of the Provider is shown. However, any other data shared by the provider through its profile server will be shown as well if available, such as her/his full name, company and home page.

Finally, a list of recommended Offerings is also displayed, based on similarity they share with the currently selected one. If offers a convenient way for users to browse the SEDIMARK catalogue based on their preferences.

5 Offering provision and consumption

This chapter describes how SEDIMARK Participants can create, consume and manage their Offerings in the Marketplace.

5.1 New Offering registration

An authenticated Participant can publish a new Offering in the Catalogue using the *Publish* button in the navigation bar. She/he is then redirected to a form consisting of the following steps:

1. Defining the Asset: to indicate whether the Offering should reuse an existing Asset or not, and provide a high-level description of its content.
2. Setting the access to the Asset: more precisely, to parametrize the query that will allow the user's Connector to fetch the Asset.
3. Associating some policies to the Offering.
4. Reviewing and submitting the Offering.

We will now review how each of these steps work to achieve the publication of an Offering.

5.1.1 Asset definition

The first step to the publication of an Offering is to define the Asset the Offering is about. The user is offered the possibility to reuse an existing Asset or to create a new one from scratch (see Figure 9). Existing Assets refer to the ones that may be present in the NGSI-LD Context Broker of the user, if he/she uses the SEDIMARK toolbox, either because she/he manually created them, or because it was generated by other SEDIMARK tools such as the AI orchestrator. More information about how such Assets are generated can be found in Deliverable SEDIMARK_D4.4 [3]. If the user selects an existing Asset, its description and access type will be pre-filled directly with the data fetches from the NGSI-LD Context Broker.

When creating a new Asset from scratch, the user is required to provide some high-level description of the Asset, a title and at least one keyword. These will primarily be used to populate the Offering description prospective Consumers will access from the Catalogue. Other optional information can be provided, such as the creator of the dataset (which can differ from the publisher) or a picture to illustrate or document the Asset.

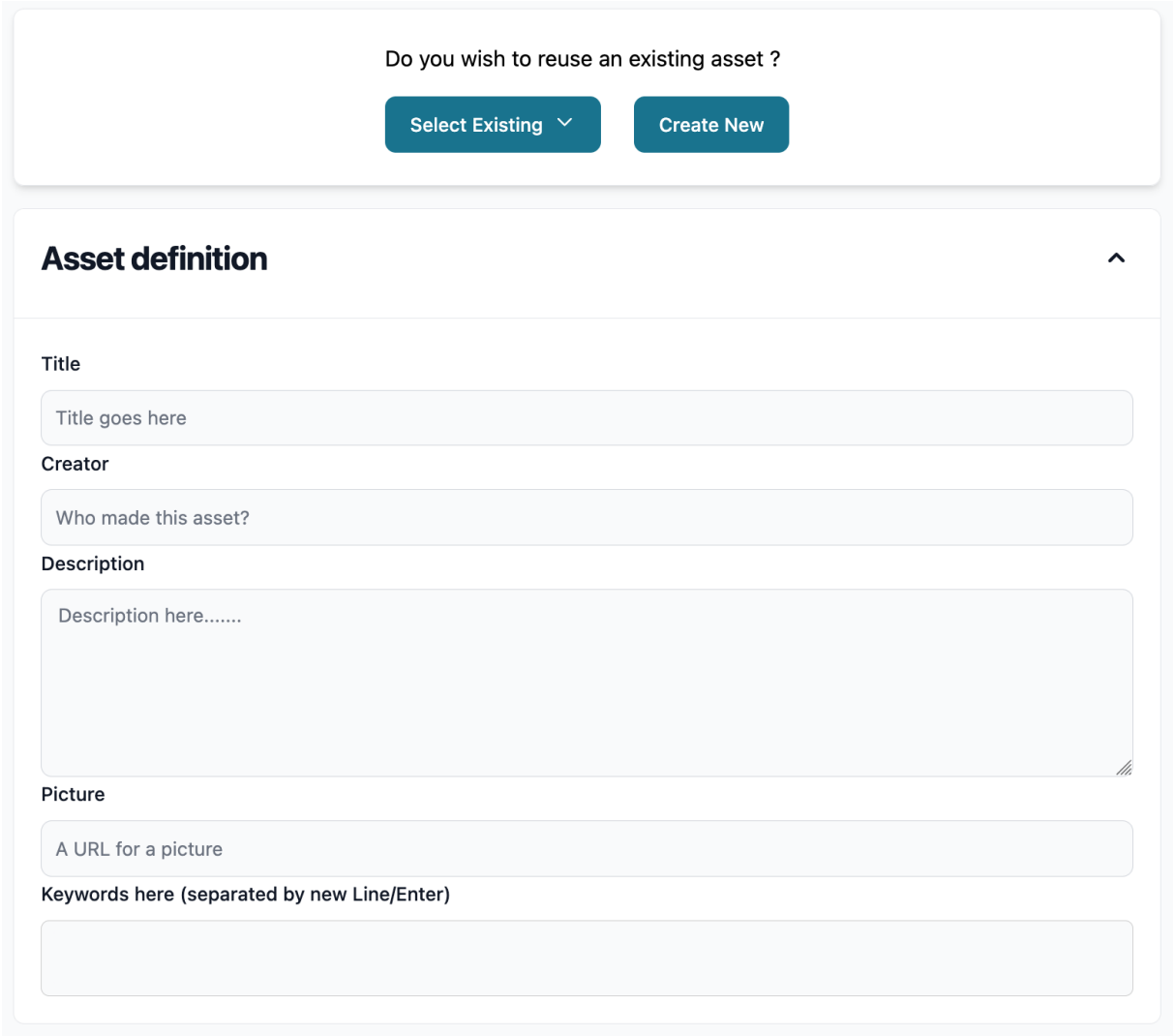

Figure 9 - Marketplace Offering publication form: Asset definition (©SEDIMARK)

5.1.2 Asset access

After the high-level description of the Asset, the user needs to indicate its location, to enable her/his connector to access it, as shown in Figure 10. SEDIMARK currently supports only HTTP requests to fetch the Asset, so the user is asked to provide the URL pointing to the location of his/her dataset, and can optionally add headers to this request (for instance to indicate the format of the data, or provide an API key or token if the request endpoint is protected).

Users can also optionally associate a license to their datasets, by providing either a license name, URL to the license terms, or the license terms themselves.

To finally turn the Asset into a publishable Offering in the SEDIMARK Catalogue, users can set policies ruling its usage. Currently, the Marketplace only allows a time period policy, i.e. enabling users to select a period of time during which their dataset is accessible to other participants. However, the Marketplace automatically adds other policies before publication, to ensure that:

- Only members of the SEDIMARK ecosystem can consume the Offering.

- The Consumer has acquired the Offering in the Marketplace (i.e. obtained a data token from the DLT corresponding to this Offering).

Since policies are using the Open Digital Rights Language (ODRL) [9], the Marketplace user interface can easily be extended to support other types of policies (such as restricting the access to specific participants or in designated locations).

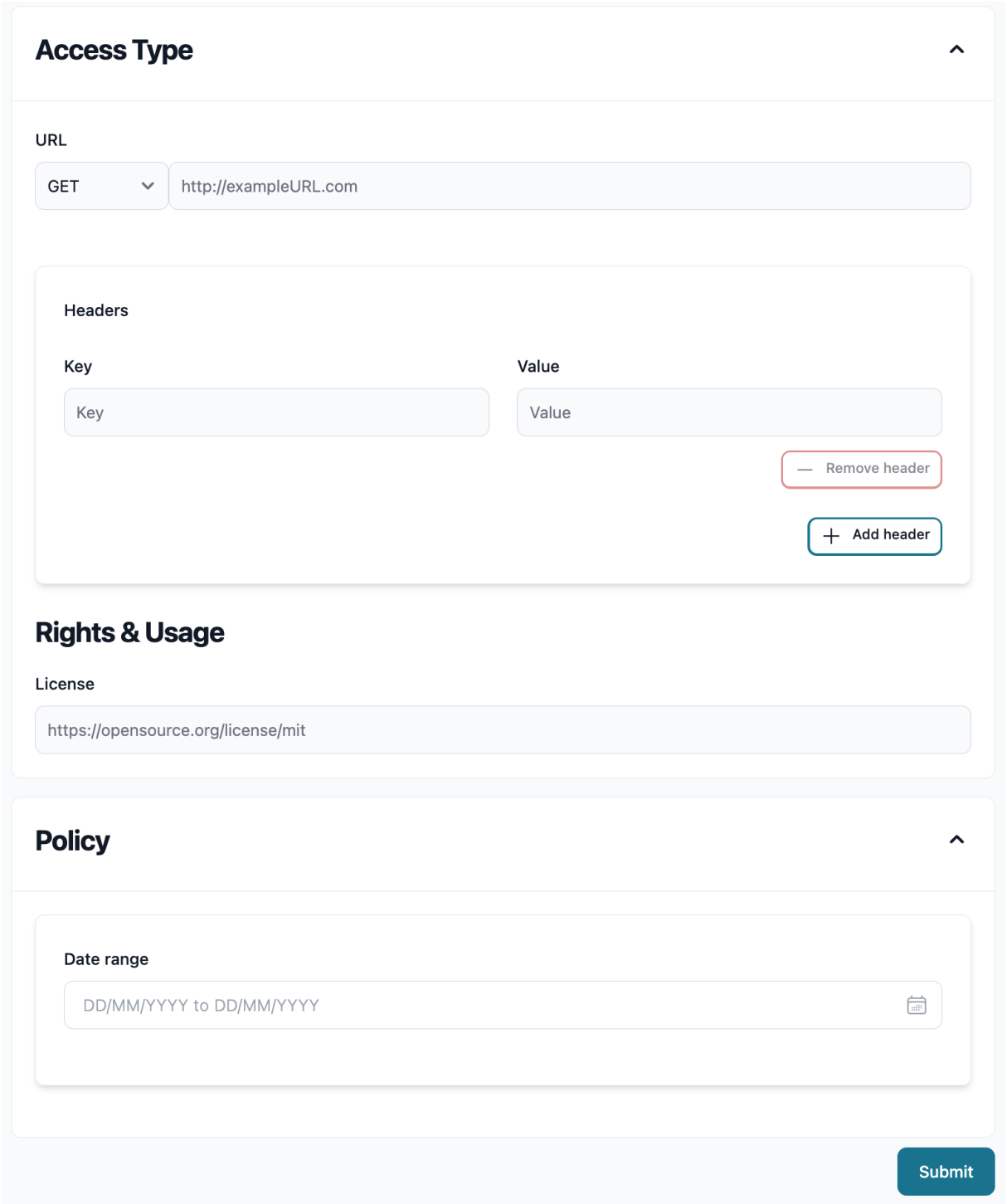

Figure 10 - Marketplace Offering publication: access & policies (©SEDIMARK)

Once done setting validity period of her/his dataset, the user can move on the last step: reviewing the Offering and submitting it for publication in the Catalogue. Since this step simply shows the Offering as it should appear online, alongside a submit button, it is not shown here. Such a preview of a Catalogue Offering can be checked by the reader in section 4.

5.2 Offering management dashboard

The *Dashboard* button located in the navigation bar enables authenticated users to access, at any time, a dashboard to manage their consumed and provided Offerings. This dashboard is composed of 3 tabs, accessible from a side bar:

- **Overview**: for quick facts about the participant's usage of the platform.
- **Offerings**: for the participant to manage the Offerings she/he published in the SEDIMARK Catalogue.
- **Contracts**: where all contracts involving the user are listed, as a provider or a consumer.

The following sections review in detail each of these pages.

5.2.1 Overview

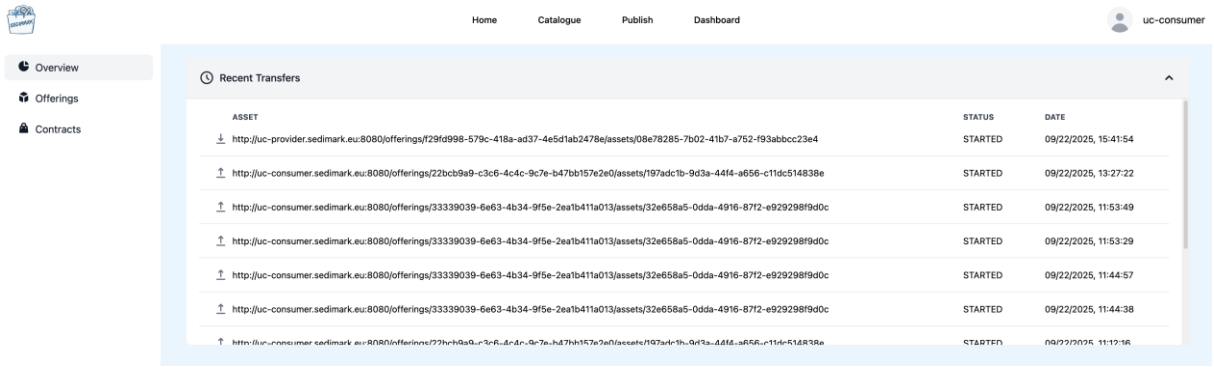

Figure 11 - Marketplace Offering management dashboard: overview (©SEDIMARK)

As shown in the figure above, the overview section simply lists the most recent data transfers in which the user is involved, therefore indicated the most recent activities of the contracts the user is part of. The download/upload icon in a transfer entry indicates whether the user is a consumer/provider of the Asset.

5.2.2 Offerings

This part of the Offering management dashboard enables users to review and edit their provided Offerings. The latter are displayed as a list, in which each entry is succinctly described by its title, identifier, creation date and dataset endpoint URL, as shown in Figure 12. Each entry can be expanded to reveal the Asset full description, as well as its license and keywords. Each Offering can be deleted, resulting in a subsequent removal from the SEDIMARK Catalogue.

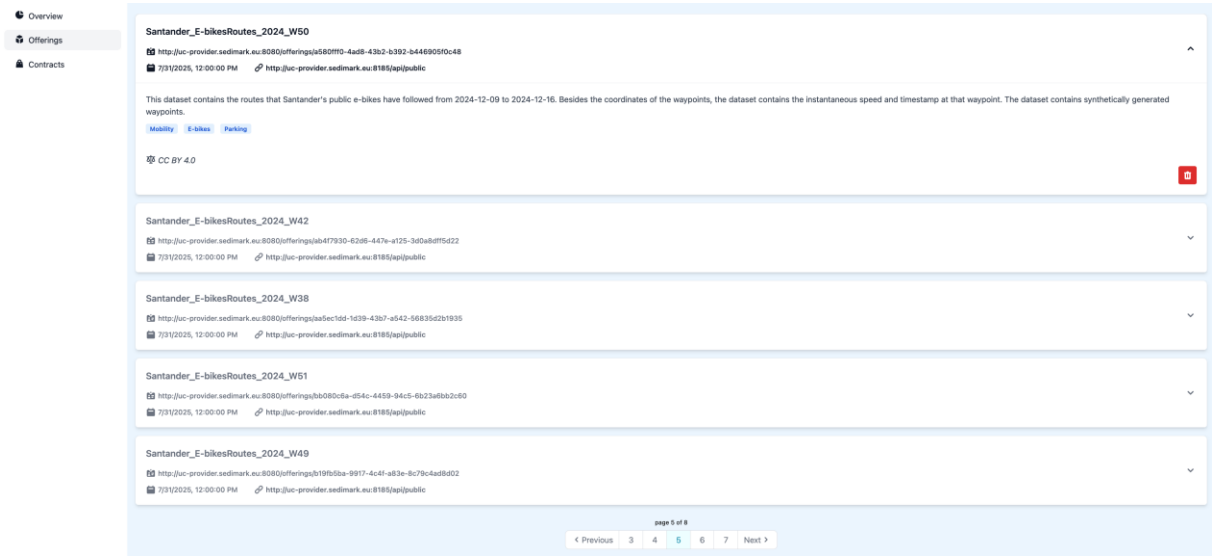

Figure 12 - Marketplace Offering management dashboard: Offerings (©SEDIMARK)

5.2.3 Contracts

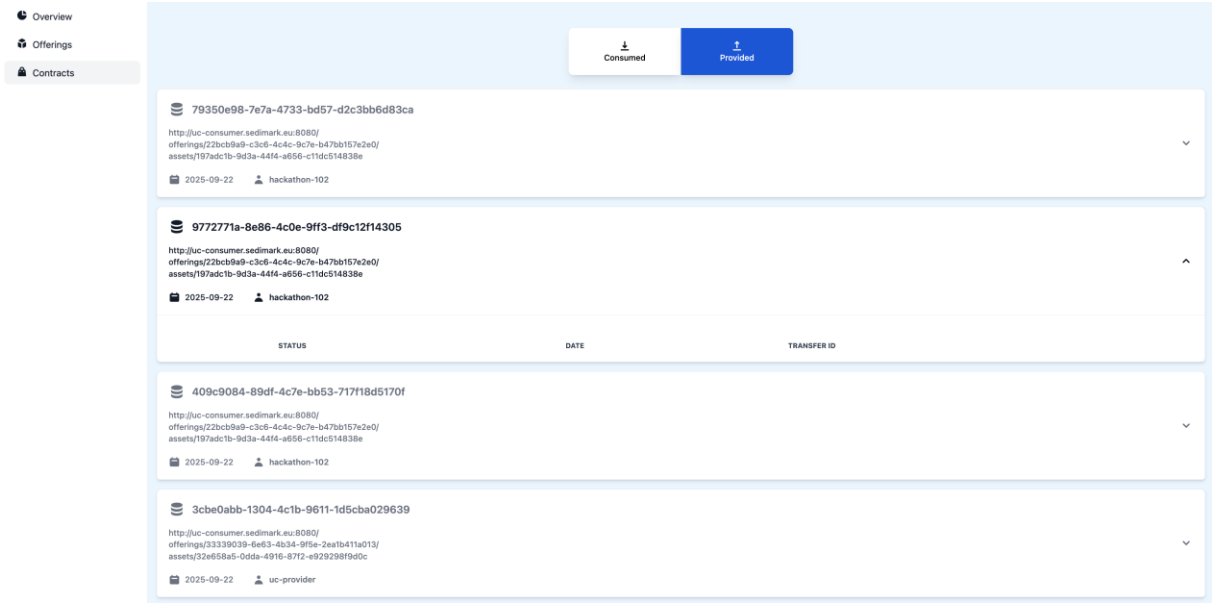

Figure 13 - Marketplace Offering management dashboard: Contracts (provided) (©SEDIMARK)

The Contracts management dashboard is very similar to the asset one described previously. It also consists of a list of Contracts, but the latter is split in two parts, to separate consumed contracts from provided ones (see Figure 13). For each contract, the user can access its ID, the Asset it refers to, as well as the creation date and the username of the other party involved in the contract. The list of most recent transfers for the given contract is shown upon selection.

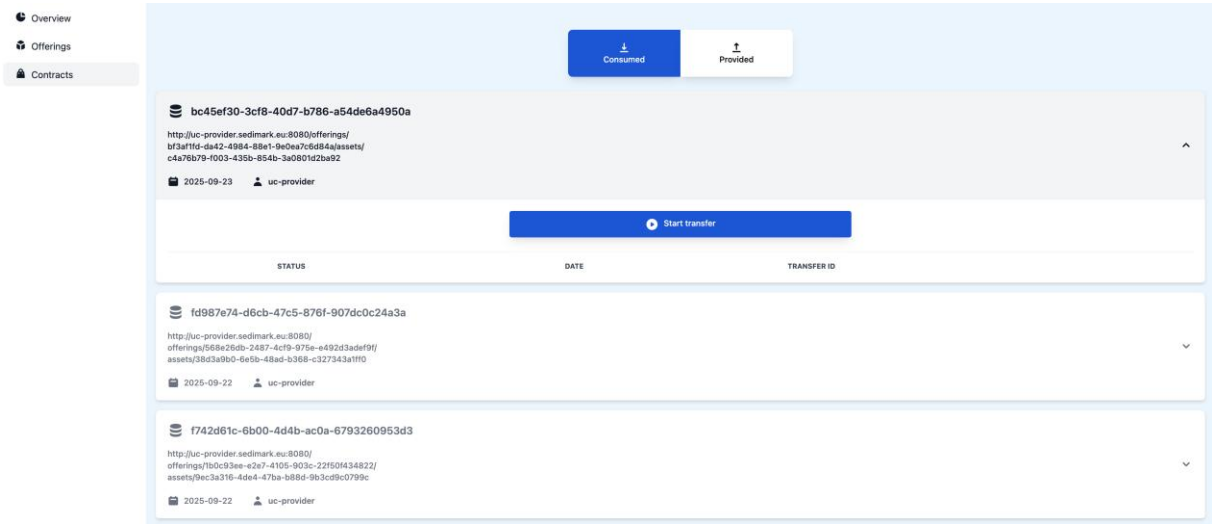

Figure 14 - Marketplace Offering management dashboard: Contracts (consumed) (©SEDIMARK)

For contracts where the user is a consumer, expanding an entry also reveals a *Start Transfer* button to initiate the acquisition of the dataset wrapped in the Offering (see Figure 14). As shown in Figure 15, it opens a modal suggesting two ways for the user to acquire the data:

- **Push mode**: the user provides an endpoint URL, where the dataset will be uploaded to by her/his Connector.
- **Pull mode**: this mode requests the provider's Connector to expose the data, so the user can download it himself/herself. The provider's Connector answers by giving a public URL, secured by an access token given only to the user.

The reader willing to get deeper into how data transfers are operated can refer to section 4.4.1 of Deliverable SEDIMARK_D4.2 [2].

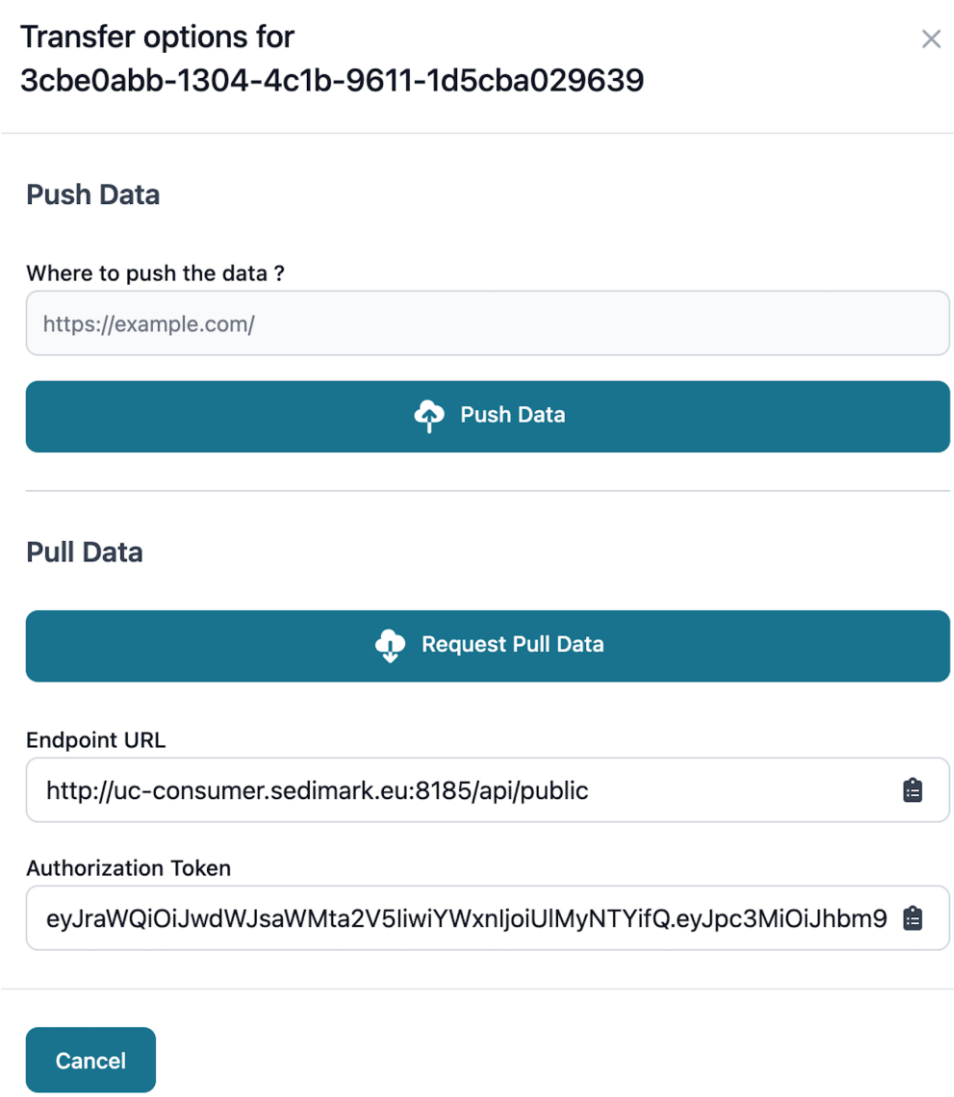

Figure 15 - Modal to select a data transfer type when consuming an Offering (©SEDIMARK)

6 Orchestrator UI

This chapter introduces the Orchestrator UI, initially described in Chapter 6 'Data Processing Dashboard' of Deliverable SEDIMARK_D4.5 [1]. The Orchestrator UI is part of the Data Processing Orchestration (DPO) and serves as a key component of the SEDIMARK ToolBox, enabling data consumption, post-processing, data manipulation, machine learning techniques, and asset preparation. The general architecture of the SEDIMARK ToolBox is illustrated in Figure 16.

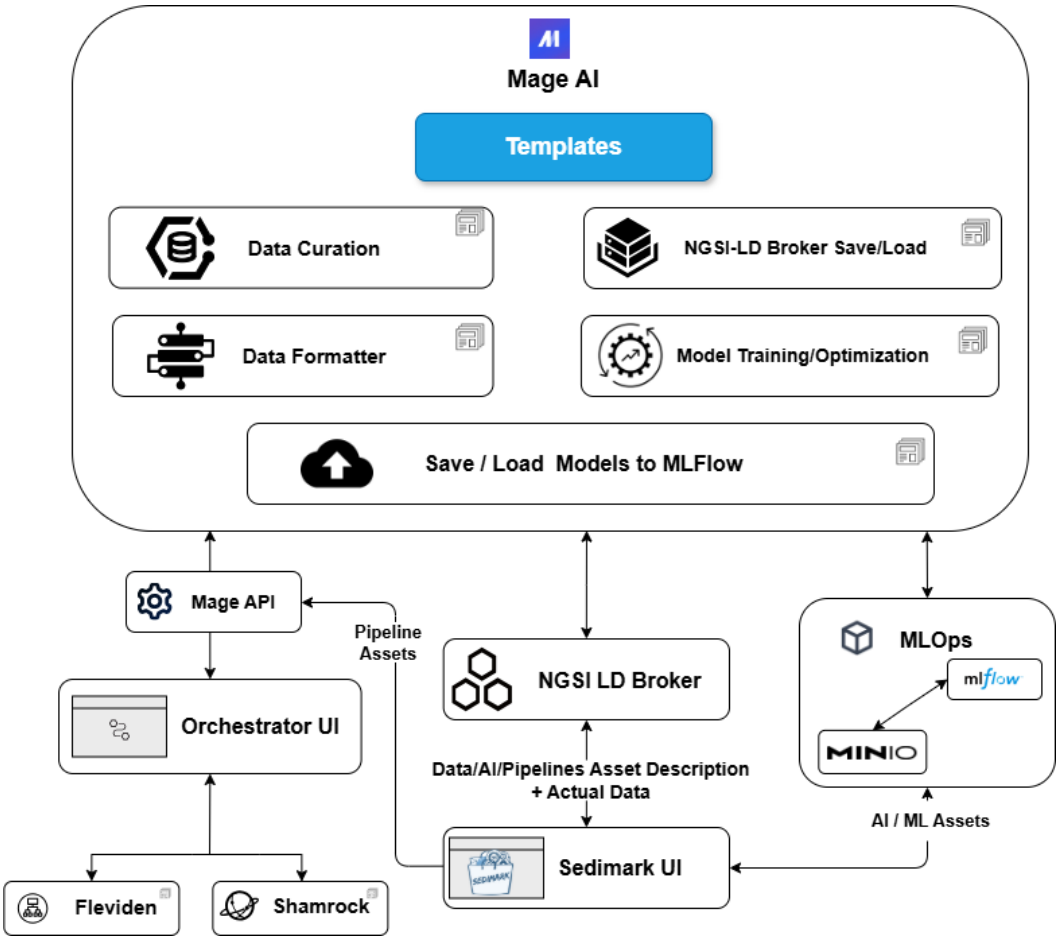

Figure 16 - SEDIMARK ToolBox architecture depiction (©SEDIMARK)

The DPO, described in more detail in subsection 3.3 'Data Processing Orchestration' of Deliverable SEDIMARK_D3.2 'Energy efficient AI-based toolset for improving data quality' [10] represents a multi-stage data processing architecture designed to manage the flow, transformation, and storage of data across various system components, such as:

- **Orchestrator UI**: application for the end user interaction
- **MageAI**: main engine for the definition, triggering and execution of ETL pipelines
- **MageAPI**: MageAI wrapper that provides security and provisioning
- **NGSI-LD Broker**: assets metadata and data assets curation
- **MLflow and MinIO**: AI models assets and workflows curation and storage
- **Fleviden and Shamrock**: Federated Learning (FL) service assets system configuration

The Orchestrator UI is designed as a user-friendly platform accessible to both technical and non-technical users. The interface is built to emphasizes accessibility with a clean layout that organizes related information using colour coding and icons for quick identification. An illustration of the Orchestrator UI is depicted in Figure 17 and is organized into two main sections:

- Operations represented on the left side by a vertical menu interface and composed of interaction elements that enable the management, creation, configuration and exporting of workflows.
- Dashboard represented on the right side of the interface and is the main visualization for workflow enabling execution, variables configuration and monitoring.

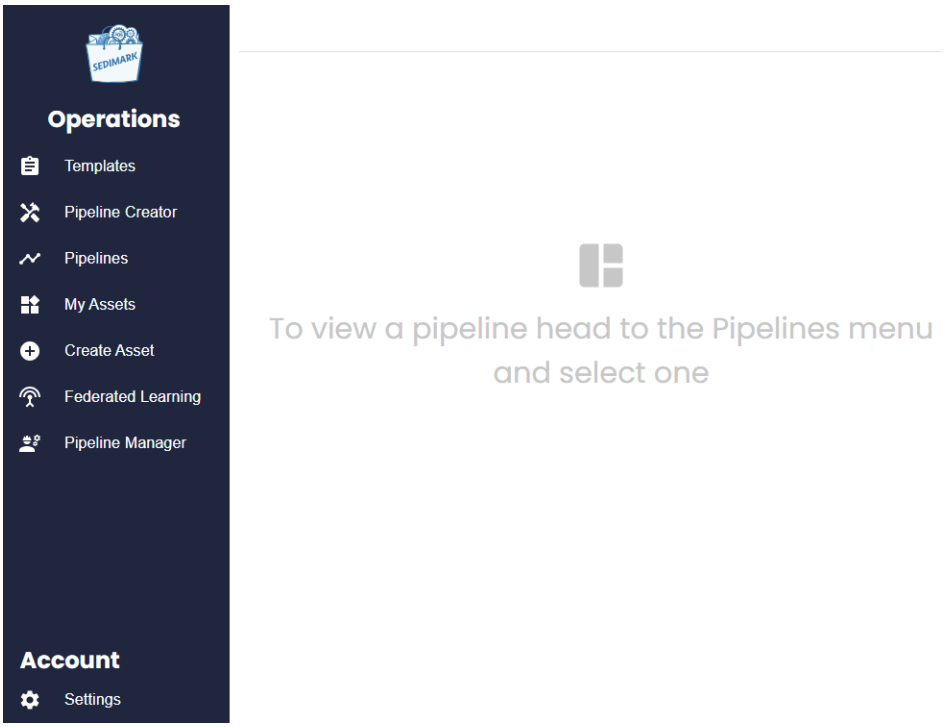

Figure 17 - Orchestrator UI application interface (©SEDIMARK)

The Operations menu enables users to create new workflows either from Orchestrator built-in workflows found under the Templates section or by building them manually from blocks using the Pipeline Creator functionality. Furthermore, existing workflows can be loaded into the Dashboard using the Pipelines interaction.

On the Dashboard side, multiple instances of pipelines can be run and managed in parallel through multiple tabs, thus enhancing efficiency. Moreover, for each rendered workflow, users can access logs at the block level and across all workflows runs, along with hardware-related performance metrics. A representation of a pipeline visualization displayed on the Dashboard of the Orchestrator UI is depicted in Figure 18, and shows a three-block workflow namely the Anomaly Annotator. Each block of the representation comprises user configurable variables and display logs option.

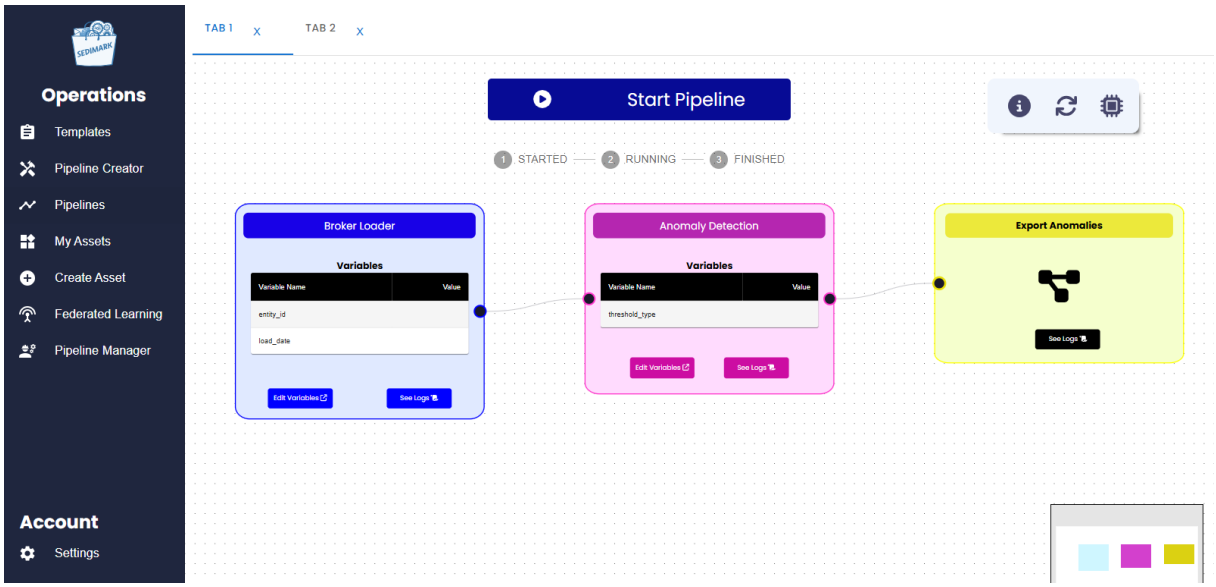

Figure 18 - Orchestrator UI Workflow visualization (©SEDIMARK)

The Pipeline Creator option enables users to manually create workflows in the AI Pipelines Studio sandbox as illustrated in Figure 19. The workflows can be constructed either by using the list of predefined built-in blocks or by generating them on the fly using the RAG (retrieval augmented generation) method. There are three types of blocks that can compose a workflow colour-coded for identification purpose: Data Loaders displayed in blue, Transformer in pink and Data Exporter in yellow.

The Automatic Block generation RAG tool options are illustrated in Figure 20. Using this interface, users can load, save as template, edit, visualized the output code and input the prompt required for generating the workflow block.

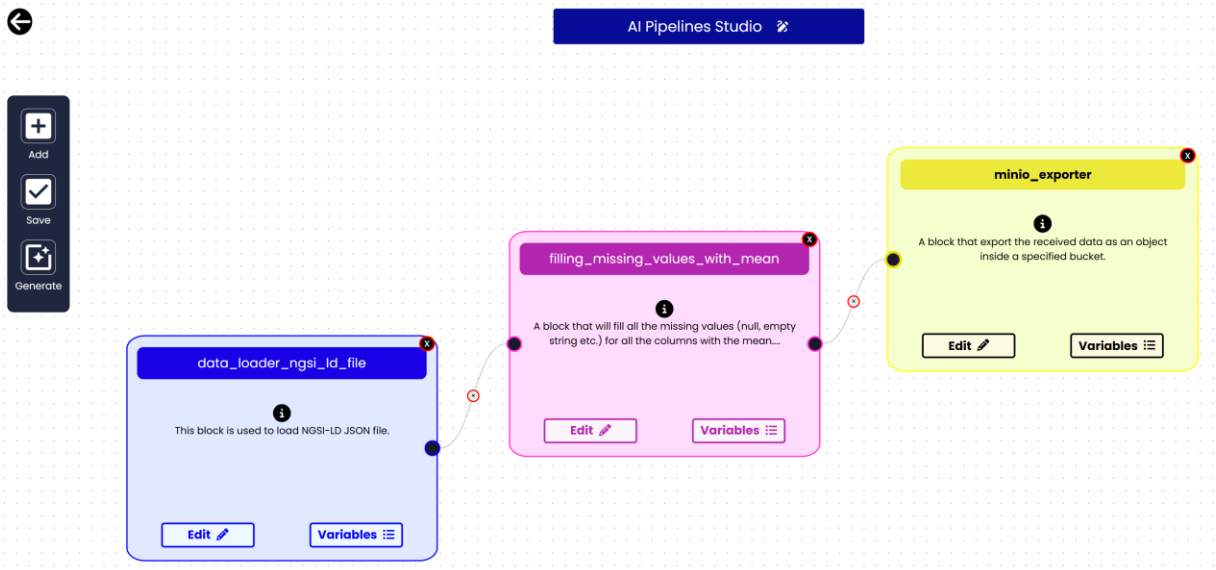

Figure 19 - Orchestrator UI Workflow builder (©SEDIMARK)

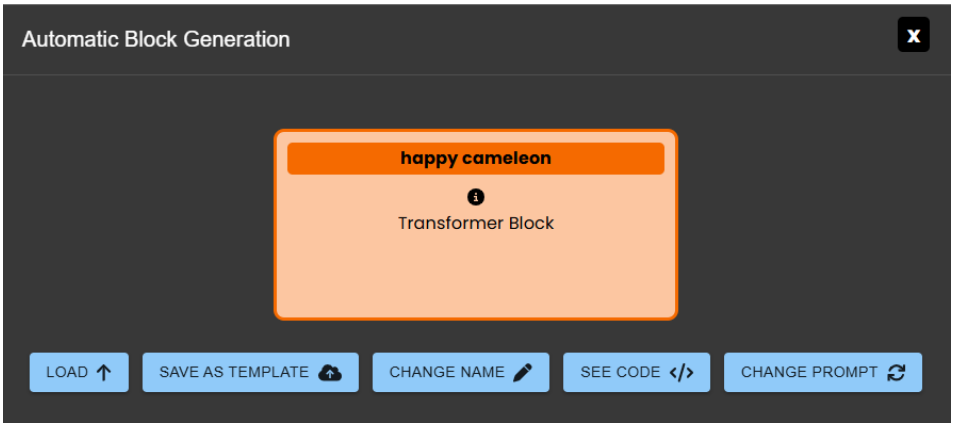

Figure 20 - Automatic Block Generation RAG interface (©SEDIMARK)

The assets of the offerings consumed from the SEDIMARK Marketplace application are available on the local running NGSI-LD broker and can be viewed in the Orchestrator by using the My Assets functionality within Operations. Each asset comprises valuable metadata describing the asset type and its properties.

Moreover, the Orchestrator offers the option of creating asset descriptions and publishing them to the NGSI-LD broker by accessing the Create Asset operation. This functionality is tied to the exploitation of workflow results, two illustrations of setting up the Asset Description creation are presented Figure 21 and Figure 22. Thus, end users are enabled to become publishers by leveraging their own resources, processing them using workflows, and creating subsequent asset descriptions based on the workflow results. Finally, the asset offering can be published on the SEDIMARK Marketplace based on the newly created asset description available on the NGSI-LD broker, which contains all the necessary information including the storage location of the asset.

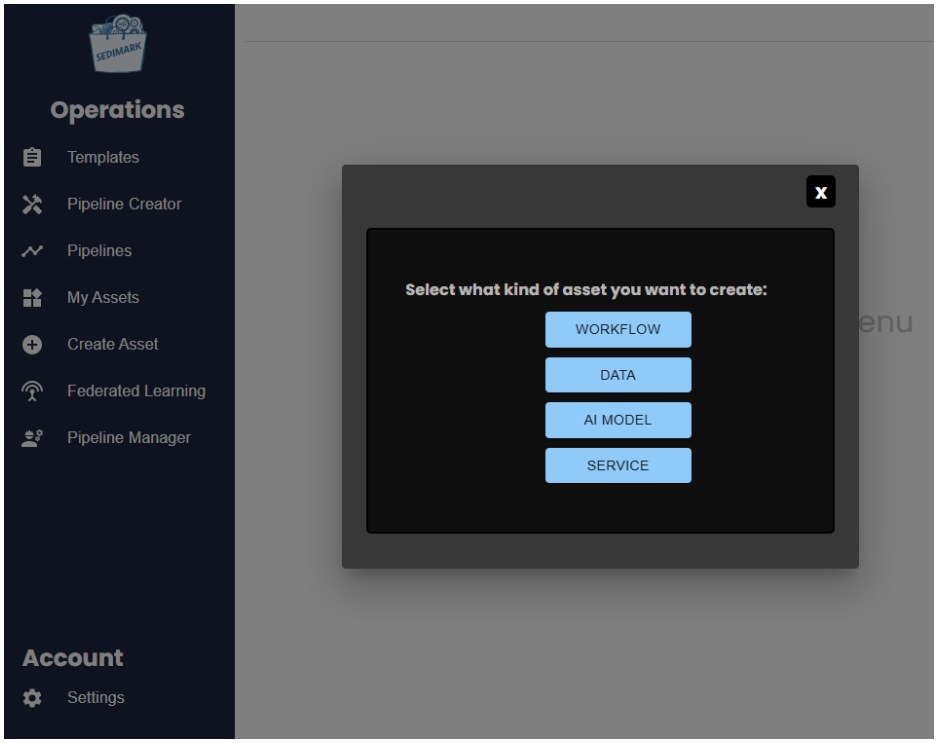

Figure 21 - Orchestrator UI Asset Description creation data types (©SEDIMARK)

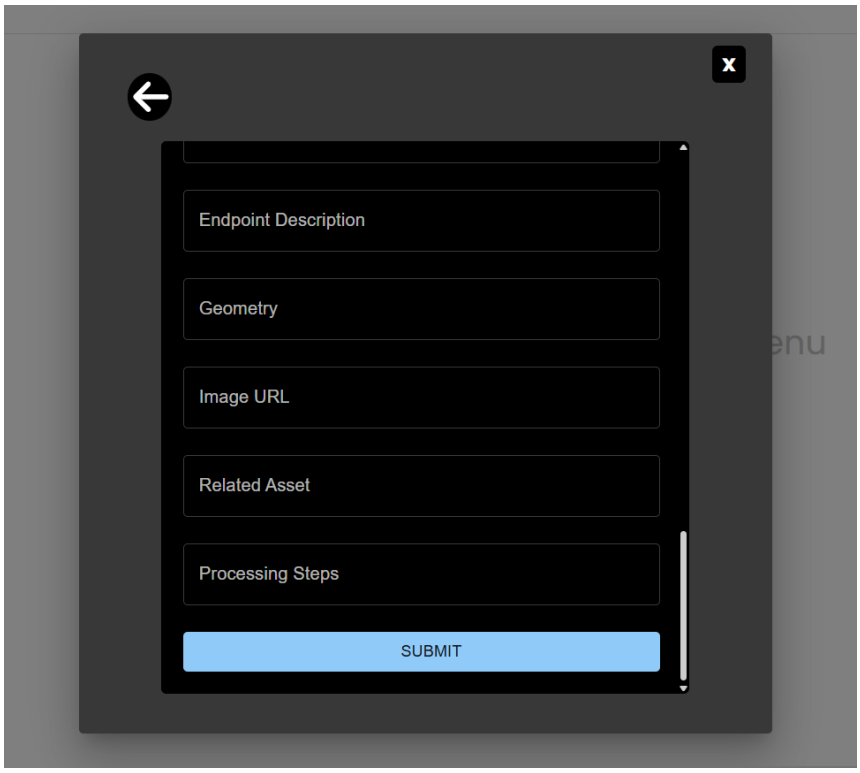

Figure 22 - Orchestrator UI Asset Description creation form (©SEDIMARK)

Another functionality of the Orchestrator is represented by the Federated Learning option which is a built-in solution that leverages the FL asset services which can be exchanged on the SEDIMARK Marketplace. This functionality enables users to configure and set-up workflows for FL either as clients or providers. As illustrated in Figure 23 the Orchestrator offers the option to set-up both the available Shamrock and Fleviden services, which can be configured by either setting up the values manually through a form or by uploading a YAML configuration file as depicted in Figure 24.

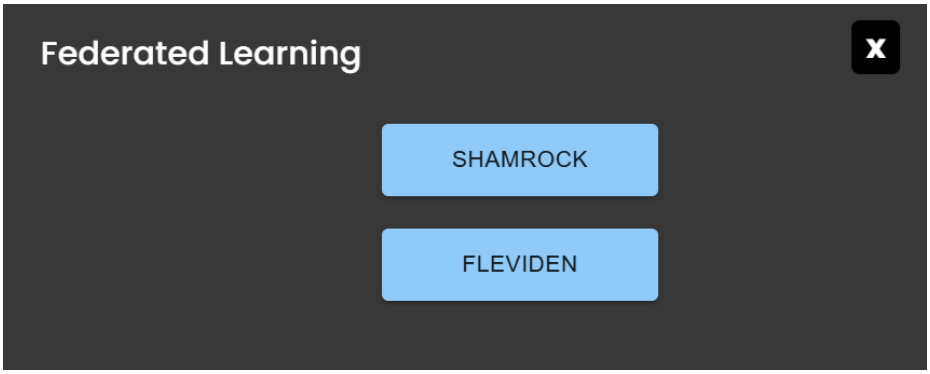

Figure 23 - Orchestrator UI FL service selection (©SEDIMARK)

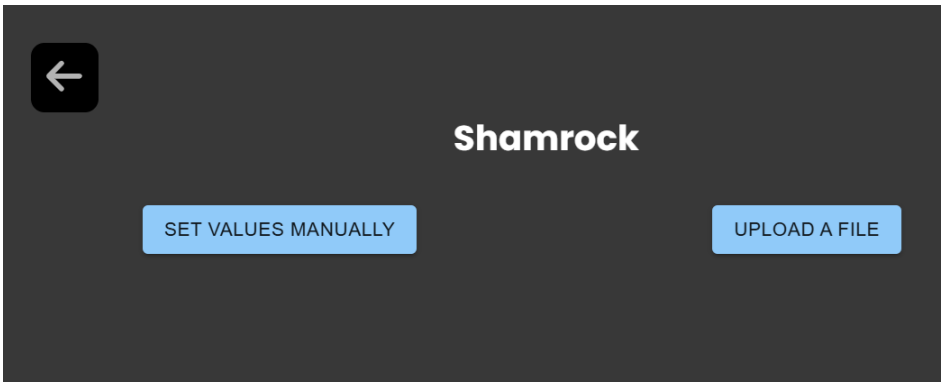

Figure 24 - Orchestrator UI FL service configuration option (©SEDIMARK)

The management of active workflows can be performed using the Pipeline Manager operation interface illustrated in Figure 25, which displays options for deleting, editing, CWL (Common Workflow Language) exporting and MageAI exporting workflows. The CWL option represents a standardized format for exchanging workflows between different types of applications, enabling workflows to be used outside of the SEDIMARK ToolBox ecosystem.

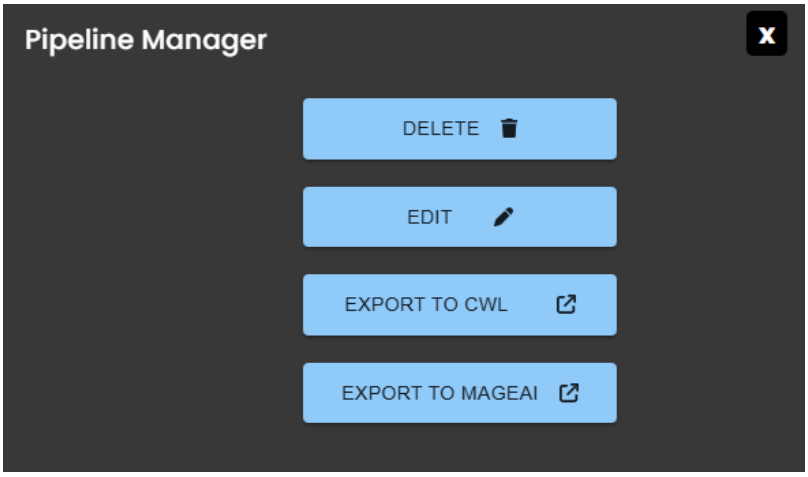

Figure 25 - Orchestrator UI workflows management (©SEDIMARK)

7 MageAI

This chapter introduces the MageAI solution, initially described in Chapter 7 'AI/ML Dashboard' of Deliverable D4.5 [1]. As part of the SEDIMARK ToolBox stack, MageAI serves as the main processing engine of the Data Processing Orchestration (DPO), managing ETL pipeline definition, execution, and triggering.

The SEDIMARK ecosystem requires powerful tools to effectively leverage its diverse asset offerings which include datasets, AI models, services, and workflows. MageAI was selected to meet this requirement due to its exceptional flexibility, performance, and ease of use. The platform provides comprehensive solutions spanning the entire data lifecycle, from initial data processing and manipulation through to advanced ML training and inference operations.

As previously established, MageAI represents the core engine of the DPO, while the Orchestrator UI, detailed in Chapter 6, provides a higher-level abstraction layer built upon MageAI's capabilities. This architectural relationship creates two distinct user experiences tailored to different technical requirements.

The general architecture and communication flow between MageAI and the Orchestrator UI is depicted in Figure 26. MageAI's comprehensive functionality suite delivers a rich user experience through essential quality-of-life features, including user authorization, resource management, an intuitive web UI interface, and real-time monitoring capabilities. Communication between MageAI and the Orchestrator UI is established through the MageAI API, a custom-developed wrapper that enables secure resource access and exposes MageAI instance functionalities to the higher-level interface.

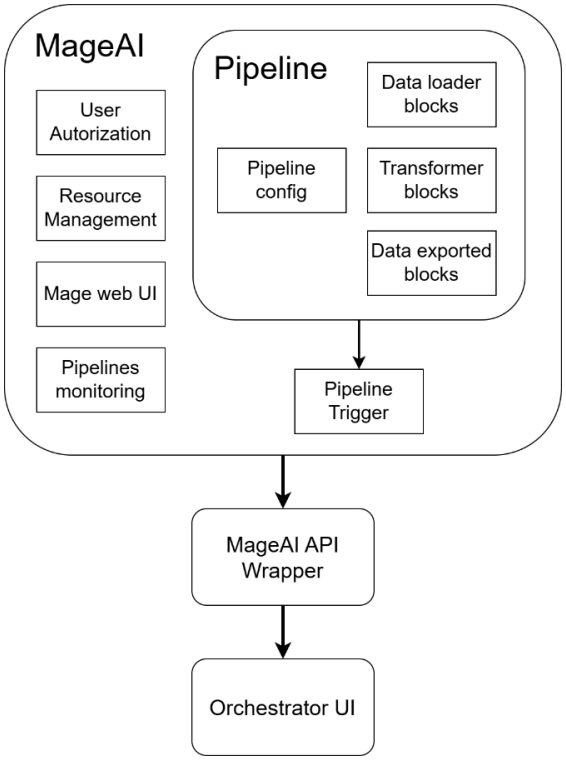

Figure 26 - Mage communication with Orchestrator (©SEDIMARK)

The Orchestrator UI simplifies pipeline creation by presenting them as user-friendly workflows that minimize required technical input. Users can leverage built-in generic workflows and configure variables at the block level, making the system accessible to both technical and non-

technical users. In contrast, MageAI offers a comprehensive development environment where experienced technical users can build pipelines and blocks from the ground up, implement custom logic, and configure sophisticated trigger mechanisms for complex data processing scenarios.

Another critical component of the DPO is the MLOps infrastructure, which streamlines and automates the complete machine learning lifecycle by providing comprehensive curation, versioning, provisioning, and storage capabilities for AI models. Within the DPO, MLOps practices are enabled through specialized tools including MLflow for model tracking and lifecycle management, and MinIO for scalable object storage. The MLOps communication flow within the DPO architecture is depicted in Figure 27.

The SEDIMARK Marketplace offerings include AI model assets, which are provisioned to the SEDIMARK ToolBox through MinIO object storage. These models can subsequently be loaded and utilized for fine-tuning or inference operations within MageAI pipelines. Furthermore, the process of publishing AI models as marketplace offerings involves a reverse operation: uploading the models to MinIO and linking the storage access information to the corresponding asset description.

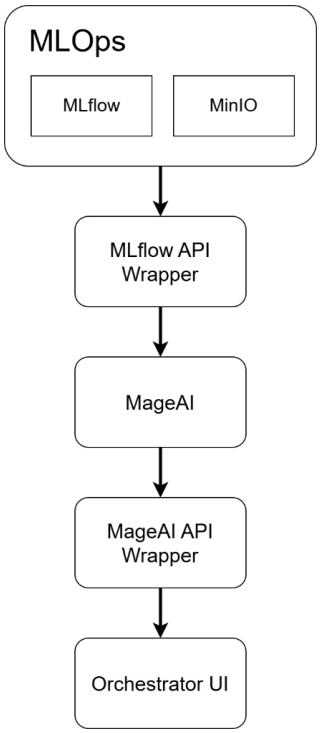

Figure 27 - MLOps communication flow (©SEDIMARK)

To ensure seamless compatibility between MageAI pipelines and Orchestrator workflows, MageAI employs a generalized pipeline architecture built on highly configurable variables that can be dynamically set within the Orchestrator interface. This architectural design, illustrated in the Figure 28, demonstrates how complex MageAI pipelines are abstracted into user-friendly Orchestrator workflows while maintaining full functionality.

The pipeline architecture centers on standardized data flow through pandas DataFrame objects, with specialized Data Interoperability blocks serving as crucial conversion layers between NGSI-LD format and DataFrame structures, enabling bidirectional data transformation. The system supports comprehensive data processing through two primary categories:

- **Data Preprocessing**, which includes data cleaning, transformation, anonymization, feature engineering, time series preprocessing, and data validation operations
- **Data Manipulation**, encompassing AI model training, inference, data aggregation and summarization, and KPI computation capabilities

External data source integration enables users to enrich their datasets and metadata information beyond marketplace offerings, while the integrated MLOps component provides essential model provisioning, storage, versioning, and lifecycle management throughout the pipeline execution. The output DataFrame maintains detailed variable information that serves as the foundation for generating comprehensive data asset metadata, facilitating the creation of new marketplace offerings based on processed results. This architecture ensures that regardless of pipeline complexity, users can leverage both the advanced capabilities of MageAI and the simplified interface of the Orchestrator UI according to their technical requirements.

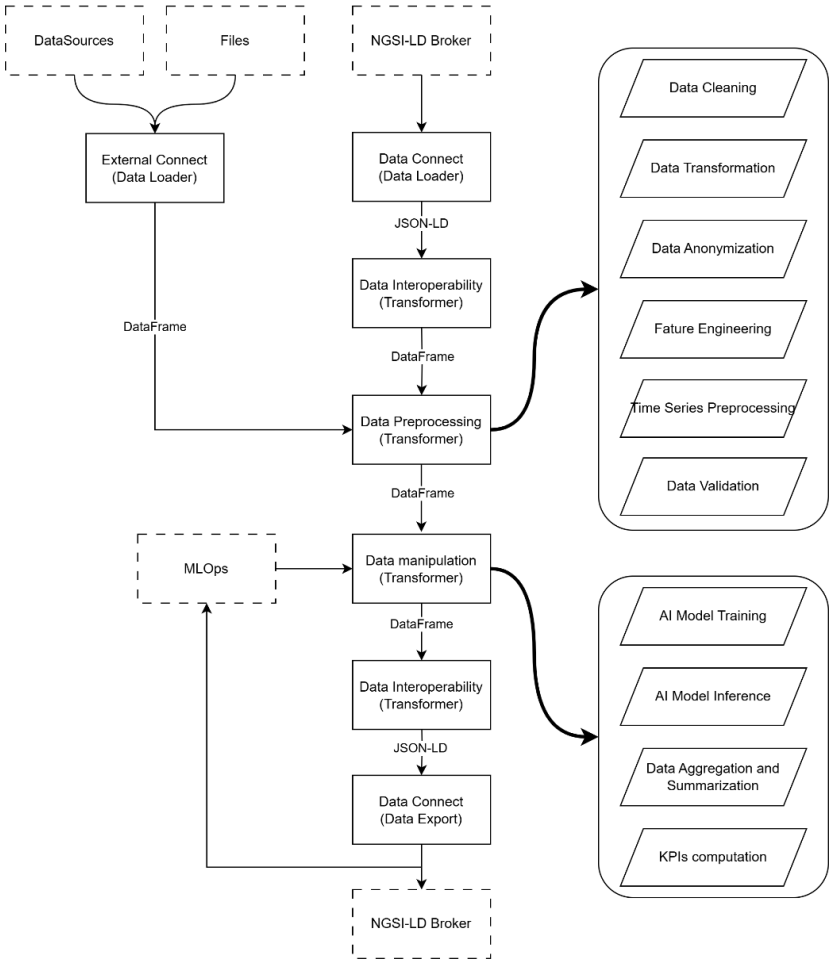

Figure 28 - MageAI Pipeline Architecture (©SEDIMARK)

New pipelines can be created from the ground up using the MageAI Web UI interface, with access to the Pipelines menu illustrated in Figure 29. The interface displays a comprehensive list of existing pipelines, including essential information such as status, name, description, type,

last updated date, creation date, associated tags, number of blocks, and number of defined triggers.

Pipeline creation is initiated by accessing the 'New' button and selecting the 'Standard (batch)' type, which opens a configuration dialog allowing users to specify the pipeline name, description, and relevant tags. This streamlined creation process provides immediate access to MageAI's full pipeline development capabilities while maintaining organized project management through the tagging and documentation system as shown in Figure 29.

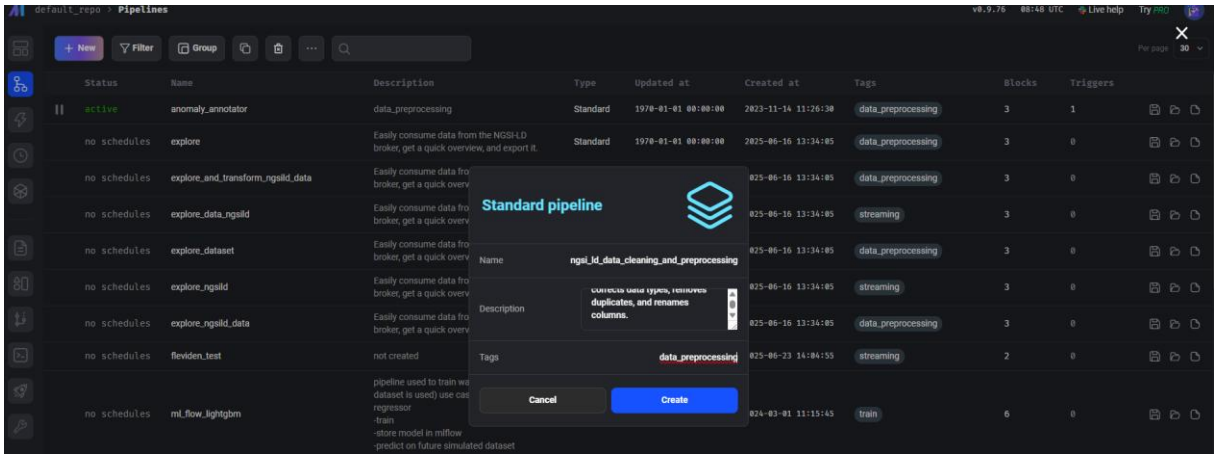

Figure 29 - Mage Web UI pipelines menu (©SEDIMARK)

MageAI pipeline creation offers flexible development approaches to accommodate diverse technical requirements and use cases. Users can leverage existing pre-built blocks from the default SEDIMARK workspace or develop custom blocks using Python, providing the versatility needed for specialized data processing scenarios. To ensure seamless integration with the generic MageAI Pipeline Architecture and enable on-demand use case adoption through the Orchestrator UI, developers must adhere to specific architectural requirements when creating pipelines.

While custom pipelines can be developed with complete flexibility, adhering to the standardized Data Preprocessing or Data Manipulation pipeline format enables seamless integration with the generic pipeline architecture. Generic pipelines serve as orchestrating frameworks that internally call and execute these specialized processing pipelines, creating a modular and reusable system architecture.

This architectural standardization allows complex Python-based MageAI pipelines to be abstracted into user-friendly Orchestrator workflows while preserving full functionality and customization capabilities. The result is a system that supports both advanced technical development and simplified workflow management, accommodating users across the technical spectrum within the SEDIMARK ecosystem.

An example of a Data Preprocessing pipeline visualized within the MageAI web UI is illustrated in Figure 30, demonstrating a comprehensive data preprocessing workflow, compatible within the generic pipeline architecture depicted in Figure 28, this pipeline comprises multiple configurable steps:

1. **NGSI-LD Data Ingestion and Conversion**: Initial data ingestion from the NGSI-LD broker with automatic conversion to DataFrame format for downstream processing

2. **Missing Values Handling**: Configurable strategies for managing missing data, including options to drop affected rows or columns, impute values using statistical measures (mean, median, mode), or fill gaps with user-defined constant values
3. **Data Type Correction**: Automated conversion of columns to specified data types with support for datetime parsing to custom string formats, ensuring data consistency throughout the pipeline
4. **Duplicate Removal**: Intelligent deduplication functionality that removes duplicate rows while providing options to retain the first occurrence, last occurrence, or remove all duplicates entirely based on user preferences
5. **Column Management**: Flexible column operations using mapping dictionaries to rename columns and selectively drop unnecessary columns from the dataset

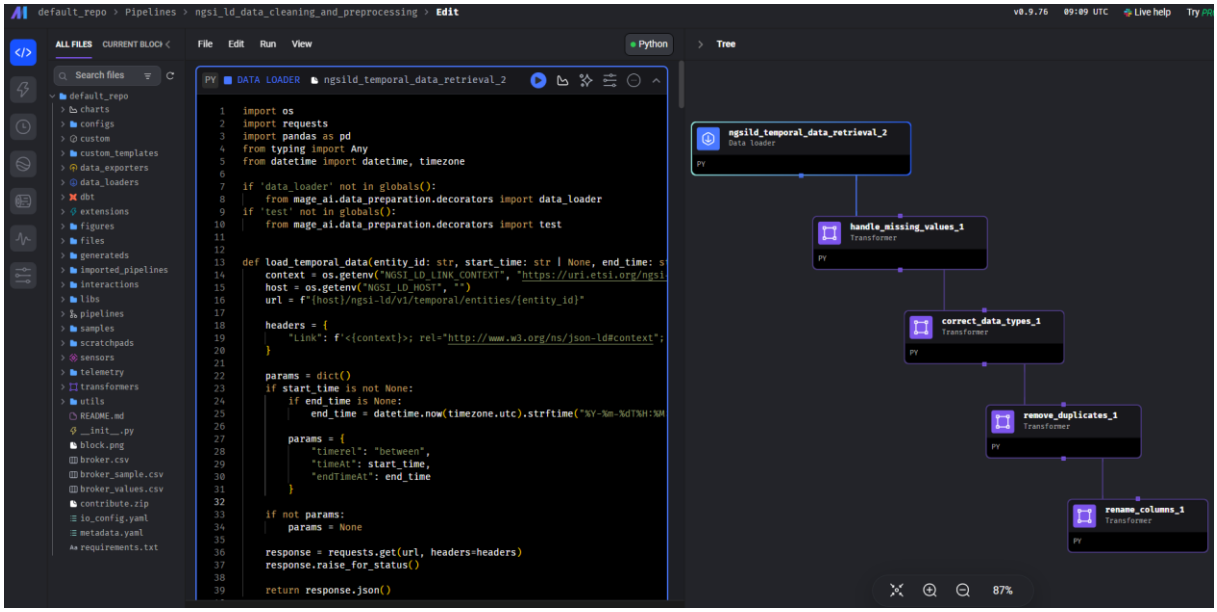

Figure 30 - MageAI data preprocessing pipeline example (©SEDIMARK)

Each preprocessing step is optional and fully configurable through the Orchestrator UI's block workflow variable configuration system, allowing end users to specify target columns and preferred techniques without requiring direct pipeline modification. The corresponding workflow imported into the Orchestrator UI provides a simplified interface for these complex operations, as visualized in Figure 31, demonstrating how technical MageAI pipelines are abstracted into user-friendly Orchestrator workflows.

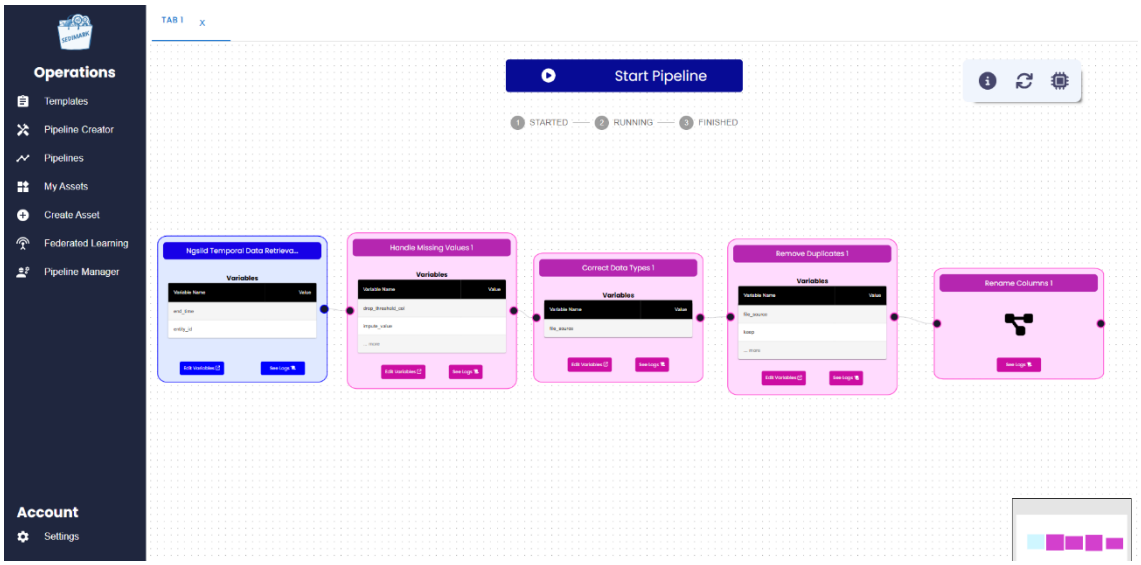

Figure 31 - Orchestrator UI workflow visualization over the MageAI pipeline (©SEDIMARK)

8 Recommender

8.1 Overview

SEDIMARK aims to provide personalisation services to users of the marketplace, so that they can easily and quickly find assets to purchase, regardless of the nature of these assets, namely, regardless of whether these are data, AI models, or services. Personalisation comes into place with providing results that are tailored to the user's needs and preferences, so that it is more likely that they will have a better experience and they will be more satisfied with their interactions with the system. Within SEDIMARK, the personalised services are mostly related to providing recommendations to participants (mostly to consumers, but without neglecting providers in some scenarios that will be described below) when they look for an Asset to purchase. In this chapter, we describe how we advanced the Recommender System to date. Briefly, in this chapter, we further refined the content-based recommender to provide more reliable results to users. In particular, we focused on two aspects of content-based recommender: (i) techniques to improve the dataset metadata; and (ii) techniques to improve the dataset retrieval based on queries.

8.2 Overview of Recommender Systems

Broadly speaking, the goal of a Recommender System (RS) is to suggest “relevant” or “good” items to the user. To design a recommender system, one needs to address the following three main points [11]:

- what do we define as “items” in the RS, for example, the RS design for Netflix users would define movies as items, Spotify’s items are songs or artists.
- how do we define “good” or “relevant” recommendation, and this is closely related to the next point.
- how do we evaluate the performance of the RS

To better understand all stakeholders involved in the RS, consider Figure 32 (adapted from [11]).

Figure 32 - Stakeholders in RS (adapted from [11] (©SEDIMARK))

In SEDIMARK, a provider offers a dataset, ML models or services or all of them, the system is the recommendation module as part of the SEDIMARK toolbox and the user is the consumer using the Offerings. Providers are responsible for adding descriptions (metadata) for their offerings, so that those offerings can be accurately retrieved when querying the marketplace. The metadata should also include appropriate keywords describing the suitable tasks for the offerings. To this end, in SEDIMARK, we provide several algorithms for metadata enrichment, in particular, filling in the missing keywords for assets being advertised on the marketplace.

Overall, RS can be divided into the following categories [12]:

1. **Collaborative filtering:** these methods are based on interactions between items and users as recorded by the system. For example, in Netflix, each user has a history of seen movies, therefore for these types of RS the entire history of viewed movies for each user is used as an input into the RS. The main goal of collaborative filtering is to find similar users and recommend items based on the items liked by similar users, such methods are referred to in the literature as neighbourhood-based methods. Often the user-item interactions are stored in the form of a sparse matrix. Methods that fall in this category include (i) latent space methods such as matrix factorisation techniques, and (ii) deep learning techniques such as multilayered perceptrons or convolutional neural networks.
2. **Content-based:** the goal of content-based methods is to find similar items to the items that the current user likes. In this case, each item is described with a set of attributes and the goal of the content-based algorithm is to group similar items based on their attributes. Using the Netflix example above, if the user likes comedy movies, the system will find other comedy movies and recommend those to the user.
3. **Community-based:** the recommendations generated by community-based methods rely on the preferences of the user's "friends". Apart from user-item interactions, these methods also need social networks to describe the friendships between users.
4. **Demographic:** the assumption of the methods falling into this category is that people from different demographic areas should have different recommendations. An example could be recommending articles based on the language of the user's country.
5. **Knowledge-based:** these methods rely on domain knowledge. One example of such a method is the case-based approach [13], where the problem description defines the user's needs and the solutions are the recommendations.
6. **Hybrid:** these methods rely on a combination of two or more techniques described above. The goal is to use multiple methods to overcome problems faced by each particular method. For example, a hybrid method combines collaborative filtering and a community-based method and uses the community-based approach to overcome the "cold start" problem, which arises when a new user joins the system and has not consumed any particular item yet.

In the current version of SEDIMARK, the goal of the RS is three-fold: (i) to recommend missing keywords; (ii) retrieve top-N assets that best match to the query submitted by the marketplace user; and (iii) find top-N most similar assets to the given asset. The current version of the SEDIMARK recommendation module includes several methods for keyword recommendation as well as several methods for query-based asset retrieval. These are described in more detail next. Note that the current version of the implementation does not include any personalisation, as there is no data within SEDIMARK as of yet to deploy personalised recommendations.

8.3 Design of the Recommender module

8.3.1 Internal Structure of the Recommender module

The Recommender module is a functional component that is part of the "SEDIMARK specific components" of the Marketplace Enabler as discussed in Deliverable SEDIMARK_D2.2 [2]. The goal is to provide a complementary service to the users of the SEDIMARK platform so that they can receive personalised results when performing queries for discovering Offerings

and Assets through the Marketplace. The internal structure of the Recommender module and the interfaces and interactions with external components are depicted in the figure below:

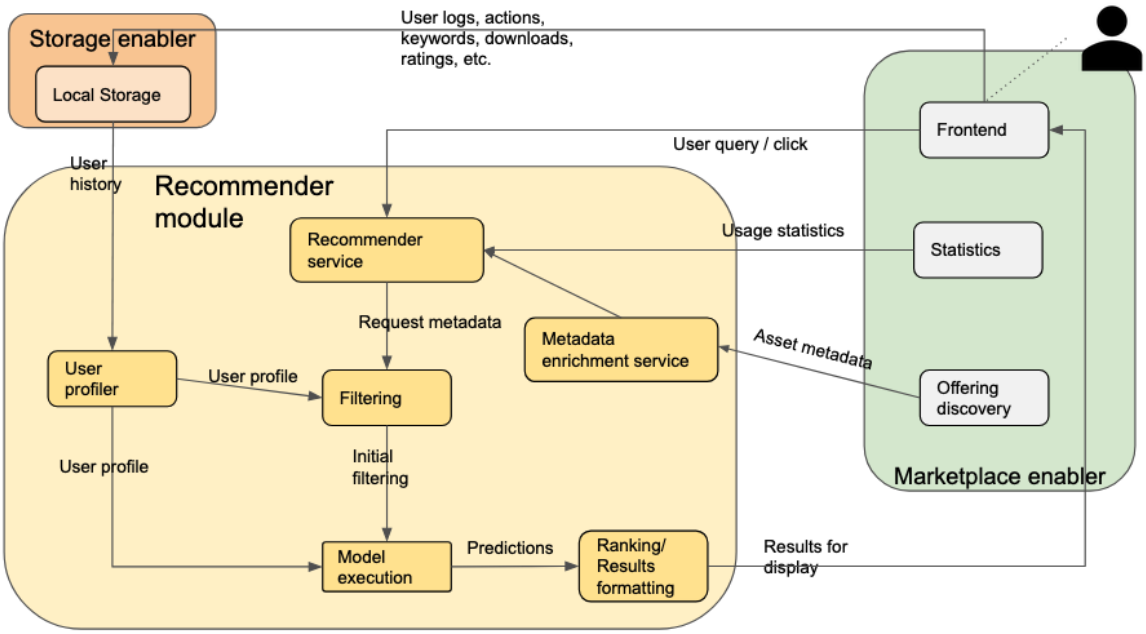

Figure 33 - Internal structure of the recommender module and its interactions with external modules (©SEDIMARK).

The current version of Recommender module comprises six sub-modules, each of which is responsible for a separate task of the recommendation service. More detailed information is given below:

- **Metadata Enrichment service:** this component is crucial in filling in the missing keywords so that the recommender service can retrieve the most relevant asset based on the given query.
- **Recommender service**: this component plays the role of the wrapper service that manages the recommendation functionalities. It is responsible for interacting with the external components and especially the Frontend, in order to (i) receive the recommendation query, (ii) analyse it, (iii) extract the required information, (iv) convert the query into a specific format to be used internally in the service and then (v) start the process for executing the recommendation pipeline to get the final results and return them to the Frontend to be displayed to the end user.
- **User profile**: this component is the main component enabling the personalisation of the recommendations, computing the user profiles based on user's past interactions with the Marketplace and user demographics.
- **Filtering**: this is the component that does an initial filtering of the items/assets to be recommended to the user, so that a reduced list of candidate items will be the input to the Recommender model, aiming to reduce the complexity of the computations and speed up the model execution.
- **Model execution:** this is the actual component that uses input about the user profile and the candidate items, as well as external information (statistics) and executes the trained

recommendation model to make predictions about how well the candidate items match the user profile.

- **Ranking/Results formatting**: this component handles the final step of the recommendation module that takes the results of the recommender model and ranks the candidate items (based on several criteria), creating the final ordered list of items to be recommended to the user in the correct format to be used by the Frontend to display the list to the user.

As can be seen in Figure 33, the Recommender module interacts mainly with components of the Marketplace enabler and of the Storage enabler. The detailed description of those components was given in SEDIMARK_D4.5 [1]. Below, we describe the interactions and the information that is exchanged between the respective components:

- **Recommender Service ← Frontend**: SEDIMARK assumes that the user of the platform (either a provider or a consumer) will navigate the Frontend, discovering new offerings and assets for purchase. Any **queries** that the user makes to the system for asset/offering discovery will be forwarded to the Recommender Service, so that it uses the context of the query in order to provide personalised recommendations to the user. Additionally, considering that recommendations are also provided when users click on interesting items on the Frontend or when they purchase items, any user "**clicks**" on the Frontend are also being forwarded to the Recommendation Service for launching a new recommendation process. Note, that since in the current version of SEDIMARK we still do not have access to personalised user queries, to test our modules we use publicly available datasets of datasets containing user queries. These queries are currently divided into train/set queries, so that they can be used to train and test the metadata enrichment and recommendation services.
- **Recommender Service ← Statistics**: The Recommender systems normally use extra contextual information about recent trends and popularity of items for ranking of items in order to help users explore more the available lists of items. To do so, the Recommender Service needs to periodically get information from the Statistics module regarding the usage of the items (i.e. how many likes/dislikes, how many times downloaded or clicked) and the recency of the actions on the items (i.e. how many clicks/downloads the last day, week, month).
- **User Profiler ← Local storage**: One important task of the Recommender Service is to be able to identify the user preferences over the items, so that it can recommend items that have higher chance of getting a positive reaction by the user. In order to do so, the User Profiler module needs to get demographic information about the user (i.e. gender, age, occupation, location) and the history of the user interactions with the Marketplace, i.e. past clicks, purchases, likes, etc. The Local Storage is assumed to keep the logs of these interactions of the user with the Frontend in a way that the User Profiler can easily look for and get access to.
- **Metadata Enrichment Service ← Offering Discovery:** The Recommender Service needs to know the full catalogue of available items and contextual information about the items so that it can identify the most preferable for the user. However, as mentioned in the earlier sections, sometimes such information can be incomplete. Therefore, the Metadata Enrichment Service queries the Offering Discovery module and receives information about the Assets that are available from the providers, as well as any available information that is extracted from the Offering descriptions for these Assets. The Metadata

Enrichment Service then proceeds and fills in the missing metadata for the assets on offer. This ensures that all missing metadata can be fixed and improves the accuracy of the Recommender Service.

- **Recommender Service ← Metadata Enrichment Service**: The Recommender Service receives all up-to-date information about the available assets from the Metadata Enrichment Service so that it can produce accurate recommendations.
- **Recommender Service → Filtering**: the Filtering module gets the query information and the Asset list from the Recommender Service in order to perform a first filtering of the items to reduce the candidate list.
- **Filtering ← User Profile**: the Filtering module also receives the user profile information to help take the filtering decisions, to avoid filter out preferable items or to filter out items that the user has disliked or purchased in the past.
- **Model Execution ← User Profile**: the Model Execution module needs to receive all the information about the user profile to use it as input for the model to be executed.
- **Model Execution ← Filtering**: the Model Execution module needs to get from the Filtering module the (filtered) candidate list of items to be recommended, along with their contextual information.
- **Ranking/Results Formatting ← Model Execution**: after the Model Execution module runs the inference of the model it outputs predictions about candidate items which are forwarded to the Ranking module in order to produce the final ranked list of items in the correct format to be sent to the Frontend. The Ranking module can also include methods for adding diversity or novelty to the ranked list, to improve the quality of recommendations.
- **Ranking/Results Formatting → Frontend**: after the final ranked list of recommendations is computed it is then converted to the proper format and sent to the Frontend to be displayed to the user.

8.3.2 Recommendation data flow

The data flow for the recommendation service was presented in D2.2 in Section 7.8 [7]. Here in Figure 34, we present an improved version of the data flow for the sake of completeness. When a user interacts with the Frontend of the marketplace, the Recommendation module receives a recommendation request, which includes the metadata of the request (i.e. the search query, the item clicked, etc.). Then, the Recommendation module sends the information to the User Profiling to get the profile and preferences of the user. The User Profile computes this information periodically, getting the historical data and the logs from the Logging module. The Recommendation module also interacts with the Offering Discovery and the Statistics module to get the candidate list of items to be recommended, the metadata of the items and the usage statistics. Then, having all the required information, the Recommendation module filters out the unneeded candidate items, runs the Recommender model (using the Model Inference module), ranks the predictions and formats the results in the proper way to be sent to the Frontend for displaying them to the user.

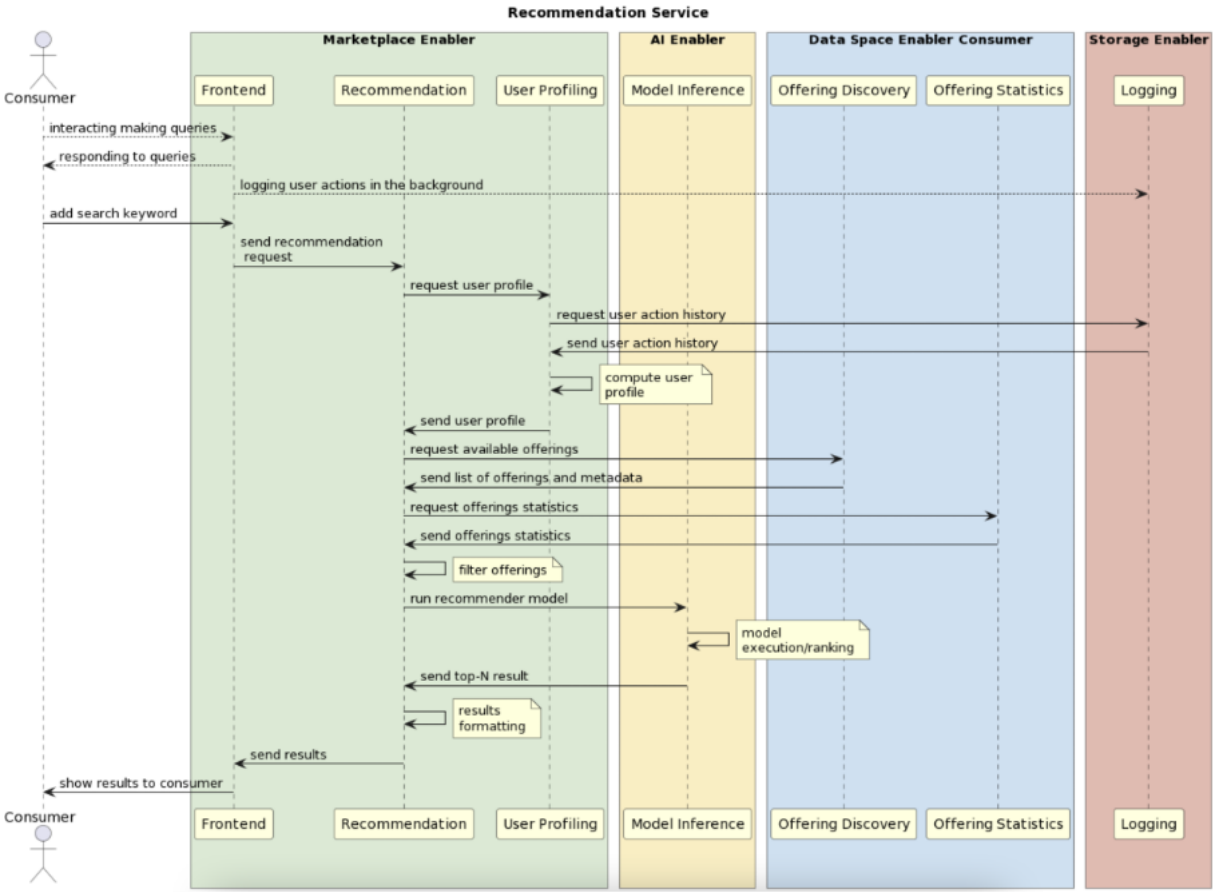

Figure 34 - Recommendation service data flow example (©SEDIMARK)

8.3.3 User and item profiling

One key requirement for building recommender systems is to have representative features to describe the users and the items that are going as input to the Recommender model. Significant research has been devoted into extracted features from datasets. Within SEDIMARK, the decentralised nature of the Recommender System assumes that all the user information will be stored locally on the user's Local Storage, and this will be used in order to extract demographic information about the user, as well as historical data with regard to the user interaction with the Marketplace.

The user profile information is split into two parts:

- **Demographic information**: This can be either added directly by the user to the system or can be gathered by the system via a questionnaire when the user first logs into the system:
  - **Age/gender** is normally used as part of the user profile in Recommender systems, however this information seems mostly irrelevant for recommending datasets and models, but might be useful when recommending services.
  - **Occupation** is considered an important factor, since depending on the domain of occupation related assets can be recommended.
  - **Domains of interest** is also an interesting feature for this type of recommendations. For example, researchers can select communications or biology as domains of interest to get more recommendations for Assets from these domains.

  - User **location** can also be used to recommend services and datasets that are in areas close to the user. For example, in case of weather monitoring, a user might be more interested into getting a weather dataset from their city or country compared to a city on the other side of the world.

- **Historical data**: this is captured by the Frontend and stored on the Local storage, from where the Recommender module gets the information. Within SEDIMARK, historical data for the user can be the following:
  - User **activity** on the Frontend, i.e. clicks, purchases, etc.
  - Previous user **search queries** to identify trends in what the user is looking for and compute their preferences through that information.
  - User **likes/dislikes** on specific Assets of the Marketplace, which will help the Recommender system have direct information about the interests and the preferences of the user.
  - User **reaction** to recommendations, by clicking on a recommended item or disliking the recommendation. This will help the recommender system improve its results.

The item profile information is used to characterise the assets that are candidates for recommendation by the recommender system. This information will be extracted from the Offering descriptions that the providers will define and will be received by the Recommender system as part of the interaction with the Offering Discovery module. The example description of the Assets is inspired by the DescribeML language, which is a tool used to describe ML datasets [5]. Considering that in SEDIMARK there are three main types of assets, different item profile information is aimed to be used by the system:

- **Dataset profile information**: the information regarding datasets that might be used as item features by the Recommender system is the following:
  - **Domain of interest**, i.e. water, energy, health, transport, environment, etc.
  - **Type** of dataset, i.e. streaming or fixed dataset
  - Dataset **category**, i.e. measurements, tabular, user data, etc.
  - Dataset **features**, i.e. what are the column names in the dataset table.
  - **Location**, in a predefined format
  - **Purpose**, defining what was the purpose for gathering and sharing the dataset.
  - **Target** usage, i.e. classification, regression, recommendations, etc,
  - **Statistics** regarding the usage of the dataset
  - **Size**, in a predefined format,
  - **Flags**, i.e. if the dataset is being available as part of distributed model training within SEDIMARK.
  - Extra **keywords** that can be processed and converted to features.
- **AI models profile information**: for the profile of AI models in addition to most of the above-mentioned dataset profile information we assume that the following information will be useful for the Recommender:
  - **Dataset** used for training the model.
  - **Domain of interest**
  - **Target activity**, i.e. classification, regression, rating prediction, etc.

- **Type** of model, i.e. simple, DNN, LSTM, etc.
- Model **description**, in the predefined format of SEDIMARK
- **Training framework** used, i.e. Tensorflow, Keras, PyTorch

- **Service profile information**: it is assumed that the service profile information required will be similar to the dataset information.

8.4 Implementation

8.4.1 Overview

The current version of SEDIMARK specifies three versions of recommendations as follows:

- **asset keyword recommendation**: to fill in missing keywords within the asset catalogue
- **query-based recommendations**: this takes place when the user is performing a query aiming to discover some offering or asset.
- **item-based recommendations**: this takes place when a user interacts with an item (click, purchase, rate, etc.)

More details on the models that are implemented for each of the scenarios are given below.

8.4.2 Asset keyword recommendation

High-quality asset metadata can contribute significantly to the accuracy of asset recommendations in the SEDIMARK platform. However, upon examining dataset metadata from several other dataset repositories, it is clear that keyword information is often missing or incomplete. To illustrate this, we present in Table 1 metadata of datasets collected from two different platforms (COVID data extracted from Zenodo repository [14] and Research Data Australia - ARDC- repository [15]). As can be observed from the table, as much as 40% of datasets lack a set of keywords.

Data repository	Zenodo-COVID	ARDC
# Records	2558	127313
# Records with Description	2531	125780
# Records with Keywords	1711	70763
# Records with Description and no Keywords	1709	70495
Fraction of missing Keywords Overall	0.33	0.44

Table 2 - Overview of missing information in public dataset repositories.

As specified in SEDIMARK_D4.5 [1], one way to fill in the missing keywords is to use KeyBERT [16] language model that will extract keywords based on the description of the asset. However, as described above, the keywords in this case often describe the suitable tasks for the given asset, while the asset description is a general description of the asset. Moreover, KeyBERT works well on longer text documents, but the asset description is generally fairly short. As such, it may not be possible to extract the task from the asset description alone. Therefore, in our implementation, we decouple KeyBERT and use it alongside the training queries to find the most suitable keywords. We call our implementation **KeyRec**. In high-level, the implementation works as follows:

- we first combine all train queries into one large text document
- next, similarly as in the original KeyBERT we extract the candidate keywords using ScikitLearn [17], however differently from KeyBERT, which uses the associated text, we use all combined train queries in an attempt to find suitable tasks for the given asset
- finally, we use KeyBERT to select the most suitable keywords. For this step, KeyBERT will use the list of candidate keywords as specified in the previous step and select the most suitable ones based on the asset description.

8.4.3 Query-based recommendation

In the query-based recommendation scenario the goal is to provide to users recommendations that are related to the discovery query that the user makes for finding interesting Assets and Offerings. To do so, the process is split in two parts:

- identifying items that are related with the user query.
- personalise the list of candidate items as defined by the previous step.

For the first step, the main idea is to identify the related *features* in the dataset of Assets and then process these features in the appropriate way to use it in the Recommendation model. Considering the item features described above, there can be two three main types of features:

- numerical, which are inputted as they are in the model.
- categorical, which are usually one hot encoded before being used by the model.
- plain text, which are usually processed using mechanisms from natural language processing (NLP).

In the current implementation, the main goal is to identify *"keyword*" features that can be extracted both by the metadata of the assets and the user query, and use a model to "match" the keyword features, in order to find the assets that match best to the specific query of the user. Then, the top-N assets that have a higher similarity score with the user query are the ones that are presented to the user as recommendations (or are the candidate ones for the "personalisation" step). In this first implementation, there is no "personalisation" step, since there are yet no user data available.

BERT-based implementation

The first step of the process is to use the Assets' Self-Descriptions in order to extract keywords using a language model. The options currently are to use:

- **"KeyBERT" [16]**, which is a language model specifically made for keyword extraction, based on the Bidirectional Encoder Representations from Transformers (BERT) family of language models developed by Google [18]. The benefit of using KeyBERT is that it is an easy to use and minimal keyword extraction technique to extract keyphrases.
- **sciBERT [19]**, which is a BERT model trained on scientific text and might be more useful in terms of extracting keywords from datasets and models in the SEDIMARK scenario.

In our example dataset, the metadata that are used as input to the keyword extractor models are the dataset description, additional relevant information added by the dataset owner and information about the fields of the datasets. This process will create a list of keywords for each one of the assets.

The next step of the process is to extract the keywords from the user query using the same exact BERT model as for the asset keywords.

Then, what follows is to create a similarity matching between the keywords of the Assets and the keywords of the query. This is currently done using "*sentence transformers*" [20], either directly or through the "*sentence similarity*" [21] Python library. The goal of using the sentence transformers is to have a more meaningful way to compute the similarity of the keywords compared to standard methods that compare strings like Hammington distance [22] or Levenshtein distance [23], which do not take into account the semantic similarity of the strings. Sentence transformers are based on BERT and are used to encode the keywords into word embeddings so that they can be easily compared.

For comparison of the keyword embeddings, we use the cosine similarity as the metric and use the scores in order to rank the assets and produce the top-N ranked list to display to the user.

Indexing-based implementation

The next implementation is based on the term-based information retrieval techniques. As a first step, all assets are indexed based on the asset title, asset description and asset keywords. We use the BM25 method [24], which is based on the "bag of words" approach. During the inference, assets are ranked based on the best match to the given query. Then, as the next step, to improve the asset recommendation, we use the RM3 re-ranking method [25]. This method expands the original query based on the assets retrieved by the first pass of the BM25 model. Based on this expansion, the assets are re-ranked. As we will show, this re-ranking method offers a significant boost in the accuracy of the query matching process.

LSI-based implementation

The last implementation of the process uses Latent Semantic Indexing (LSI) [26] as implemented by the gensim Python library [27]. LSI uses singular value decomposition (SVD) to identify patterns in relationships between terms and concepts in texts. In our implementation, we use LSI to create a model of the abstracts of the assets, to use it to extract text embeddings. After building that model, we extract the embeddings for the user query, using the LSI model we built before. Then, we compute the similarity between the two sets of embeddings to create the top-N rank list. This method using LSI does not require or using a language model and can be much faster and more lightweight, without needing to download a BERT model compared to the previous described method.

8.4.4 Item-based recommendation

In this scenario, the goal is to exploit the information about user actions on the Frontend i.e. to provide similar items to the ones that the user either purchased or clicked. This functionality resembles the "items similar to your purchase" recommendations of online websites.

In our current implementation, the first step is similar to the query-based recommendation, to extract the keywords for the items (assets) to be recommended. This can be done in any of the ways described above, i.e. using KeyBERT. The next step is to create term frequency - inverse document frequency (tf-idf) embeddings of the keywords [23], so that they are easy to compare. Tf-idf is widely used in information retrieval to measure the importance of a word in a document. After that, the cosine similarity matrix of the items of the dataset is computed, by calculating the cosine similarity between the embeddings of pairs of items.

Then, the recommendation list for similar items to a target item is computed by sorting the column of the similarity matrix that corresponds to the index of the target item.

8.5 Results

To test the newly added KeyRec and the query-based asset recommendation, we used the Datafinder dataset [28]. This dataset contains dataset descriptions and keywords based on the scientific publications using the given datasets. The dataset also contains train queries generated using the Galactica LLM, incorporating the information from abstracts of the relevant scientific publications. The test queries were generated using human annotators. Note that this dataset has high-quality metadata, and as such, to test the KeyRec, we randomly drop keywords from 60% ofthe datasets. We show the results in Table 3. For comparison, we also include the results from the sentence transformers models described in SEDIMARK_D4.5.

First of all, we can note from the results that adding the RM3 re-ranker on top of the BM25 indexing adds a significant boost to the accuracy of the model and this is regardless of whether KeyRec is used or not. Therefore, in SEDIMARK, we add RM3 by default on top of the BM25 indexing model. Secondly, as discussed in Section 8.4.3, we can observe that the indexing model indeed outperforms the embedding model (based on the sentence transformers). This is likely because the dataset descriptions and keywords contain short texts, rather than lengthy text documents, and therefore there is no need to use complicated embedding models in this case. Lastly, we note that adding KeyRec can significantly boost the overall performance of the query-based recommender model, even outperforming KeyBERT.

Method	P@5	P@5 p-value	R@5	NDCG@5
BM25 description	0.1312	—--	0.2785	0.2451
BM25 description+KeyRec	0.1347	0.0605	0.2772	0.2395
BM25 description + KeyBERT	0.1312	0.1	0.2678	0.2344
BM25+RM3 description	0.1432	0.0015	0.3003	0.2586
BM25+RM3 description+KeyRec	**0.1558**	**0.0001**	**0.3109**	**0.2635**
BM25+RM3 description + KeyBERT	0.1474	0.0043	0.2986	0.2544
Embeddings description	0.1333	0.0846	0.2635	0.236
Embeddings description+KeyRec	0.1411	0.0377	0.2806	0.2516
Embeddings description + KeyBERT	0.1347	0.0751	0.266	0.2408

Table 3 - Comparison of different methods for the query-based recommender

The next set of experiments demonstrate the likely output of the SEDIMARK Recommender module. Note that since currently there is no related data within the project, we used the datasets from [16]. This includes a list of research datasets with a number of accompanying features, i.e. description, related papers, modalities, tasks, number of papers using it, keywords, etc. For showcasing the recommendations, an initial simple user interface on Jupyter Notebook was developed, with a simple input box, where the user can input their query for an Asset. There is no limit on the length of the input or how the input should be formatted. The processing part of the recommendation system will convert the query to embeddings with one of the methods discussed in the previous sections.

Figure 35 - Example search bar of recommender in Jupyter notebook

As also discussed above, two types of recommendations are provided as output of the query. First, there is a list of recommendations regarding datasets that best fit the query of the user. Also, for each recommended dataset, a list of "related datasets" is also presented. This was developed to emulate the action when a user "clicks" or "purchases" a dataset.

8.5.1 Query based recommendation examples

In Figure 36 below, we provide some results for example queries:

Query: "reviews" → looking for dataset that include user reviews

Acronym	Keywords
Coffereview Dataset	coffee reviews review bean grading
Yelp Review Polarity	polarity yelp positive reviews review
Cookie	amazon conversational dataset recommendation agent
Amazon Product Data	reviews amazon dataset ratings metadata
BeerAdvocate	beeradvocate beer ratings reviews review
Amazon Fine Foods	finefoods reviews foods amazon ratings
Amazon Review	amazon reviews review sentiment dvds
Casino Reviews	casino dataset reviews google sentiment
Commonsense-Dialogues	commonsense dialogues crowdsourced social contexts
Multi-Domain Sentiment Dataset v2.0	ratings sentiment reviews amazon domains

Figure 36 - Example recommendation results with query "reviews" (©SEDIMARK)

Query: "speech recognition"

Recommendations:

Acronym	Keywords
VoxForge	voxforge transcribed speech dataset recognition
MAVS	audio smartphone smartphones speaker recognition
CI-AVSR	cantonese audio avsr dataset recognition
CN-Celeb-AV	celeb dataset audio cn recognition
Fontenay Dataset	transcribed handwritten texts fontenay recognition
MIntRec	intent multimodal mintrec recognition audio
ARVSU	utterances addressee arvsu recognition utterance
ADVANCE	multimodal aerial recognition dataset audio
KOHTD	handwritten handwriting recognition papers dataset
MSI	eeg emotion hierarchical recognition induction

Figure 37 - Example recommendation results with query "speech recognition" (©SEDIMARK)

Query: "audio processing"

Recommendations:

Acronym	Keywords
VoxForge	voxforge transcribed speech dataset recognition
MAVS	audio smartphone smartphones speaker recognition
CI-AVSR	cantonese audio avsr dataset recognition
CN-Celeb-AV	celeb dataset audio cn recognition
Fontenay Dataset	transcribed handwritten texts fontenay recognition
MIntRec	intent multimodal mintrec recognition audio
ARVSU	utterances addressee arvsu recognition utterance
ADVANCE	multimodal aerial recognition dataset audio
KOHTD	handwritten handwriting recognition papers dataset
MSI	eeg emotion hierarchical recognition induction

Figure 38 - Example recommendation results with query "audio processing" (©SEDIMARK)

8.5.2 Item based recommendation examples

Regarding item-based recommendations, we recommend 5 similar datasets to the one the user has selected. Example results are shown below:

Dataset: "Coffereview Dataset"

Recommendations:

Query for asset: Coffeereview Dataset

Acronym	Keywords
DBRD	dbrd sentiment reviews dataset review
BeerAdvocate	beeradvocate beer ratings reviews review
NSMC	korean reviews nsmc review naver
AmazonQA	amazonqa amazon reviews questions review
Amazon Review	amazon reviews review sentiment dvds

Figure 39 - Example recommendation results with query "Coffeereview Dataset" (©SEDIMARK)

Dataset: "CIFAR-10"

Recommendations:

Query for asset: CIFAR-10

Acronym	Keywords
20Newsgroup (10 tasks)	dataset classes pycontinual classification class
VOC-MLT	voc voc2007 classes tailed class
CIFAR-100	classes images class label labelers
Tobacco-3482	dataset images tobacco classes 3482
Stickers	stickers sticker images image alpha

Figure 40 - Example recommendation results with query "CIFAR-10" (©SEDIMARK)

8.6 Future work

As specified in SEDIMARK_D4.5 [1], the future work for the asset recommender will focus more on the aspect of personalisation. However, since SEDIMARK does not have user-defined historical data as it has not been deployed yet, and it is hard to acquire such data from third parties for testing purposes, we focused more on improving the query based recommender in this deliverable.

Overtime, after SEDIMARK is deployed, we will be able to collect historical data for each user such as for example the user's past purchases, preferences for certain items or services, and so on. As the system builds enough data, we aim to include more sophisticated RS models based on collaborative filtering as specified in Section 8.2. As can be seen in the previous section, current results are encouraging, but there is still quite some room until they are "production-ready". To improve the quality of our recommendations even further we aim to use hybrid techniques, specifically in cases of new users joining the system and to avoid the cold start problem.

Collaborative filtering techniques rely on user-item interactions and one major concern with such data is user privacy. To this end, we aim to develop decentralised approaches, where users do not need to share their raw private data with other users or global servers. But instead, each user builds the model privately only sharing the model updates. Although research has shown that such models could be reverse-engineered or are susceptible to a wide range of attacks [29], [30]. We aim to address such issues as follows:

- **User privacy**: collaborative filtering models generally consist of two types of parameters, one related to users and the other related to items. We aim to keep the user's model parameters private and share the parameters of the items during the model training process [19]. To improve the user privacy further we also aim to add differential privacy to the shared parameters [31].
- **Back door poisoning attacks**: FL systems are susceptible to back door attacks [29], [30], where a malicious user can alternate the final model in order to get some gain. For example, in the case of a recommender system the aim of a malicious user is to promote certain item to increase their revenue. We aim to employ techniques in order to prevent such attacks such as for example [32], where the authors propose a detection method consisting of four main parts: (i) reverse engineering, (ii) global reverse trigger generation, (iii) outlier detection and (iv) model repair.

9 Open Data enabler

The Open Data enabler is a component of the SEDIMARK ecosystem whose goal is to promote data sharing in the Marketplace by providing free and open datasets to all participants through the SEDIMARK Catalogue. A high level description can be found in section 6.6 of Deliverable SEDIMARK_D2.3 [4]. This chapter focus on the architecture of this component, its implementation and how to operate it.

9.1 Architecture

Because it aims at populating the Offering Catalogue with datasets any Participant can access for free, the Open Data enabler itself is actually a Participant in the SEDIMARK ecosystem, acting solely as an Offering Provider. As such, the Open Data enabler does not need any components to be installed on the Participants premises to work. It is hosted on Atos premises. It can therefore be continuously improved during the lifetime of SEDIMARK, for instance by adding new Offerings, with no impact on its Participants: the changes will only be reflected in the Catalogue. Besides, having a Participant dedicated to the development of one of SEDIMARK's component is an advantage for its consortium, offering an additional platform to test them.

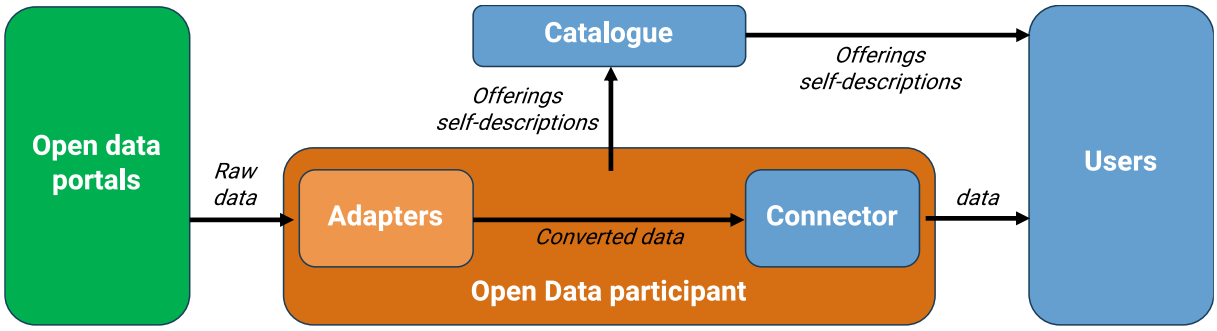

Figure 41 - Open Data enabler architecture (©SEDIMARK)

Figure 41 depicts the working principle of the Open Data enabler. As a Participant, the Open Data enabler is equipped only with the minimal set of components required to be a Provider participant in the SEDIMARK ecosystem, namely:

- **A DLT Booth**, to manage its identity and expose its DID document and SEDIMARK membership verifiable credential.
- **A Connector**, to enable contract negotiation and data transfers between the Open Data Enabler and other participants.
- **An Offering Manager**, to manage its own Offerings and publish them in the Catalogue.

Other SEDIMARK components, such as AI and data processing toolboxes, as well as the Marketplace frontend, aren't necessary for it to run.

To fetch data from open data portals, it relies on *adapters*: their role is to ensure that any open dataset to be exposed in the SEDIMARK ecosystem is wrapped into an Offering which can be published in the SEDIMARK Catalogue. An example implementation of such adapter can be found in the SEDIMARK GitHub organization [33].

9.2 Exposing DCAT based datasets

The adapter provided in the SEDIMARK GitHub organization is designed to expose datasets based on the DCAT (Data Catalog) model [34] by providing a simple Python Flask client [35], consisting of an endpoint to create SEDIMARK compatible Offerings directly from the URLs pointing to such DCAT datasets. The generated Offerings can then be published in the Catalogue by the Open Data Enabler operator. Example of published Offerings includes datasets from the Santander City Council [36] or the Salted project [37].

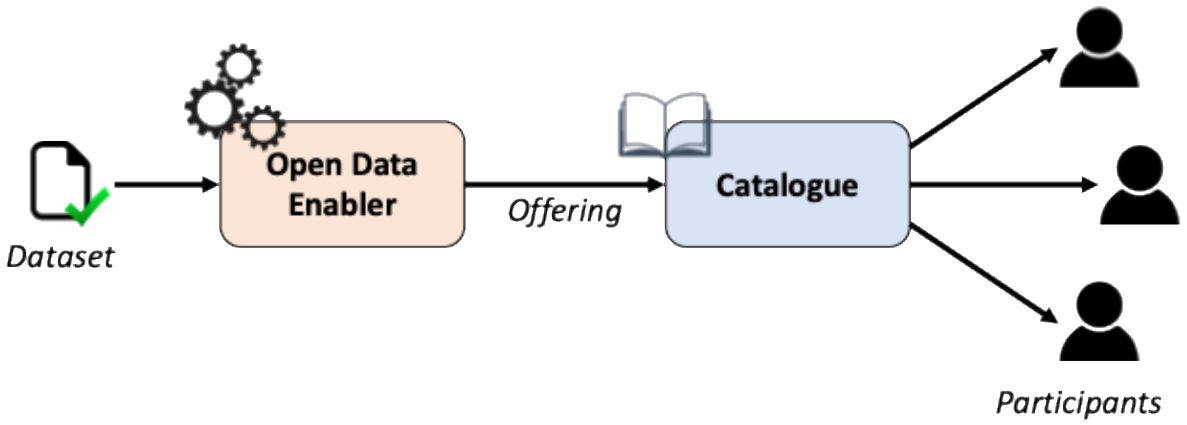

Figure 42 - Open Data Enabler Offering Publication Workflow (©SEDIMARK)

10 Data sharing incentives

This chapter describes how SEDIMARK intends to foster data sharing in its marketplace through two approaches. On the one hand, use cases operators will act as participants in the SEDIMARK ecosystem to showcase its functionalities: section 10.1 gives such an example with the city of Helsinki. On the other hand, section 10.2 presents some initial ideas of features to be implemented in the marketplace frontend to enable users to emphasize the quality of their datasets.

10.1 City of Helsinki as a Participant in the Marketplace

SEDIMARK's core features will be continuously evaluated through its 4 use cases, with a special attention to include:

- Connecting remote data platforms seamlessly for enhanced collaboration and insights.
- Enabling efficient and privacy-preserving data sharing while ensuring data security and confidentiality.
- Offering diverse, high-quality, certified data and services to meet various industry needs and standards.
- Supporting the EU's Common Data Spaces initiative for fostering innovation and digital transformation in the EU.

All pilot sites will contribute to enrich the Marketplace, acting as Participants in the SEDIMARK ecosystem, in order to provide high quality public offerings showcasing its catalogue to prospective users.

The City of Helsinki will be one of such pioneer Participants, demonstrating how the SEDIMARK marketplace can be used by stakeholders in city mobility to foster data sharing, activating engagement and collaboration, making use of Forum Virium's expertise guiding projects to co-create smart city innovations that enhance urban residents' quality of life while minimizing environmental impact. It will at first provide dataset offerings based on its mobility data platform [38]. As the SEDIMARK project goes on, it will not only expose its data APIs as service offerings, but also keep collaborating with various stakeholders to create new projects and pilots, aiming at improving the usability and usefulness of data, as well as developing tools for a mobility digital twin to enable the testing and development of new smart mobility solutions in real urban environments.

All information resources are managed in accordance with the FAIR model, which defines the principles for the fair and efficient utilization of data in an organization [39], to ensure the high quality and interoperability of the provided data.

Hackathons serve as a key incentive tool within the smart city ecosystem to foster innovation, collaboration, and problem-solving. By bringing together multidisciplinary teams to tackle concrete challenges in decentralized data marketplaces, hackathons provide a dynamic environment for skill development, networking, and career advancement. They also act as platforms for prototyping new concepts, enabling participants to rapidly test and iterate solutions using real-world data and tools.

In the context of SEDIMARK, hackathons support the co-creation of a shared understanding of the data marketplace business model — including clearer definitions of operator roles, responsibilities, and technical requirements. This paves the way for future collaboration with

various stakeholders. The hackathon can be organized either virtually or locally on-premises, with respect to available program resources.

10.2 Marketplace frontend features to foster data sharing

The first version of the SEDIMARK marketplace is a proof-of-concept demonstrating a data space supported by the IOTA tangle as a Distributed Ledger Technology (DLT). This unique characteristic makes it a flagship platform to incentivise data sharing between its participants.

On the one hand, the DLT ensures that Offerings, as well as the contracts binding its providers and consumers, are recorded in an immutable registry and therefore transparently traceable. On the other hand, the SEDIMARK Connector [40] is based on the Eclipse Dataspace Components Connector [41]: it uses the same Data Space Protocol API and therefore facilitates the potential interaction of the SEDIMARK marketplace with other data spaces [42].

SEDIMARK also stands out for its added value tools, which lies in two areas:

- **The Marketplace**: through an integrated recommender system, users can be advised on relevant Offerings depending on the search they do in the SEDIMARK Catalogue and the Offerings they select when they browse it.
- **The SEDIMARK toolbox [43]**: aside from Offering publication and consumption, SEDIMARK comes with a toolbox providing data processing and AI orchestration components to its participants, tailored to produce Assets directly publishable in the marketplace.

Yet, the SEDIMARK platform offers many perspectives to further foster data sharing among its Participants. Its ontology [44] has been designed to welcome data quality metrics (see section 3.5 of Deliverable SEDIMARK_D3.2 [10]), in order to augment Offerings to be published with valuable information about the quality of the datasets. Data processing tools can therefore be augmented to integrate an assessment of such quality and update the dataset's metadata accordingly. Then, these indicators could be displayed in the Offering representations in the Marketplace user interface to inform prospective Consumers.

The integration between the data processing toolbox and the Marketplace can also further be improved to foster data sharing among Participants, by enabling automations depending on the *type* of Asset contained in a consumed Offering. As part of the work done in WP3, SEDIMARK assets can be of several types, depending on what they represent. For instance, beyond the generic *Data Asset*, other types such as *AI Model Asset* can represent AI models or pipelines created by a Provider in her/his Mage AI orchestrator. In such cases, upon consuming such an Asset, the Marketplace could offer another option on top of downloading the data: it could for instance directly ingest the new AI model in the Mage AI instance of the consumer, so he/she can seamlessly start using her/his new model without manually uploading it.

11 Conclusions

This report recaps the final approach to the architecture of the SEDIMARK Marketplace and data sharing platform and hence describes its main objective, as well as introduces its constitutive modules and describes their functionalities within the SEDIMARK platform.

SEDIMARK built a decentralised marketplace supported by the IOTA tangle as a Distributed Ledger Technology, in order to encourage users from diverse backgrounds to exploit and benefit from its various functionalities. On top of such data sharing marketplace, a toolbox provides different techniques to help cleaning, protecting, discovering and even enriching the offerings, thus paving the way for diverse businesses and research scenarios to make use of them and get in return a significant profit.

To achieve such objective, the current deliverable presented the modules developed to shape the SEDIMARK Marketplace. The exercise starts with a recap of the basic actions covered by such platform as well as with the presentation of a brief analysis on the users expected to appear in there.

The Marketplace acts as an entry point for SEDIMARK participants to perform every necessary step to onboard the ecosystem, as well as browse the Catalogue of Offerings and interact with other participants, by providing or consuming Offerings through contracts' negotiations. This deliverable described how all these operations can be performed in the Marketplace user interface. It starts by a step-by-step guide to explain how new users can perform their registration and be welcomed into SEDIMARK.

The presentation continues with the introduction into the Catalogue of Offerings that is presented in the Marketplace and where any user, whether registered or simple visitors, have the chance to browse and discover the Offerings available in SEDIMARK. The actions performed by Participants in the Catalogue shape how the Recommender system, a specificity of the SEDIMARK platform, advises them on what Offerings they may be interested in. This Recommender deserves a section of its own and thus the report delivers a detailed explanation on how it works, which are its main design guidelines, and how data flows within and from it. Eventually, a depiction of how SEDIMARK implements the Recommender appears, alongside a summary of the initial results obtained in the test processes conducted to validate the Recommendation models.

To understand the aforementioned Offerings, the report includes a complete go through what they are and how they can be published in the Catalogue. This includes all metadata to describe the Assets they represent, as well as their usage policies and licensing. Authenticated users can then further manage their provided or consumed Offerings via a specific dashboard, as well as trigger and monitor corresponding data transfers.

Willing to submit the most complete experience possible, the Marketplace also sets up secondary dashboards that lead to data processing orchestration and to build up AI/ML pipelines for model training, both federated and distributed.

SEDIMARK's vision also involves dealing with diverse data sources. To make it a reality the Open Data enabler plays a relevant role since it is in charge of promoting data sharing in the marketplace by providing free and open datasets to all participants. This report explains it architectural description and describes its implementation, stressing out the absence of any dedicated component to be installed on the participants' premises.

Finally, the document reflects on some perspectives on how the SEDIMARK platform can be improved to further incentivise data sharing among its participants.

12 References

[1] “D4.5 Data sharing platform and incentives - First version”, [Online]. Available: https://sedimark.eu/wp-content/uploads/2024/01/D4.5-Data-sharing-platform-and-incentive%E2%80%93First-version.pdf

[2] “D4.2 Decentralized Infrastructure and Access Management - Final Version”.

[3] “D4.4 Edge data processing and service certification – Final version”.

[4] “D2.3 SEDIMARK Architecture and Interfaces. Final version”.

[5] “D5.4 Integrated releases of the SEDIMARK platform. Final version”.

[6] “D5.6 Demonstrators integration, testing and assessment of system performance. Final version”.

[7] “D2.2 SEDIMARK Architecture and Interfaces. First version,” p. 83.

[8] “SEDIMARK News & Events,” SEDIMARK. Accessed: Sep. 22, 2025. [Online]. Available: https://sedimark.eu/news/

[9] “ODRL Information Model 2.2.” Accessed: Dec. 15, 2023. [Online]. Available: https://www.w3.org/TR/odrl-model/#policy

[10] “D3.2 Energy efficient AI-based toolset for improving data quality. Final version”.

[11] S. Milano, M. Taddeo, and L. Floridi, “Recommender systems and their ethical challenges,” *Ai & Society*, vol. 35, pp. 957–967, 2020.

[12] Y. Koren, S. Rendle, and R. Bell, “Advances in Collaborative Filtering,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., New York, NY: Springer US, 2022, pp. 91–142. doi: 10.1007/978-1-0716-2197-4_3.

[13] D. Bridge, M. H. Göker, L. McGinty, and B. Smyth, “Case-based recommender systems,” *The Knowledge Engineering Review*, vol. 20, no. 3, pp. 315–320, 2005.

[14] “Zenodo.” Accessed: Sep. 29, 2025. [Online]. Available: https://zenodo.org/

[15] R. Sharifpour, M. Wu, and X. Zhang, “Large-scale analysis of query logs to profile users for dataset search,” *Journal of Documentation*, vol. 79, no. 1, pp. 66–85, Apr. 2022, doi: 10.1108/JD-12-2021-0245.

[16] M. Grootendorst, “KeyBERT: Minimal keyword extraction with BERT. Zenodo; 2020.”

[17] O. Kramer, *Machine learning for evolution strategies*, vol. 20. Springer, 2016.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” 2019.

[19] I. Beltagy, K. Lo, and A. Cohan, “SciBERT: A Pretrained Language Model for Scientific Text.” 2019.

[20] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” 2019.

[21] Susheel, *Sentence Similarity*. (Nov. 16, 2023). Python. Accessed: Dec. 15, 2023. [Online]. Available: https://github.com/Susheel-1999/Sentence_Similarity

[22] G. T. Reddy *et al.*, “An Ensemble based Machine Learning model for Diabetic Retinopathy Classification,” in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–6. doi: 10.1109/ic-ETITE47903.2020.235.

[23] D. K. Po, “Similarity based information retrieval using Levenshtein distance algorithm,” *Int. J. Adv. Sci. Res. Eng*, vol. 6, no. 04, pp. 06–10, 2020.

[24] S. E. Robertson and K. S. Jones, “Relevance weighting of search terms,” *Journal of the American Society for Information Science*, vol. 27, no. 3, pp. 129–146, May 1976, doi: 10.1002/asi.4630270302.

[25] V. Lavrenko and W. B. Croft, “Relevance-based language models,” in *ACM SIGIR Forum*, ACM New York, NY, USA, 2017, pp. 260–267.

[26] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, “Latent semantic indexing: A probabilistic analysis,” in *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 1998, pp. 159–168.

[27] “Gensim: topic modelling for humans.” Accessed: Dec. 15, 2023. [Online]. Available: https://radimrehurek.com/gensim/models/lsimodel.html

[28] V. Viswanathan, L. Gao, T. Wu, P. Liu, and G. Neubig, “DataFinder: Scientific dataset recommendation from natural language descriptions,” *arXiv preprint arXiv:2305.16636*, 2023.

[29] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How To Backdoor Federated Learning,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, S. Chiappa and R. Calandra, Eds., in Proceedings of Machine Learning Research, vol. 108. PMLR, Aug. 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[30] C. Xie, K. Huang, P.-Y. Chen, and B. Li, “DBA: Distributed Backdoor Attacks against Federated Learning,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:213447399

[31] E. Duriakova *et al.*, “PDMFRec: A Decentralised Matrix Factorisation with Tunable User-Centric Privacy,” in *Proceedings of the 13th ACM Conference on Recommender Systems*, in RecSys ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 457–461. doi: 10.1145/3298689.3347035.

[32] K. Wei *et al.*, “Federated Learning With Differential Privacy: Algorithms and Performance Analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020, doi: 10.1109/TIFS.2020.2988575.

[33] *Sedimark/open-data-enabler*. (Sep. 22, 2025). Python. Sedimark. Accessed: Sep. 23, 2025. [Online]. Available: https://github.com/Sedimark/open-data-enabler

[34] R. Albertoni, D. Browning, S. J. D. Cox, A. G. Beltran, A. Perego, and P. Winstanley, “Data Catalog Vocabulary (DCAT) - Version 3.” W3C Recommendation, Aug. 22, 2024. [Online]. Available: https://www.w3.org/TR/vocab-dcat-3/

[35] “Welcome to Flask — Flask Documentation (3.1.x).” Accessed: Sep. 23, 2025. [Online]. Available: https://flask.palletsprojects.com/en/stable/

[36] “Datos Abiertos Santander |.” Accessed: Sep. 23, 2025. [Online]. Available: http://datos.santander.es/

[37] “Welcome - CKAN SALTED.” Accessed: Sep. 23, 2025. [Online]. Available: https://ckan.salted-project.eu/

[38] “Data Catalog,” Mobility Lab Helsinki. Accessed: Dec. 21, 2023. [Online]. Available: https://mobilitylab.hel.fi/data/

[39] M. D. Wilkinson *et al.*, “The FAIR Guiding Principles for scientific data management and stewardship,” *Scientific Data*, vol. 3, no. 1, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18.

[40] “Sedimark/sed-connector: Connector for integrating data providers and consumers with the SEDIMARK marketplace. Extends EDC Connector functionality and bridges external systems with the SEDIMARK decentralized components.” Accessed: Sep. 23, 2025. [Online]. Available: https://github.com/Sedimark/sed-connector

[41] *EDC Connector*. (Dec. 06, 2023). Java. Eclipse Dataspace Components. Accessed: Dec. 08, 2023. [Online]. Available: https://github.com/eclipse-edc/Connector

[42] "Dataspace Protocol 2024-1 | IDS Knowledge Base." Accessed: Sep. 23, 2025. [Online]. Available: https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol

[43] "Sedimark/Sedimark-Toolbox: Deployment for Sedimark Toolbox to deploy an instance of it and connect to the Sedimark Marketplace." Accessed: Sep. 23, 2025. [Online]. Available: https://github.com/Sedimark/Sedimark-Toolbox/tree/main

[44] "Sedimark/ontology: SEDIMARK ontology documentation." Accessed: Sep. 23, 2025. [Online]. Available: https://github.com/Sedimark/ontology