SEcure Decentralised Intelligent Data MARKetplace

D2.2 SEDIMARK Architecture and Interfaces. First version

Document Identification	
Contractual delivery date:	30/09/2023
Actual delivery date:	29/09/2023
Responsible beneficiary:	NUID UCD
Contributing beneficiaries:	All
Dissemination level:	PU
Version:	1.0
Status:	Final

Keywords:

Data marketplace, decentralisation, interoperability, machine learning, distributed ledger technology, data quality, requirement analysis, functional architecture, system interfaces

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. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

Document Information

Document Identification			
Related WP	WP2	**Related Deliverables(s):**	-
Document reference:	SEDIMARK_D2.1	**Total number of pages:**	109

List of Contributors	
Name	**Partner**
Elias Tragos Aonghus Lawlor Diarmuid O'Reilly Morgan Erika Duriakova Honghui Du Neil Hurley	NUID-UCD
Arturo Medela Maxime Costalonga Jairo Rojas-Delgado	ATOS
Luc Gasser Léa Robert Romain Magnani Franck Le Gall Philippe Cousin Gilles Orazi	EGM
Nikolaos Georgantas Maroua Bahri	INRIA
Andrea Vesco Alberto Careli	LINKS
Nikos Babis Ioannis Tsogias	MYT
Gabriel-Mihail Danciu Iuliana Stroia-Vlad	SIE

List of Contributors	
Name	**Partner**
Tarek Elsaleh Adrian Hilton	SURREY
Luis Sánchez Jorge Lanza Pablo Sotres Juan Ramón Santana	UC
Panagiotis Vlacheas Panagiotis Demestichas Grigoris Koutantos Loizos Koutsantonis Konstantinos Almpanakis Dimitrios Triantafyllou	WINGS

Document History			
Version	**Date**	**Change editors**	**Change**
0.1	13/02/2023	NUID-UCD	First draft of ToC
0.11	27/06/2023	NUID-UCD	Initial draft of Section 5
0.2	30/06/2023	UC	Initial version of Section 3
0.21	04/07/2023	NUID-UCD	Contributions to Sections 2, 6.1
0.22	11/07/2023	NUID-UCD	Contributions to Sections 6.1, 6.2, 6.3
0.23	12/07/2023	SIE	Contributions to Sections 3, 6.12, 6.6
0.24	17/07/2023	MYT	Contribution to Section 5
0.25	19/07/2023	WINGS	Contribution to Section 5
0.3	25/07/2023	NUID-UCD	Updates to Section 5
0.31	26/07/2023	SURREY	Contributions to Section 6.8
0.32	27/07/2023	UC	Contributions to Sections 4, 5, 6.7, 6.13
0.33	28/07/2023	INRIA	Contributions to Sections 4, 5, 6.4, 6.6
0.34	28/07/2023	SIE	Contributions to Section 5
0.4	31/07/2023	UC	Updates to Section 3
0.41	31/07/2023	EGM	Contribution to Sections 5, 6.4

Document History			
Version	Date	Change editors	Change
0.5	04/08/2023	SURREY	Contributions to Sections 5, 6.6
0.51	08/08/2023	INRIA	Contribution to Sections 6.4, 6.6
0.52	09/08/2023	UC	Contributions to Section 7
0.53	10/08/2023	ATOS	Contribution to Sections 3, 6.11, 6.12
0.54	10/08/2023	ATOS	Contributions to Sections 3, 6.11, 6.12
0.6	18/08/2023	NUID-UCD	Contributions to Section 7
0.61	22/08/2023	UC	Contributions to Section 6.13
0.62	22/08/2023	WINGS	Contributions to Section 6.5
0.7	25/08/2023	INRIA	Updates to Section 5, 6.4, 6.6
0.71	28/08/2023	ATOS	Contribution to Section 8
0.72	29/08/2023	SIE	Updates to Section 3
0.73	29/08/2023	WINGS	Updates to Section 6.5
0.8	01/09/2023	NUID-UCD	Contributions to Sections 1, 10 and Executive summary
0.81	01/09/2023	SURREY	Updates to Section 6.6
0.82	01/09/2023	LINKS	Contributions to Sections 6.9, 6.10, 7
0.9	04/09/2023	ATOS	Contributions to Sections 4, 7
0.91	04/09/2023	UC	Contributions to Section 9
0.92	04/09/2023	INRIA	Updates to the sections 5, 6.4, 6.6
0.93	04/09/2023	LINKS	Contributions to Section 5
0.94	05/09/2023	NUID-UCD	First consolidated version, fixing formatting.
0.95	06/09/2023	MYT	Contribution to Section 6.4
0.96	07/09/2023	UC	Updates to Sections 7,9
0.97	07/09/2023	NUID-UCD	First version for internal review
0.98	15/09/2023	NUID-UCD	Addressed comments of the internal review
0.99	20/09/2023	NUID-UCD, INRIA, UC, SURREY	Final fixes based on internal review comments

Document History			
Version	**Date**	**Change editors**	**Change**
0.991	22/09/2023	ATOS	Quality Review Form
0.992	26/09/2023	NUID-UCD	Quality Revisions
0.999	27/09/2023	ATOS	Quality Review Form 2
1.0	29/09/2023	ATOS	FINAL VERSION TO BE SUBMITTED

Quality Control		
Role	**Who (Partner short name)**	**Approval date**
Reviewer 2	Gilles Orazi (EGM)	15/09/2023
Reviewer 1	Nikolaos Georgantas, Maroua Bahri (INRIA)	15/09/2023
Quality manager	María Guadalupe Rodríguez (ATOS)	22/09/2023
Project Coordinator	Arturo Medela (ATOS)	29/09/2023

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
BDVA	Big Data Value Association
CKAN	Comprehensive Knowledge Archive Network
CLEC	Cross Domain Low Energy Computing
CPS	Cyber Physical Systems
CSMT	Compact Sparse Merkle Tree
DAPS	Dynamic Attribute Provisioning Service
DCAT	Data Catalog Vocabulary
DID	Decentralized Identifiers
DIH	Digital Innovation Hub
DLT	Distributed Ledger Technology
DSBA	Data Space Business Alliance
DSSC	Data Spaces Support Center
Dx.y	Deliverable number y belonging to WP x
EC	European Commission
EDC	Eclipse Data Space Connector
EU	European Union
FAIR	Findability Accessibility Interoperability Reusability
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
IBFT	Istanbul Byzantine Fault Tolerance
IDS	International Data Spaces
IDSA	International Data Spaces Alliance
IEC	International Electrotechnical Commission

Abbreviation / acronym	Description
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IOTA	Internet of Things Application
ISO	International Standards Organisation
JSON-LD	JavaScript Object Notation for Linked Data
MBB	Model Building Block
ML	Machine Learning
NFR	Non-Functional Requirement
NFT	Non Fungible Token
NGSI-LD	Next Generation Service Interfaces - Linked Data
ODRL	Open Digital Rights Language
ONNX	Open Neural Network Exchange
RAM	Reference Architecture Model
RS	Recommender Systems
SGAM	Smart Grid Architectural Model
SME	Small Medium Enterprise
SQL	Standard Query Language
SSI	Sensitive Security Information
TLS	Transport Layer Security
UI	User Interface
UML	Unified Modelling Language
VC	Verifiable Credential
VDI	Verifiable Database Integrity
WP	Work Package

Executive Summary

Data marketplaces and data spaces have become a new trend aiming to provide platforms so that companies and researchers can exchange datasets in a secure way. Data has become a new currency and it is crucial to be able to get access to high quality datasets in order to quickly, easily and accurately extract knowledge towards meeting your objectives. Most such existing data sharing platforms are though centralised, gathering all datasets in the cloud or some central servers, without providing high privacy or without giving providers or consumers the necessary supplies to assess or improve the quality of the datasets they exchange. SEDIMARK aims to change the domain of data marketplaces contributing to the decentralisation of data exchange platforms, providing their users with the needed tools for improving data quality and building knowledge upon the data.

This document presents the first complete version of the SEDIMARK functional and system architecture, aiming to provide details on what functionalities the SEDIMARK platform will provide and how these will interact in order to meet the main objectives of the project. SEDIMARK builds upon the concepts of trust, decentralisation, interoperability, data quality and intelligence in order to provide a fully decentralised data and services marketplace, where providers and consumers will be able to share their data and build knowledge upon them.

Before presenting in detail the system architecture, this document provides a complete list of the terms and concepts defined and used within the project so that the readers can understand how these terms are used within the context of SEDIMARK. The main actors are also defined, mainly split into (i) providers, who are providing data, ML models, or services and (ii) consumers, who are consuming the assets that are provided.

This deliverable leverages on the results of the SEDIMARK deliverable D2.1 [1], which presented the main project use cases and the list of functional and non-functional requirements. These requirements are analysed with respect to the functionalities that the project platform should support towards realising the project objectives and this led to the definition of the functional components of the system and the design of the functional view of the SEDIMARK architecture. Upon performing this exercise, SEDIMARK defined 10 main functional enablers, grouping the functional components with respect to the functionalities they provide. Thus, the functional enablers within SEDIMARK provide tools for (i) data processing, (ii) data curation, (iii) distributed storage, (iv) AI processing, (v) DLT connectivity, (vi) data space connectivity, (viii) marketplace services, (viii) trustworthiness, (ix) interoperability, and (x) open data connectivity.

This document also presents the main internal and external interfaces of the SEDIMARK platform, providing a first view on how the components will interact with each other and how the main services will be provided. For the latter, example data flows are also presented, showing the messages exchanged between key components, towards providing the service. An initial system view of the architecture also shows how the platform can be instantiated in the real world.

This first version of the SEDIMARK architecture aims to provide guidelines and solutions for designing decentralised data space and marketplaces architectures for researchers and engineers and also ideas and concepts to EU initiatives (i.e. the DSSC) so that common reference architectures can be built. The second and final version of the SEDIMARK architecture, after receiving feedback from the technical workpackages will be presented in the next deliverable D2.3 in September 2024.

1 Introduction

1.1 Purpose of the document

This document serves as the first version of the SEDIMARK architecture aiming to provide an in-depth description of various architectural views and the roadmap on how the views were extracted. The main goal is to provide an architecture that supports the main concepts of SEDIMARK for full decentralisation, trustworthiness, intelligence, data quality and interoperability. This document is considered as one of the main deliverables of the project, because the main technical and evaluation activities will be based on the description of the functional components of the architecture and their interactions. SEDIMARK follows the concept of agile innovation-driven methodology for the development of the decentralised marketplace. This means that the architecture document should be considered as a "live" document that will be continuously updated and improved as the technical development and testing activities of the rest of the workpackages will evolve, aiming to identify omissions or issues with the initial architecture draft, so that these can be fixed by adapting the components or by adding missing components and removing components that are either not useful or duplicated. A new version of the SEDIMARK architecture will be provided in Month 24 (September 2024) in the Deliverable D2.3.

Considering that this document does not provide a fully functional architectural framework, but rather only a high-level document presenting initial concepts and ideas (not tested), it is intended for a limited audience, primarily for the project consortium to use it for driving the technical activities of the project in the rest of the work packages. Additionally, other researchers and developers in the areas of interest of the project will also find interesting ideas about developing decentralised data and services marketplaces. Moreover, EU initiatives and other research projects should consider the contents of the deliverable in order to help derive common architectures and concepts for creating data spaces and building marketplaces on top of them focusing on improved trustworthiness, data quality and intelligence.

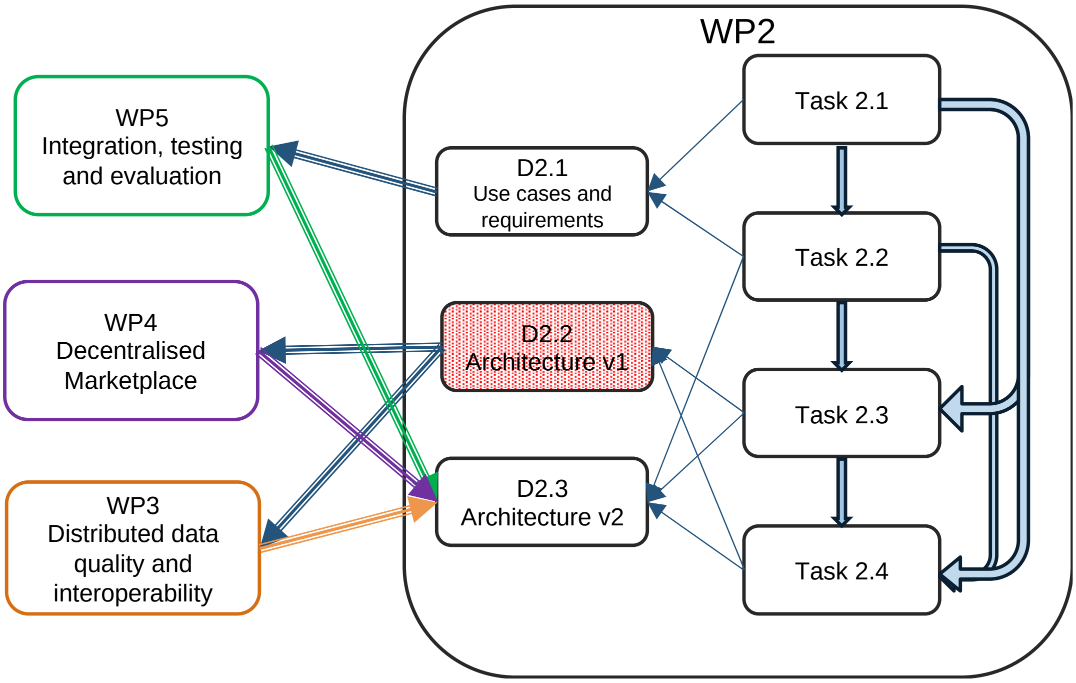

WP2
WP5
Integration, testing and evaluation
WP4
Decentralised Marketplace
WP3
Distributed data quality and interoperability
D2.1
Use cases and requirements
D2.2
Architecture v1
D2.3
Architecture v2
Task 2.1
Task 2.2
Task 2.3
Task 2.4

Figure 1: Relationship between D2.2 and other deliverables, tasks, and workpackages.

1.2 Relation to another project work

This deliverable is the output of work done in the first year of the project in Task 2.3. Figure 1 shows the interaction of the activities within WP2 and the relation with the rest of the work packages. As it is depicted, the work presented in this deliverable is based on the outputs of the work done in Tasks 2.1 (project use cases) and 2.2 (system requirements) and was presented in Deliverable D2.1. More detail about the methodology to derive the project architecture is given in Section 2. The output of the work presented in this deliverable will also be used as input to the rest of the work packages for driving their activities for the development and testing. Additionally, the rest of the work packages will provide feedback to Task 2.3 and D2.2 for the next cycles of the architecture development in order to improve the architecture and provide the final version of the document in Deliverable D2.3.

1.3 Structure of the document

This document is structured in 10 major chapters:

Chapter 1 is the current chapter and presents the introduction to the document.

Chapter 2 presents the methodology that SEDIMARK followed in order to derive the system architecture.

Chapter 3 presents some key EU initiatives and projects related to data spaces and data marketplaces that SEDIMARK used as baselines for building the concepts and the architecture that is presented in this deliverable.

Chapter 4 presents the glossary for the terms and concepts that are used throughout this document and that will be used in the rest of the project deliverables. Chapter 4 also presents the interconnections and the relationships between the concepts and terms.

Chapter 5 presents the results of the activity of requirement analysis based on the requirements presented in Deliverable D2.1 [1]. The goal of this activity is to analyse the requirements with respect to the SEDIMARK objectives and extract the needed functionalities and functional components that should be included in the functional architecture.

Chapter 6 presents the functional view of the SEDIMARK architecture, with the description of the functional enablers and the functional components and their interactions.

Chapter 7 presents a first description of some example data flows for some services provided by SEDIMARK, showing how the different components of the architectures interact in order to provide these example services.

Chapter 8 presents a first view on the interfaces within the SEDIMARK platform and between SEDIMARK and external parties/modules.

Chapter 9 presents the system view of the SEDIMARK architecture.

Chapter 10 presents the conclusions of the document discussing the major outcomes and the future steps.

1.4 Glossary adopted in this document

Due to the importance of the glossary in the definition of the SEDIMARK architecture, there is a dedicated section (Section 4) that provides a complete view of the terms and concepts defined and used within SEDIMARK.

2 Methodology for designing the SEDIMARK architecture

As discussed in Deliverable D2.1 [1], SEDIMARK adopts a use case driven approach for designing the overall system and the system architecture. In this respect, the system will be designed initially based on the specific use cases at hand and then generalised so that it can cater for more diverse scenarios. A general process for designing software architectures is described in [2]. As discussed there, the architecture design process is an iterative one and involves several steps that have to be repeated in order to produce an optimal architecture design in the end. Also, the architecture design process includes the definition of various Architectural Views, which represent several structural aspects of an architecture, showing how it addresses the concerns of the stakeholders. The core architectural views defined in [2] are:

- **Context view**: describes the relationships and the interactions between the various system elements.
- **Information view**: describes the way the system handles information and the related data structures.
- **Functional view**: describes the system's functional elements, interfaces and interactions and is considered the cornerstone of the architectural views.
- **Development view**: supports the software development process for building, testing and maintaining the system.
- **Deployment view**: describes how the system will be deployed, the hardware needs and how the software is mapped to the runtime environment.
- **Operational view**: describes how the system will be operated, administered and supported when it is in production.
- **Concurrency view**: describes the elements of the system that can be running concurrently.

The process of designing the architecture of SEDIMARK follows the guidelines described in [2] for describing the architectural views. However, SEDIMARK will not put an emphasis right now on the *Operational* and the *Concurrency* views, since the project doesn't aim to operate the system in a production environment. This deliverable will focus on the *Functional* view, the *Information* view and the *Context* view, while the *Deployment* and *Development* views will be left for the deliverables of WP5.

The SEDIMARK methodology for designing the system architecture is inspired by both [2] and [3] and includes the following steps:

- Definition of the project use cases.
- Analysis of the use cases towards the project objectives;
- Analysis of security and privacy threats on the use cases;
- Definition of the information and context models;
- Definition of the system requirements (both functional and non-functional);
- Analysis of the requirements towards the project objectives;
- Extraction of the system's functional components and functional groups;

- Identification of the interconnectivity between the system entities and the interfaces that interconnect them;
- Extraction of the deployment view, towards placement of the functional groups on physical components;
- Refinement of the architecture based on input and feedback from the technical WPs;

The overall process is also shown in Figure 2.

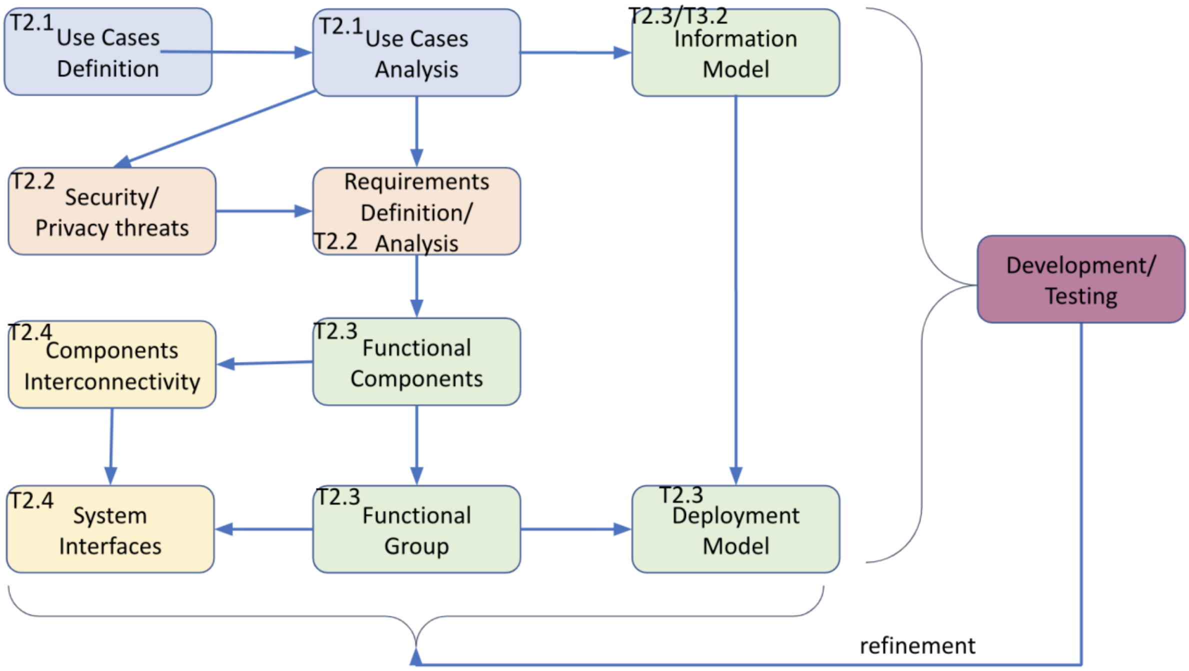

T2.1 Use Cases Definition
T2.1 Use Cases Analysis
T2.3/T3.2 Information Model
T2.2 Security/ Privacy threats
Requirements Definition/ T2.2 Analysis
Development/ Testing
T2.4 Components Interconnectivity
T2.3 Functional Components
T2.4 System Interfaces
T2.3 Functional Group
T2.3 Deployment Model
refinement

Figure 2: SEDIMARK process for designing the architecture

SEDIMARK follows a bottom-up and use-case driven architecture design approach. The project's use cases play a major role in the design of the system, that's why the first step of the overall process is the detailed definition of the project use cases. This was done in SEDIMARK deliverable D2.1 [1], where the four main project use cases were defined in detail, describing the objectives, the components, the stakeholders, the services to be offered, the data to be generated, the flows of information, the challenges and the key performance indicators. This initial definition of the use cases gives insights into the key points of interest of each use case with respect to the SEDIMARK objectives and how the project objectives can serve the use case requirements.

The second step of the process is the analysis of the use cases in order to identify the key attributes that a system should have in order to meet the objectives of the use cases. This process is usually done together as part of the requirement extraction process that was described in SEDIMARK Deliverable D2.1 [1]. In a parallel step, an initial trust and security analysis is also done in order to identify in general the assets that need to be protected and the ways to provide trust in the overall system.

The next step of the process is to define the Context view and the initial domain and information models, describing the main elements of the SEDIMARK system (as part of the project glossary) and how these are interconnected in order to serve the purposes of the architecture.

The security analysis and the use case analysis are used to drive the requirement definition process that was described in SEDIMARK Deliverable D2.1 [1]. This activity resulted in the definition of 9 major categories of functional requirements. This leads into the process of analysis of the requirements in terms of their importance and their mapping towards the project objectives. This is done based on the requirements categories of D2.1 and based on their mapping to the architectural layers (which will be described in Section 6) aiming to identify the needs of the SEDIMARK functional architecture in terms of functional components. The requirement analysis process will extract, as a result, the functional components for each of the categories of the requirements, and these components will correspond to specific functionalities that will be developed in the technical work packages of SEDIMARK. The functional components and their definition are given in Section 6 in this deliverable.

After extracting the functional components, the next step in the process is to group them to functional groups and define the way these groups are interconnected defining an initial set of the internal system interfaces. Additionally, an initial version of the external interfaces will also be defined, as open interfaces to make the connection of SEDIMARK with external projects easy and open.

Finally, the deployment model of SEDIMARK will be defined in WP5, aiming to provide details on how the architecture can be instantiated in various use case scenarios and how the functional components/groups can be mapped to physical devices.

The architecture definition process is an iterative one, that will provide the architecture in two steps: (i) an initial version at the end of the first year of the project (which is presented in this current deliverable) and (ii) the final version at the end of the second year of the project (to be presented in deliverable D2.3), following a refinement step with feedback given back to the architecture definition as the development and testing activities are done in the technical work packages. The refinement process has to be seen as a continuous process which evolves as the technical activities in the project mature.

3 State of the art architectures for marketplaces

Nowadays, there are multiple alternatives providing a framework for decentralised marketplaces. Particularly, we can highlight:

- IDS (International Data Spaces) [4], a standard promoted by the IDSA (International Data Spaces Association), which provides a secure decentralised framework for the creation of data spaces to exchange data assets.
- Gaia-X [5], an initiative that provides a framework to enable a decentralised ecosystem to exchange data and service assets based on a common trust framework.
- I3-MARKET [6] is addressing the gap between trusted and secure solution for federation of data marketplaces.
- SMART4ALL [7] offers a concept of Market-as-a-Service, reducing the development effort, e.g., to move from an idea to prototype, required by startups or mid-caps that create products for various industry domains.
- OMEGA-X [8] aims to implement a data space entirely focused on the energy domain. This will include federated infrastructure, data marketplace and service marketplace, involving data sharing between different stakeholders and demonstrating its value for concrete energy use cases while guaranteeing scalability and interoperability with other data space initiatives.

Such initiatives have many synergies that can be exploited towards the creation of a single standard that could enable a truly decentralised market within Europe. To this end, a new association called DSBA (Data Spaces Business Alliance) [9] has been created, and includes as members the FIWARE, IDSA, Gaia-X and BDVA (Big Data Value Association) associations, which are currently working on decentralised Data Spaces and marketplaces.

3.1 International Data Spaces (IDS)

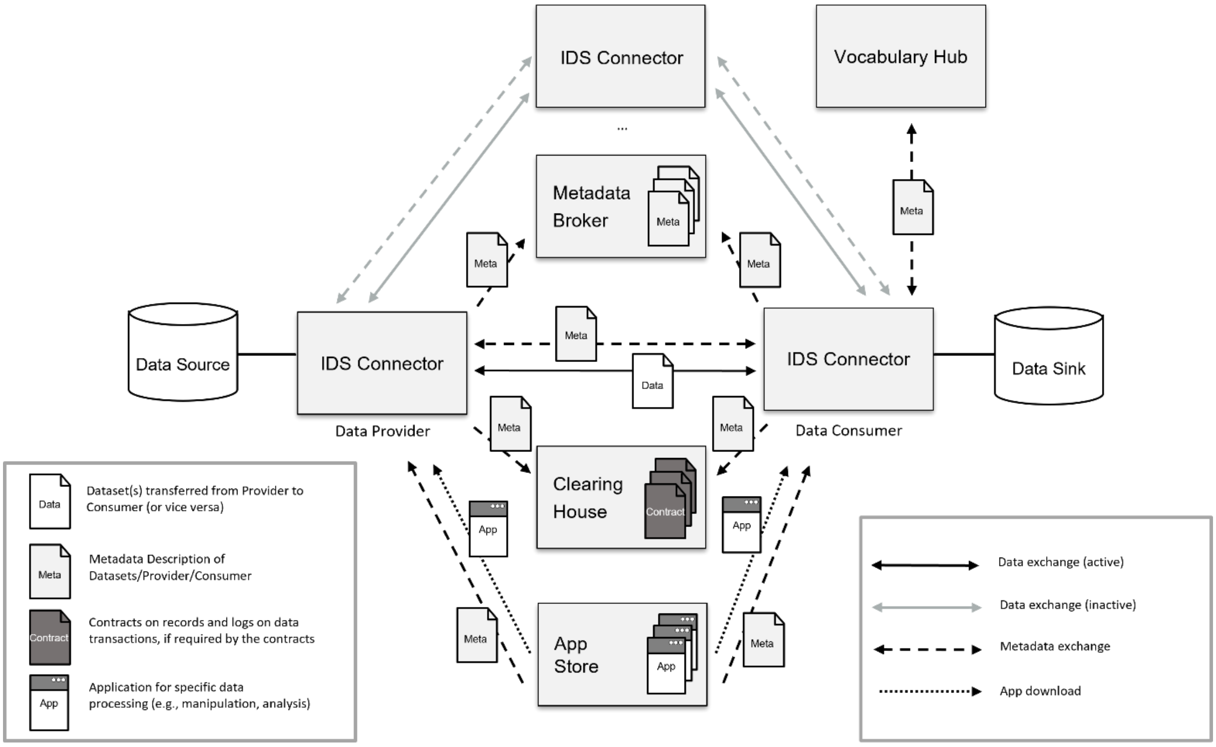
IDS is the standard created by the IDSA to enable decentralised and self-sovereign data spaces that allow the exchange of data between different participants. The conceptual architecture is described in the document IDS-RAM (IDS Reference Architecture Model) that provides the foundations and components required for establishing a data space. The latest version available at the moment of writing this deliverable is the 4.2 [4]. The technical specification of the components, as well as the protocols enforced in a data space is defined by the IDS-G (IDS Global), that can be found in a GitHub repository [10].

The core component of the IDS architecture is the Connector. Connectors are essential for any interaction within a data space, as they are responsible for establishing secure and trustworthy communication between participants. To ensure a reliable and regulated data exchange, participant Connectors are required to sign legally binding contracts that define the terms and conditions under which data can be accessed and utilized. These contracts encompass access and usage control policies, ensuring that data remains protected throughout its lifecycle. To reinforce this trust, all components that interact within a data space must be certified, guaranteeing that technical enforcement of agreements and adherence to the architecture's defined processes are upheld. The rest of the required components for a data space based on IDS are described below:

- **Identity Provider:** This component facilitates the identification, authentication, and authorization of IDS participants within a data space. It ensures that only authorized individuals and entities gain access to the data and services offered by the IDS ecosystem. It includes the Dynamic Attribute Provisioning Service (DAPS) server, that authenticates the different participants in the data space.

- **Metadata Broker**: Acting as a reliable data discovery engine, the Metadata Broker enables data providers to register their assets. Participants can leverage this component to search and locate relevant data sources, ensuring efficient data discovery and retrieval.
- **Clearing House**: As a logging service, the Clearing House plays a crucial role in recording and storing transactional information between Connectors. It maintains a comprehensive record of data contracts and agreements, providing transparency and accountability for all data exchanges within the IDS ecosystem.
- **Vocabulary Hub**: The Vocabulary Hub serves as a centralized repository that grants participants access to domain-specific vocabularies. By extending the basic common Information Model [9], IDS provides a common language for participants to describe and understand data, fostering seamless communication and interoperability across different domains.
- **App Store**: Within each Connector, the App Store serves as a secure repository of software applications that can be deployed. These applications enhance the functionality and capabilities of Connectors, enabling customization and adaptability to meet specific data processing and analysis requirements.

All of them conform the architecture described in the IDS-RAM, depicted in Figure 3. Besides any interaction in a dataspace by these components must follow the IDS Information Model [11].



IDS Connector
Vocabulary Hub
...
Metadata Broker
Meta
Data Source
IDS Connector
Data Provider
IDS Connector
Data Consumer
Data Sink
Data
Clearing House
Contract
App
App Store
Dataset(s) transferred from Provider to Consumer (or vice versa)
Metadata Description of Datasets/Provider/Consumer
Contracts on records and logs on data transactions, if required by the contracts
Application for specific data processing (e.g., manipulation, analysis)
Data exchange (active)
Data exchange (inactive)
Metadata exchange
App download

Figure 3: IDS Reference Architecture Model

3.1.1 Data Space Protocol

As part of the evolution of the IDS standard, IDSA is currently working on the provision of a new protocol for data spaces, named Data Space Protocol, currently in its 0.8 version [12].

The Data Space Protocol is an ongoing work, which is born to support the existing technical development of the EDC (Eclipse Data Space Connector). Therefore, both developments are being carried out in parallel.

This protocol includes the specifications to define the interactions between components, and is meant to be the main technical specification for the IDS-RAM. In particular, the current version of the protocol is organized in 4 main topics, specified in different documents, that serves as the basis for the main interactions between participants in a data space:

- Data Space Model and Terminology: it defines the ontologies and taxonomies to enable the interoperability between participants.
- Catalog Protocol: defines how datasets are published and accessed, providing the necessary common interfaces. In the Data Space Protocol, datasets are represented following the Data Catalog Vocabulary (DCAT) ontology.
- Contract Negotiation Protocol: defines the set of interactions required to conduct contract negotiations between two participants, ensuring that both participants agree on the terms for access and control rules. Contracts are represented following the Open Digital Rights Language (ODRL) ontology.
- Transfer Process Protocol: defines how the data exchange is carried out once an agreement has been reached between two participants. Differently from the previous specifications from the IDS-G, no protocols are defined for this exchange, but the control of the transfer process.

All of the specifications rely on the use of the Connector as the core component of a data space, using HTTPS to define all the interfaces for the different interactions. Besides, two planes are defined: the control plane, which is covered by the Data Space Protocol Specification; and the data plane, which is only in charge of the data exchange.

3.2 Gaia-X

The Gaia-X association is established with the primary objective of developing a comprehensive framework, accompanied by policies and rules, to facilitate the creation of federated cloud services across various cloud-based service providers. Similar to IDSA, Gaia-X places significant emphasis on the principles of data sovereignty and trust. Not only does Gaia-X enable the decentralization of data-related services, but it also extends this concept to encompass infrastructure services, leveraging self-sovereign identities based on W3C Verifiable Credentials [13].

A distinguishing feature of Gaia-X is the adoption of self-descriptions for all services and participants within its ecosystem, utilizing verifiable credentials. These verifiable credentials are digitally signed by trusted entities, affirming the accuracy and validity of the claims made within them. To establish a trusted environment, Gaia-X employs trust anchors and establishes chains of trust based on them. Gaia-X primarily focuses on digitizing the comprehensive description of all essential elements required for cloud services and data exchange, standardizing the vocabulary and the mandatory components of self-description. The overarching goal of Gaia-X is to empower participants by returning control over their data and fostering a fairer landscape for cloud service providers. Gaia-X's primary focus is on the meta level, encompassing the description of services, participants, and data to be exchanged, while the actual data exchange itself is considered beyond its scope. For instance, Gaia-X acknowledges a Connector, as defined by IDSA, as one of the potential implementation technologies for data exchange but does not delve into the specifics of data exchange mechanisms.

The foundations for Gaia-X are specified in the Gaia-X Architecture Document [14], which defines the core concepts required to enable a Gaia-X Ecosystem. Hence, the document defines the minimum requirements for any participant, along with the required essential services to enable the ecosystem.

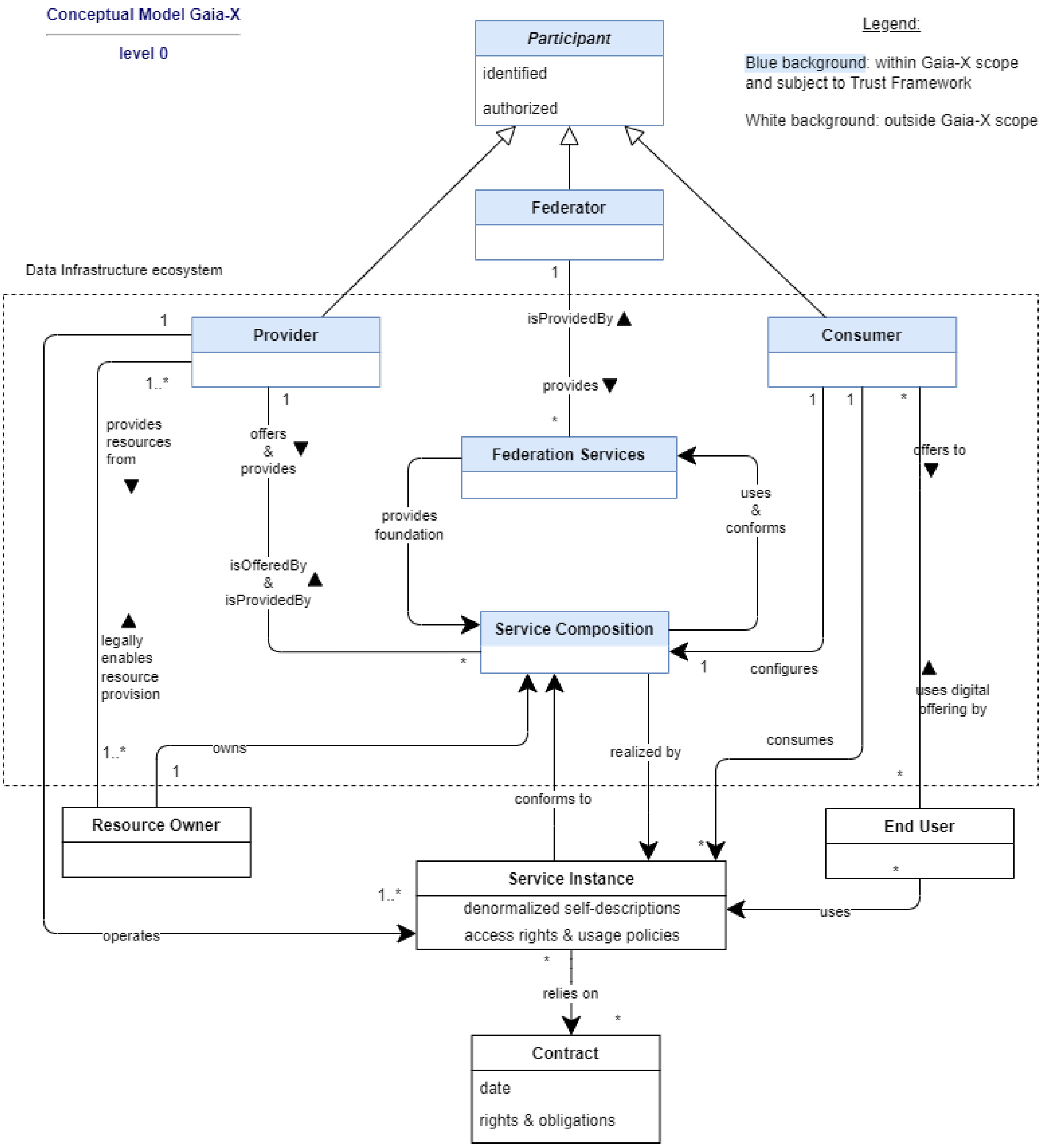


Figure 4: Gaia-X Conceptual Model Level 0

As shown in Figure 4, which represents the conceptual model for the entities that are part of a Gaia-X Ecosystem, there are three types of Participants: Consumer, Provider and Federator. The first two types, similarly to other data space initiatives such as IDS, represent the entities that exchange the resources in a transaction, while the federator is in charge of enabling the ecosystem, providing the minimum services to do so. These services are listed as follows:

- Federated Catalogue: enables the discovery of existing services through an indexed repository of self-descriptions, which is populated through an inter-catalogue synchronization.
- Identity and Access Management common vocabulary: to cover all the security-related processes (authentication, identification, authorization and credential management) within an ecosystem.
- Data Exchange services: enable data exchange between participants, providing the services to enable Data Agreements on resources, as well as the logging of transactions. Thus, Providers can enforce their terms and conditions.
- Gaia-X Trust Framework: is composed of the mechanisms to ensure Participants adhere to the policy rules in an ecosystem.
- Portal and APIs: enable the Participant onboarding process and the management of existing Participants in the ecosystem. It also provides service discovery, orchestration and sample services.

3.3 Data Space Support Centre (DSSC)

DSSC is a recently funded project by the EGI, Open Ecosystem for Research and Innovation, coordinated by the Fraunhofer institute, which include several initiatives related to the creation of data spaces, such as Gaia-X, IDSA or BDVA. The main goal of the project is to set up and operate a support centre for data spaces, aligned with the European Strategy for Data. Hence, the DSSC is created to establish shared data spaces that collectively foster an interoperable environment for data sharing. This will allow reusing data in several sectors while upholding EU values. The following are the main benefits for public administrations and businesses, according to their goals [15]:

- Enabling the availability of technologies, processes, legal frameworks, standards, and tools (e.g. Community of Practice, Blueprint) for the deployment of data spaces;
- Fostering the adoption of above technologies and standards to enable the reuse of data across sectors by different stakeholders with a multidisciplinary approach based on co-creation and interaction;
- Contributing to the generation of sustainable and scalable products and services for the global market leveraging the use of shared data in business model development or in efficient, effective and repeatable policy decision-making.
- Ensuring that more data becomes available for use in the economy and society, while keeping those who generate the data in control.
- Facilitating the sharing of data, hereby creating a positive impact in the daily lives of citizens and giving confidence to businesses and public administrations.

The DSSC is also defining its own Conceptual Model, which defines and organizes the fundamental concepts and terms associated with data spaces. This Conceptual Model is at the core of its specifications and aims at establishing a standardized vocabulary that ensures clear communication and shared understanding among stakeholders, so that consistent interpretation is enabled and ambiguity is reduced when discussing data spaces across different contexts.

In Figure 5 at the highest level of DSSC's Conceptual Model, the Basic Interactions of DSs and the ecosystem perspective have been defined. A significant role is given to the DS Governance Framework (DSGF) which controls at least one registry that contains all data that is needed for the data space to function as intended. The DSGF is the set of principles, standards, policies (rules/regulations) and practices that apply to the governance, management, and operations within a data space. This includes the ways in which its contents are maintained and further developed, as well as the ways in which they are enforced, and how any conflicts are being resolved. The rest of actors and core elements (specifically the DS Participants and the Products provided and consumed within the DS) are arranged around these two elements. Moreover, the federation of Data Spaces, which was not evident in other initiatives has been incorporated.

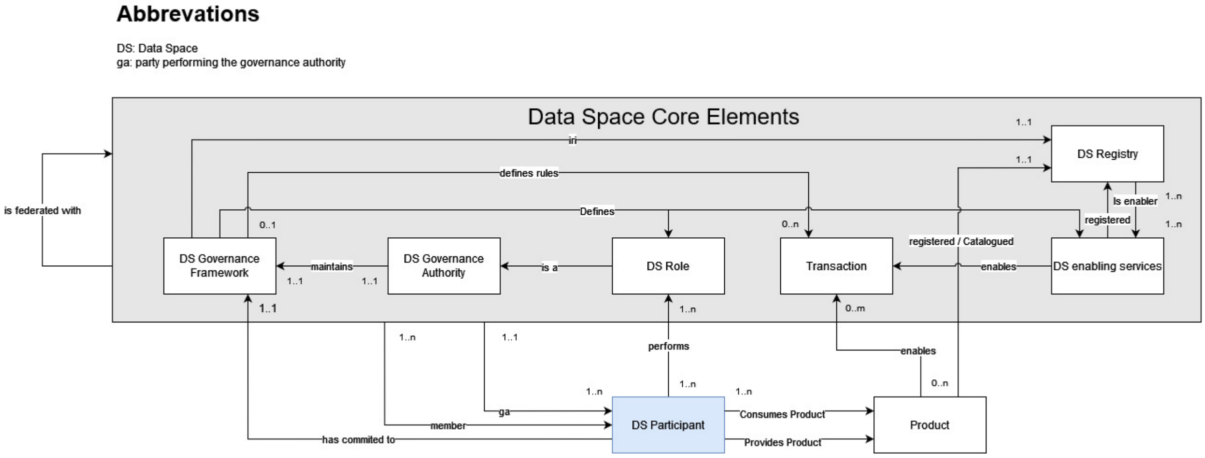


Figure 5: DSSC Conceptual Model Level 1

The DSSC is promoting a definition for a data space, which says that: “A Data Space is an infrastructure that enables data transactions between different data ecosystem parties based on the governance framework of that data space” [16]. A data space should be generic enough to support the implementation of multiple use cases. In order to realize such a definition, in each data space present today, and the ones to be developed in the future, several building blocks need to be considered when setting up: they delineate areas where choices are required to enable effective and trusted sharing of data among participants. From a technical standpoint, the DSSC Blueprint is composed of several Building Blocks that are arranged around three main pillars: Interoperability, Trust and Data value.

In terms of Interoperability, the DSSC holds that data spaces should provide a solid framework for efficient data exchange among participants, supporting the complete decoupling of data providers and consumers, making data services FAIR (Findable, Accessible, Interoperable and Reusable). This requires the adoption of a “common lingua” every participant uses, which materialised in adopting interoperable APIs for data exchange and the definition of compatible data models. Interoperable mechanisms for the traceability of data exchange transactions and data provenance are also needed.

In terms of Trust, DSSC is proposing that data spaces should bring technical means for guaranteeing that participants in a data space can trust each other and exercise sovereignty over the data they share based on user-controlled consent. This requires the adoption of interoperable standards for managing the identity of participants, verifying their trustworthiness, and enforcing policies agreed upon for data access and usage control.

Finally, in terms of Data value, the DSSC is promoting that data spaces should provide possibilities for participants to generate value from sharing data (i.e., going beyond FAIRness towards data quality and further on to creating data value chains). Therefore, data spaces often can contain multi-sided markets if participants intend to trade, buy, and sell data services as part of their business model. This requires the adoption of interoperable mechanisms enabling the description of terms and conditions (including pricing) linked to data service offerings, the publication and discovery of such offerings and the management of all the necessary steps supporting the lifecycle of contracts that are established when a given participant acquires the rights to access and use data.

As part of the first steps, the DSSC will create different assets throughout the project lifetime, which are all publicly shared. At the time of writing this deliverable, the DSSC has already released the first version of the Starter Kit [17], a short guidance document for the creation of data spaces; and a Glossary [18], with a set of terms related to data spaces, to avoid any ambiguity.

Finally, the DSSC is also promoting annual events and a community forum to create a network of stakeholders involved in data space initiatives.

3.4 i3-MARKET

The i3-MARKET project [6] is a comprehensive data management initiative aiming to create a secure and robust single European data market. It uses a federation of data marketplaces to allow secure and privacy-preserving data sharing across multiple platforms. This project primarily focuses on industrial data, addressing the lack of trusted and secure solutions for the federation of data marketplaces.

The data storage system in i3-MARKET has a two-fold architecture, incorporating both decentralized and distributed storage. These storage systems handle data like identity information, shared semantic models, meta-information about datasets, semantic queries, and smart contract templates. They are implemented through technologies such as Hyperledger Besu and CockroachDB.

The decentralized storage system, built on Hyperledger Besu, operates as a blockchain-based distributed ledger network. It manages distributed identities and smart contracts while also storing linked blocks and world state information. This system uses a permissioned setup with IBFT 2.0 consensus and allows the execution of smart contracts.

On the other hand, the distributed storage system uses CockroachDB, a distributed SQL database that provides a SQL interface to other i3-MARKET framework components. This system manages data synchronization among different i3-MARKET nodes and offers auditable accounting features.

For security purposes, both systems ensure that no single party fully controls the data storage system, eliminating a single point of failure. Additionally, authentication and authorization solutions are in place, relying on TLS server endpoints and client certificates. Moreover, they guarantee end-to-end security between the distributed storage service and its client services.

The distributed storage system takes appropriate measures to ensure its availability as it is critical to the functioning of the i3-MARKET network. For instance, independent clusters are deployed at each i3-MARKET instance, so issues at one instance do not affect the data at others.

Furthermore, the Verifiable Database Integrity (VDI) has been implemented to ensure the integrity of the data it contains. It employs a Compact Sparse Merkle Tree (CSMT) and an API for data management and verification of membership/non-membership proofs against the Merkle tree. The VDI is integrated into the Auditable Accounting component to ensure the secure and auditable recording of data and transactions.

The i3-MARKET project aims to break down barriers in data marketplaces by developing necessary components for interoperability and integration. The architecture of i3-MARKET is built upon trusted, decentralized, and federated software modules that allow for integration with other marketplaces. It's designed to enable secure and privacy-preserving data sharing across different data spaces and marketplaces by deploying a backplane across operational data marketplaces.

The architecture provided by i3-MARKET lets data providers and consumers share or trade data in a way that is open, fair, and interoperable. Open APIs are used in the design principle of i3-MARKET, which will be made available as an open-source project to ensure all future enhancements can be managed effectively. The architecture will be rigorously tested on use cases related to industrial, manufacturing, and wellbeing marketplaces. These use cases stand to gain from the capability to share or trade data across various marketplace instances.

Currently, there is a gap in the market as no scalable, trustworthy, and interoperable solution exists that facilitates sharing or trading of data assets across different marketplace instances. Moreover, modern data marketplace platforms are not equipped to exchange sensitive industrial data assets, as they do not yet support the required levels of security and privacy, especially not across ecosystem boundaries, in accordance with the GDPR. The i3-MARKET architecture addresses this issue by incorporating blockchain-like technologies by design into the foundational elements of the i3-MARKET architecture.

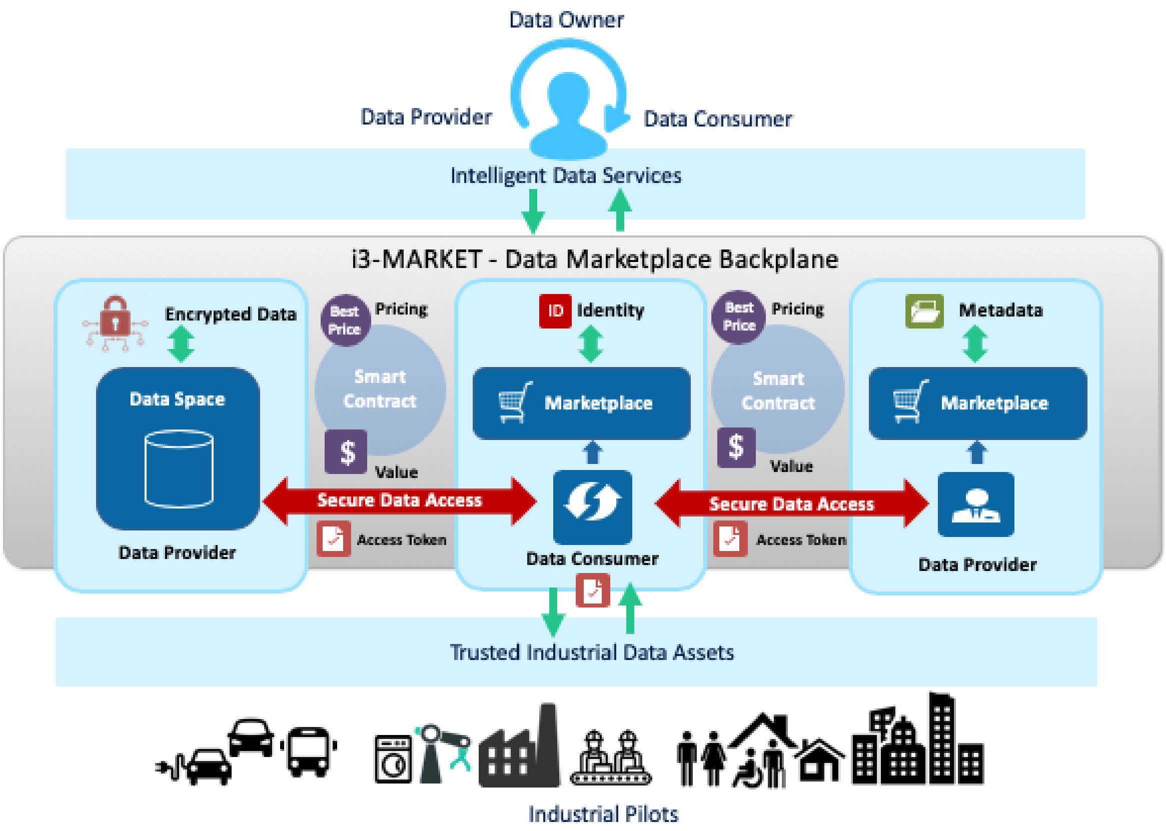


Figure 6: i3MARKET Architecture

As can be seen in Figure 6, the i3-MARKET architecture incorporates several key components to create an open, trusted, and interoperable data marketplace:

- Smart Contracts: This provides access to semantic descriptions of offered data assets, enabling data discovery across various silos. It promotes federation among different data spaces and marketplaces without needing centralized control or coordination.
- Access Tokens: These provide a transparent, cost-efficient, and quick payment solution for trading data assets. This system incentivizes data providers to offer their assets, thereby accelerating the European data economy. The tokens can also be used as internal payment within participating data spaces and marketplaces.
- Secure Semantic Data Model Repository: This allows data consumers to efficiently discover and access data assets through precise semantic queries, and integrate data into their applications/services based on a common understanding of data meaning. It allows independent data providers and consumers to exchange and use data in a meaningful way without prior information exchange.
- Trusted Industrial Data Assets: Trust is ensured through secure and trusted APIs that enable data spaces and marketplace providers to obtain identities, register data assets, fetch semantic descriptions, create and sign smart contracts, and make payments. The architecture also includes immutable and auditable smart contracts for data asset trading across data space and marketplace boundaries. These contracts must be signed by all stakeholders, including data providers, consumers, and owners.
- Legal Framework: This establishes the contractual basis for smart contracts and cryptocurrency tokens, and defines innovative business models for incentivising sharing and trading of data assets.

3.5 SMART4ALL

SMART4ALL [7] (Self-sustained customized cyberphysical system experiments for capacity building among European stakeholders) is an initiative that aims to bridge the digital divide among various regions of Europe by facilitating the transfer of knowledge and technology between academia and industry. Focusing on Cross-Domain Low-Energy Computing (CLEC) for Cyber-Physical Systems (CPS) and the Internet of Things (IoT), the initiative is set to deliver a range of unique qualities that bring together different cultures, policies, geographical areas, and application domains under a single, comprehensive vision.

The SMART4ALL consortium includes 25 partners from Central, South, and Eastern Europe. Many of these areas have historically been under-represented in European funding schemes and have lacked sufficient Digital Innovation Hubs (DIHs) to assist companies with their digital transformation processes. The initiative aims to bolster high-end research and development in South-eastern Europe (SEE) for CLEC CPS and IoT by facilitating community building, strategic development, and ecosystem learning.

SMART4ALL supports areas like digitized environment, digitized agriculture, digitized transport, and other fields that aren't adequately represented in current Smart Anything Everywhere (SAE) projects. A key component of SMART4ALL is the development of a new concept of Marketplace as a Service (MaaS), which acts as a one-stop-smart-shop offering tools, services, and platforms, primarily based on open-source technologies. The MaaS also provides adopter matchmaking capabilities tailored to the four thematic pillars of the project.

The initiative aligns with the New Skills Agenda for Europe and the Digital Skills and Jobs Coalition by collaborating with state companies, social partners, non-profit organizations, and education providers. These parties work together to tackle the digital skills shortage across Europe. Furthermore, SMART4ALL aims to support sensitive social groups by ensuring that innovation and technology positively impact their lives.

SMART4ALL's approach is to create a unified European innovation hub that can compete on a global scale. This is achieved by leveraging its DIH network, creating capacity-building opportunities, and offering novel marketplace services. The initiative has also designed an open-call strategy for Pathfinder Application Experiments (PAEs) to support innovation and accelerate the growth of SMEs and startups.

The SMART4ALL Marketplace-as-a-Service (MaaS) platform is one of its central features. This platform offers a variety of tools and services designed to accelerate the design, development, prototyping, and manufacturing phases of a start-up or SME, providing them with the necessary support to access funds and market growth opportunities.

SMART4ALL's focus is on four under-represented areas: Digitized transport, Digitized Environment, Digitized Agriculture, and Digitized Anything. By promoting solutions that offer high computing capacity and low energy consumption, these sectors are expected to gain a competitive advantage.

The initiative supports various application experiments, which include Knowledge Transfer Experiments (KTEs), Focused Technology Transfer Experiments (FTTEs), and Cross-domain Technology Transfer Experiments (CTTEs). It also plans to establish bilateral DIH-based communication channels between market and innovation producers, like SMEs, academic institutions, and research centres.

Finally, the SMART4ALL model is designed to ensure the sustainable development and growth of the supported experiments and the DIH cluster post-project. It seeks to influence policy-making in line with its vision and extend its reach beyond the project duration through continued engagement and active community building.

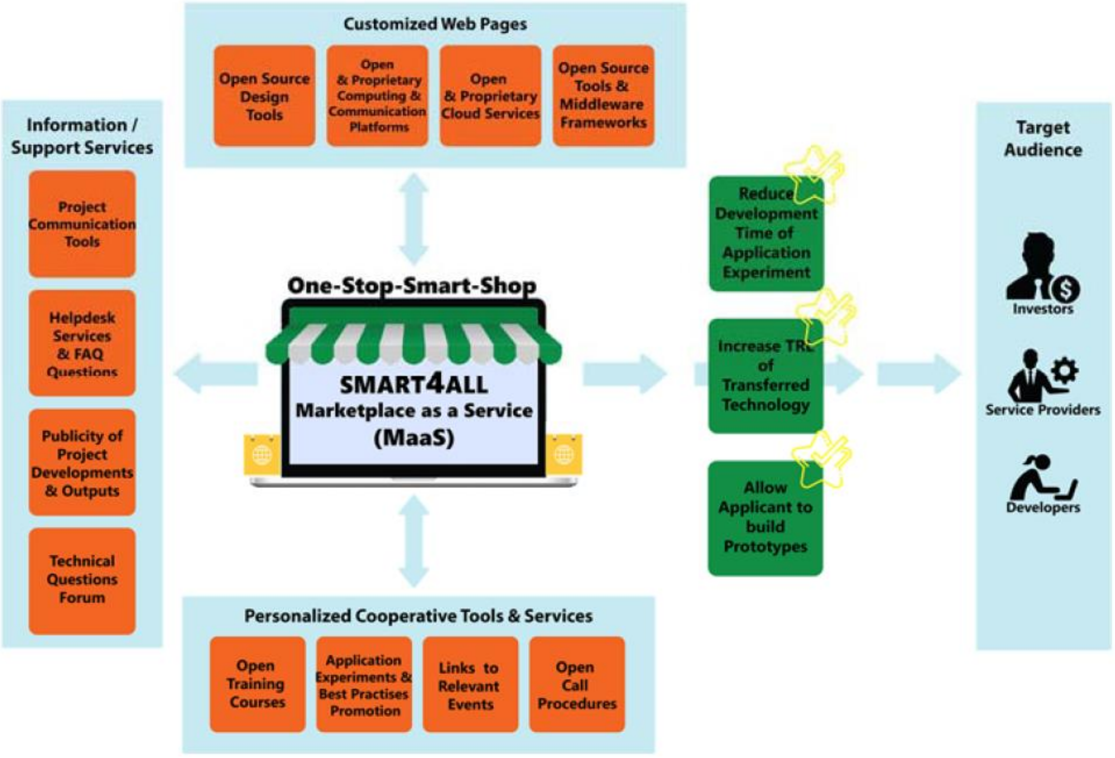

Customized Web Pages
Open Source Design Tools
Open & Proprietary Computing & Communication Platforms
Open & Proprietary Cloud Services
Open Source Tools & Middleware Frameworks
Information / Support Services
Project Communication Tools
Helpdesk Services & FAQ Questions
Publicity of Project Developments & Outputs
Technical Questions Forum
One-Stop-Smart-Shop
SMART4ALL Marketplace as a Service (MaaS)
Reduce Development Time of Application Experiment
Increase TRL of Transferred Technology
Allow Applicant to build Prototypes
Target Audience
Investors
Service Providers
Developers
Personalized Cooperative Tools & Services
Open Training Courses
Application Experiments & Best Practises Promotion
Links to Relevant Events
Open Call Procedures

Figure 7: The SMART4ALL concept

The vision for SMART4ALL's proposed marketplace is to transform it from a traditional one-stop-shop to an innovative one-stop-smart-shop via a concept known as Marketplace-as-a-Service (MaaS). The MaaS platform presented in Figure 7, is designed to foster the transfer of knowledge and technology, boost collaborative research and development, and facilitate industry-oriented academic education.

Simultaneously, MaaS serves as the central hub for publishing and sharing continuous updates on the project's advancements, targeting all interested collaborators. It primarily aims to streamline the development process, helping stakeholders such as startups, SMEs, and mid-caps transition from initial ideas to competitive prototypes, particularly in the four key SMART4ALL sectors: Digitized Transport, Digitized Agriculture, Digitized Environment, and Digitized Anything.

MaaS encompasses a variety of cloud services, platforms, tools, middleware frameworks, and design service facilities, with a focus on open-source technologies. In addition, it offers AI-based personalized services, such as tailored links to pertinent events, custom web pages, and matchmaking activities between technology providers and receivers.

The SMART4ALL Marketplace-as-a-Service (MaaS) offers a range of services to third parties, with a special focus on open technologies. The MaaS platform is specifically designed to cater to each thematic pillar and incorporates a query mechanism to facilitate easy search operations for both expert and non-expert users.

Beyond technical offerings, MaaS also provides project and financial management tools, along with access to all relevant websites. It hosts cloud services, related platforms, tools, middleware frameworks, and design service facilities, primarily emphasizing open-source technologies. The platform also offers AI-based personalized services, such as tailored links to pertinent events, custom web pages, and matchmaking activities between technology providers and receivers.

A significant portion of the project's efforts will be dedicated to assessing the readiness of various open-source technologies and creating helpful application notes on their correct usage in startups, SMEs, and mid-cap companies.

A vital aspect of the SMART4ALL marketplace is its matchmaking service, which aims to connect technology providers from academia and industry with adopters. The matchmaking service allows registered MaaS users to request specific thematic connections. Initially, this service is handled by experts with a comprehensive understanding of the network in Europe and potential partner companies for specific requests. Later stages of the project will see the incorporation of AI in this service to offer cognitive support in finding a match by combining various parameters.

3.6 OMEGA-X

The main approach of OMEGA-X implies the combination of the benefits that both IDSA and Gaia-X bring to the table, but also extend their guidelines to create a unique and complete Data Space architecture dedicated to energy related applications which in turn accommodates the project’s Use Case Families.

The methodology adopted by the OMEGA-X project follows the principles of view and viewpoints and is heavily influenced by architectural models such as SGAM (Smart Grid Architectural Model) [19], ISO/IEC/IEEE 42010:2011 [20] and Viewpoint and Perspectives [2], which comprises up to six viewpoints, namely Functional, Information, Concurrency, Development, Deployment and Operational, along with three optional ones, Security, Performance and Scalability.

Moreover, the IDSA Reference Architecture Model [4] is as well an influence of OMEGA-X. From the five layers proposed there, the project focuses so far on the following:

- High-level description of all architectural components according to the System Layer.
- Detailed description of each component, its responsibilities, and interfaces as per the Functional Layer.
- Depiction of procedures that takes place in the platform and interactions among components, as suggested in the Process Layer.

Going into design matters, various standards influenced the outcome, being the most relevant ones the GAIA-X specifications, the IDSA architecture and distributed processing considerations.

In addition, there is always three main technical challenges that Data Spaces of any kind must deal with: data interoperability, data sovereignty and trust, and data value creation, which in turn exercise an influx over design considerations.

Additional factors considered for the definition of the architecture of OMEGA-X are as follows:

- GAIA-X Digital Clearing House (GXDCH) [5]: it offers non-exclusive, interchangeable nodes operated by market operators to verify the Gaia-X rules.
- Eclipse Data Space connector [21]: it provides a framework for sovereign, inter-organizational data exchange, implementing the international Data Spaces Standard (IDS) as well as relevant protocols associated with Gaia-X.
- Vocabulary: Looking for interoperability not only at Data Space components level but also dealing with smart energy business services, the employment of a common vocabulary is a must, thus enabling the semantic and syntactic representation of energy services domain data.
- Decentralized versus centralized operation of Data Space components and services: It must be clarified whether the proposed solution relies on a central catalogue provider, or such a catalogue is a decentralized one. The very same discussion hovers on the services the Data Space enables.

As depicted in Figure 8 the OMEGA-X high-level systemic architecture comprises 4 sections:

- The Data & app marketplace, which acts as the main entry point for external users thanks to its graphical interface. It incorporates interfaces with the Identity Provider to proceed with registration and login functions, and with the Federated Catalog to enable the management of self-descriptions. It also provides a platform to negotiate contracts between offering providers and consumers.
- The Federated Infrastructure, which provides mechanisms for secure and sovereign data exchanges.
- The connectors that enable data exchanges and service provision. The idea for them is to operate either on-premises or in a cloud environment and must provide interfaces to the Federated Catalogue, the Data Exchange Services, and the Identity Management.
- A series of Compliance Services that enable trust and interoperability through the validation of the shape, content, and credentials of self-descriptions, while at the same time check the compliance with required fields and consistency rules of the Gaia-X Trust Framework.

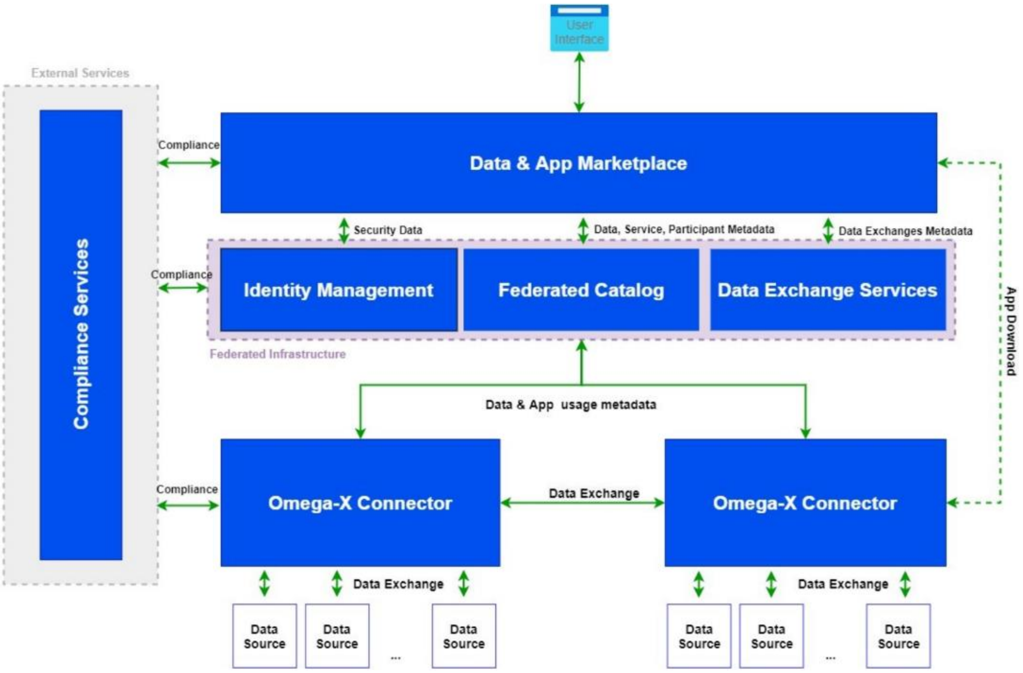


Figure 8: OMEGA-X architecture

As a main objective, the OMEGA-X marketplace aims to provide diverse kind of service offerings. On the one hand, AI services that process heterogeneous input data from various data sources and as a result provide results in different shapes (e.g., text, numeric or pure visuals). On the other hand, digital twins that simulate the behaviour of real-life assets, processes and/or systems to get specific results, also in diverse forms.

OMEGA-X offers the aforementioned models in two ways. One of them comes as a containerised model implemented at the Connector level, where an image (typically in Docker) can be downloaded and executed in the Connector itself; this way the computation shifts to the data source. The alternative implies a Software as a Service (SaaS) solution where exchanges of raw or processed data go through the Connector.

4 Terminology and domain model

Prior to the formal definition of the SEDIMARK architecture, a common understanding of the key terminology used throughout this document is required. In this sense, this section includes the initial version of the SEDIMARK glossary, including all the necessary terms and concepts as well as their definitions.

Aiming at reusability, the concepts have also been mapped to those provided by Gaia-X and IDS, trying to be as much aligned as possible.

The contents of this section will be used throughout all the SEDIMARK Work Packages, adopted in every project-related content, both internal to the consortium and externally when disseminating SEDIMARK via forums and events.

4.1 Terminology

All definitions in this section are delimited within the scope of SEDIMARK project. Besides, underscored terms refer to those also defined in this section, and are linked for the sake of reading.

4.1.1 Marketplace

An ecosystem composed of a set of Baseline Infrastructure Facilitators and a set of Participants (using their respective Toolboxes) that adhere to the functional and system architecture defined by SEDIMARK project to enable a secure decentralised Offerings' exchange.

4.1.2 Marketplace Reference Implementation (RI)

The reference implementation and deployment of a Marketplace provided within the SEDIMARK project to support the four use cases that will be demonstrated within the project. In this particular Marketplace instance, the set of Baseline Infrastructure Facilitators will be composed by different partners of the SEDIMARK consortium.

4.1.3 Participant

Participants are members of a Marketplace that act either as Providers or Consumers. Participants are identified through a digital identity (e.g. SSI) and are represented through Self-descriptions. Participants commerce with Offerings and, as a result, exchange Assets.

4.1.4 Baseline Infrastructure Facilitator

The Baseline Infrastructure Facilitators (BIFs) instantiate the minimum required building blocks to enable a Marketplace and support the trust among participants.

4.1.5 Provider

A Participant of a Marketplace that provides Assets represented by Offerings and defines terms and conditions to regulate their trading/procurement through Contracts.

4.1.6 Consumer

A Participant of a Marketplace who searches Offerings and consumes Assets represented within them.

4.1.7 Asset

A particular resource offered by a Provider. Assets can be further processed/enhanced by either the Data Processing Pipeline or the AI pipeline before being offered within a Marketplace. Further information regarding asset classification is depicted in Figure 9.

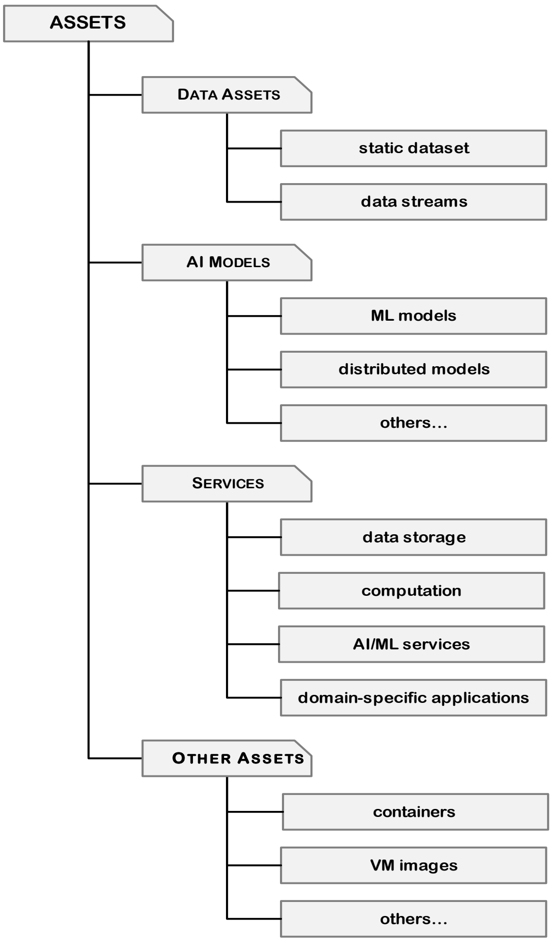


Figure 9: Asset classification

4.1.8 Marketplace Information Model

Ontology including the semantic concepts and their relationships in a Marketplace. It supports the description of Participants' Self-descriptions, Offerings, Self-listings, or Contracts, among others.

4.1.9 Self-description

An extensible document (e.g. JSON-LD) that describes a Participant following the Marketplace Information Model in a machine interpretable format. This document contains identity information and, in case of providers, a reference to its Self-listing.

4.1.10 Offering

A document (e.g. JSON-LD) that represents a set of Assets following the Marketplace Information Model. It is stored at the Provider domain. It includes the description of Assets and a reference to a Contract.

4.1.11 Self-listing

A document (e.g. JSON-LD) that contains or references the set of Offerings from a Provider following the Marketplace Information Model. It is stored at the Provider domain. Self-listings are not searchable by definition.

4.1.12 Contract

Machine interpretable document that regulates an Offering procurement (part of the Offering lifecycle) and must be agreed by the two involved Participants. It should contain a list of access and usage policies (e.g. Open Digital Rights Language (ODRL)) that rule how the Assets represented within that Offering can be consumed.

4.1.13 Agreement

A signed Contract.

4.1.14 Issuer

Trust anchors within a Marketplace to issue Verifiable Credentials.

4.1.15 Registry

Distributed ledger to provide trustworthy, non-repudiable and immutable information about Participants and Offerings. It is the central building block provided by Baseline Infrastructure Facilitators.

4.1.16 Catalogue

Any searchable (i.e. indexed) version of the offering-related information referenced in the Registry, whose purpose is to facilitate Offerings' discoverability in a Marketplace. Baseline Infrastructure Facilitators might provide an instantiation of a Catalogue.

4.1.17 Toolbox

Set of software tools required to interconnect Participants within the Marketplace. It is composed by a Connector, a Data Processing Pipeline, an AI pipeline, a set of GUIs and other added-value tools for helping Participants throughout the Assets' exchange lifecycle.

4.1.18 Connector

A software tool to enable secured peer-to-peer information exchange between Participants (e.g. Offerings and their corresponding Assets).

4.1.19 Data Processing Pipeline

Set of software tools to process and improve the quality of Data Assets, representing them using the Data Asset Information Model that is being defined in SEDIMARK. Also referred as DPP.

4.1.20 Data Asset Information Model

Representation of semantic concepts and their relationships to describe the content of Data Assets handled by the Data Processing Pipeline (e.g. NGSI-LD Smart Data Models).

4.1.21 AI Pipeline

Set of components to train machine learning models and provide inference and analytics using Data Assets and/or AI Models, representing them using the Data Asset Information Model and the AI Model Information Model that are being defined in SEDIMARK. The AI pipeline comprises components such as model training and evaluation, model formatting, model optimisation, model inference and offering ML services. The AI pipeline also utilises components from the Data Processing Pipeline for data pre-processing, cleaning and feature extraction. Also referred as AIP.

4.1.22 AI Model Information Model

Representation of semantic concepts and their relationships to describe the content of AI Models handled by the AI Pipeline.

4.1.23 Artefacts

Encompassing term referring to all entities that can be created, stored, exchanged, and manipulated to facilitate various processes and interactions within a Marketplace. These elements includes Assets, Participants' Self-descriptions, Offerings, Self-listings, or Contracts, among others. Artefacts can support the following functionalities: Generation, Formatting, Processing, Annotation and Validation. All these functionalities are carried out according to the different Information Models (i.e: Marketplace IM, Data Asset IM, AI Model IM, …)

4.1.24 Graphical User Interface Tools

The graphical user interface tools refer to a set of frontends to facilitate the usage of the marketplace functionality to all of its participants. These frontends cover:

- New participant registration
- Offering registration
- Offering catalogue browsing
- Offering lifecycle management
- Contract negotiation
- Data processing pipelines management
- AI pipelines management

4.1.25 Recommender

A software tool to provide personalised recommendations of Offerings to Consumers.

4.2 Comparison with other international initiatives

As stated before, SEDIMARK terminology has been defined trying to be as aligned as possible with other international initiatives such as Gaia-X and IDSA. In particular, Gaia-X related terms are based on Gaia-X Architecture 22.10 [14] [22]; while IDS related terms are based on its Reference Architecture Model [11], its Information Model [23] and the Data Space Protocol 0.8 [12]. In this regard, Table 1 shall provide an overview of the mapping between SEDIMARK terminology and the analogous terms defined by such initiatives.

Table 1. Terminology comparison overview.

SEDIMARK	GAIA-X	IDSA
Marketplace	Ecosystem	Data Space
Marketplace RI	CATENA-X	Mobility data space
Participant	Participant	Participant
Baseline Infrastructure Facilitator	Federator (although not exactly the same)	Data Space Authority
Provider	Provider	Provider
Consumer	Consumer	Consumer
Asset	Resource (virtual or physical)	Asset / Artifact
Marketplace Information Model	Conceptual Model	Data Space Information Model
Data Asset Information Model	-	-
AI Model Information Model	-	-
Self-description	Self-description	Self-description
Offering	Service Offering	Asset Entry
Self-listing	Catalogue (although it does not exist locally) or Service Offerings list	Catalog
Contract	Contract Template (not exactly the same)	Contract / Offer
Agreement	Contract	Agreement
Issuer	Trust Anchor	Credential Issuer
Registry	Gaia-X Registry	DataSpace Registry
Catalogue	Federated Catalogue	Metadata Broker
Toolbox	-	Participant Agent
Connector	- (not really part of the spec)	Connector
Artefact	-	- (Artifact is a totally different concept)

4.3 Concept relationships

Figure 10 shows the relationships among all the concepts defined in the previous subsections. It should be noted that links between concepts should be read from top to bottom.

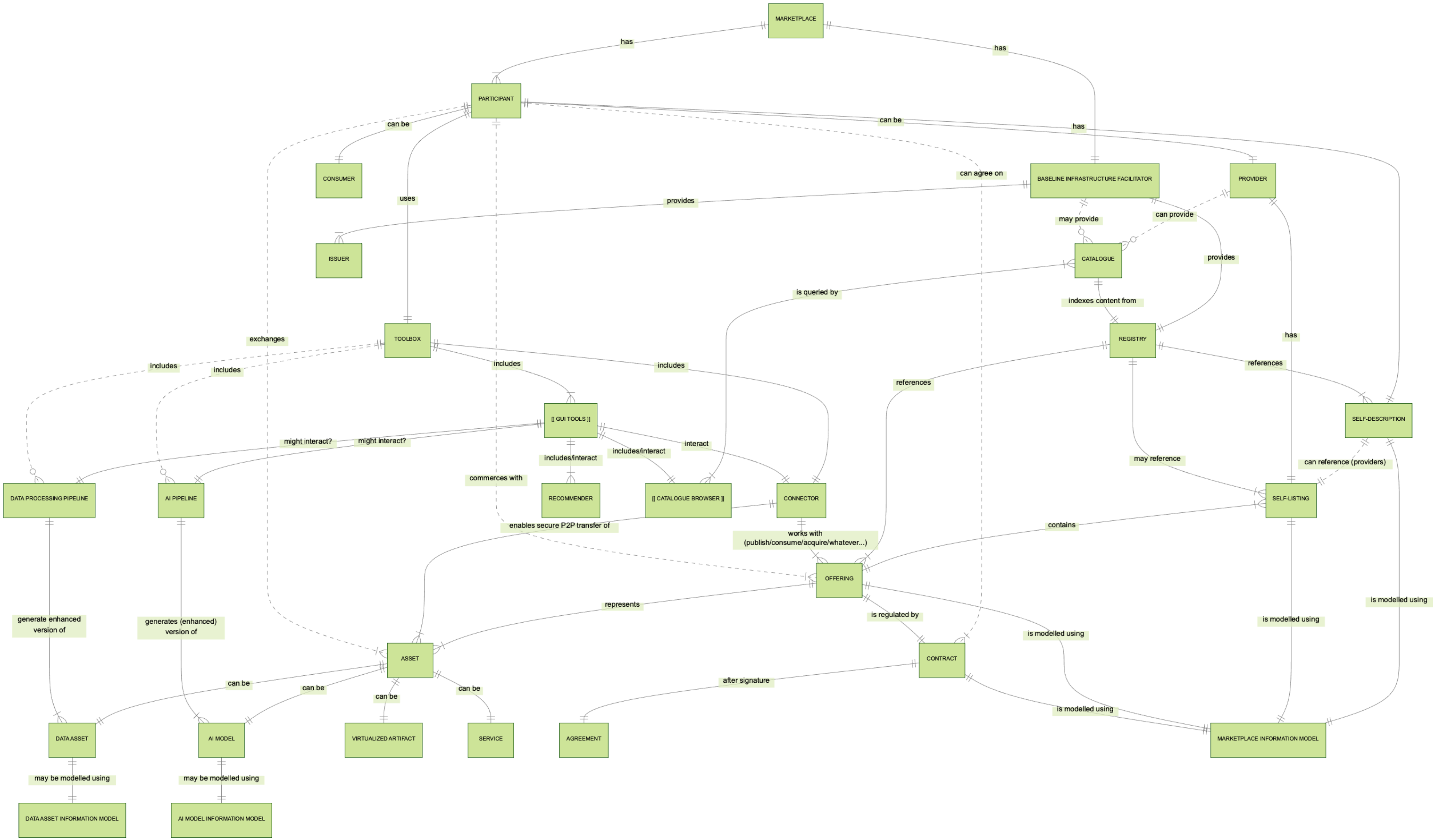


Figure 10: Glossary concepts relationships

5 Requirements Analysis

5.1 Project objectives

The primary goal of the SEDIMARK project is to design and prototype a secure decentralised and intelligent data and services marketplace that bridges remote data platforms and allows the efficient and privacy-preserving sharing of vast amounts of heterogeneous, high quality, certified data and services supporting the common EU data spaces. To translate this ambition into actionable requirements, this main goal has been split into five key objectives:

1. Design a decentralised infrastructure for a data and service marketplace allowing the easy and secure gathering, processing, discovery, sharing, integrity verification and exploitation of heterogeneous data and service sources considering the EU Data spaces design principles.
2. Develop a common ontology and a complete AI-based data management toolset, for curating heterogeneous data, improving quality and interoperability, enabling their efficient reuse in EU data spaces.
3. Develop techniques for secure and privacy preserving storage, discovery, quality and confidence ranking of data in distributed platforms, for advanced confidentiality, while at the same time allow local access to data and global exploitation of inference/information extracted from data.
4. Develop distributed Green AI techniques as a Service that allow the efficient and secure processing of large amounts of data from remote platforms, ensuring that global inference and knowledge is extracted without the need to transmit huge amounts of (possibly confidential) information to global servers.
5. Develop techniques to allow secure and fair access to data using strong access control, anonymisation, data minimisation and exploiting DLT.

Table 2 below decomposes each of these objectives into a list of concise goals the SEDIMARK platform aims at covering.

Table 2. Project objectives

ID	Description
1.1	Design a decentralised infrastructure for a data and service marketplace.
1.2	Allowing the easy and secure gathering, processing of heterogeneous data and service sources
1.3	Allowing the easy and secure discovery, sharing and exploitation of heterogeneous data and service sources.
1.4	Allowing integrity verification and exploitation of heterogeneous data and service sources
2.1	Develop a common ontology, improving interoperability, enabling their efficient reuse in EU data spaces.

ID	Description
2.2	Develop a complete AI-based data management toolset, for curating heterogeneous data, improving quality.
3.1	Develop techniques for secure and privacy preserving storage
3.2	Develop techniques for discovery in distributed platforms
3.3	Develop techniques for quality and confidence ranking of data in distributed platforms, for advanced confidentiality
3.4	Develop techniques (that) allow local access to data and global exploitation of inference/information extracted from data.
4.1	Develop distributed Green AI techniques as a Service that allow the efficient and secure processing of large amounts of data from remote platforms.
4.2	Develop distributed ensuring that global inference and knowledge is extracted without the need to transmit huge amounts of (possibly confidential) information to global servers.
5.1	Decentralized Identities with DLT
5.2	Trust Management

The following sections in this chapter review all functional requirements of the SEDIMARK platform, as described in deliverable D2.1, and maps them to these goals. The requirement analysis is being presented using tables with the following fields:

- **Identifier**: this is the identifier of the requirement in the form Req-[Cat]-[Num], where [Cat] is the category of the requirement (i.e. SEC for security, DP for Data Processing, etc.) and [Num] is an increasing index number for the requirements of the same category
- **Short Name**: this is an easy to remember short name for the requirement (see D2.1 for details).
- **Priority**: the priority level for the requirement, which can be H (High), M (Medium) or L (Low).
- **Requirement level**: the level of the requirement, which can be REQ (Required), REC (Recommended) or O (Optional).
- **Objectives**: which project objectives are related to this requirement.
- **Functional components**: these are the functionalities (drafted as components) that the system architecture should support to meet the requirement. This is the main result of the requirement analysis and these will be included in the functional view of the architecture which will be presented in Section 6.

5.2 Security, privacy and trust

Among the targets of the SEDIMARK project lies the protection of the assets exchanged. Such level of security is ensured by the establishment of trust relationships among the parties involved as well as securing the various interactions. Moreover, maintaining the user privacy in the marketplace is of paramount importance. The following security requirements match the objectives of the SEDIMARK project to enable the protection of the marketplace and its internal interacting parties.

Table 3. Security, privacy and trust requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-SEC-01	Authentication of users	H	REQ	5.1	Identity management
Req-SEC-02	Authorization policies of assets	H	REQ	3.1, 5.2	Trust management
Req-SEC-03	Origin of assets	H	REQ	3.1, 5.2	Trust management IOTA Client
Req-SEC-04	Trust Metadata on Distributed Ledger	H	REQ	1.1, 1.2, 1.3, 3.1, 5.1, 5.2	Trust management Data integrity Registry IOTA Client
Req-SEC-05	Decentralized Provisioning	H	REQ	1.1, 5.1, 5.2	Contracting Smart Contracts Service Request Service Provisioning IOTA Client Transactions
Req-SEC-06	Secure channel of the assets	H	REQ	1.2, 1.3, 1.4, 5.2	Data encryption Data integrity Offering Sharing

5.3 Data Processing

Data processing is one of the main pillars of SEDIMARK, which aims to provide the necessary tools to data providers and consumers so that they can curate their data according to their preferences and improve the quality, so that they can both use it internally but also share high quality data through the SEDIMARK marketplace.

Table 4. Data Processing requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-DP-01	Data cleaning tools	H	REQ	1.2, 2.2, 3.3	• Data curation • Data profiling • Data processing dashboard
Req-DP-02	Flexibility to handling both static and streaming data	M	REC	1.2, 2.2, 3.3	• Data curation • Data orchestrator • Data profiling • Data processing dashboard
Req-DP-03	Flexibility to handling both static and streaming data	M	REC	1.2, 2.2, 3.3, 4.1	• Data curation • Data processing dashboard • Data orchestrator • Data profiling • Data adapter
Req-DP-04	Data quality indicators	M	REC	2.1, 3.2, 3.3	• Data processing dashboard • Data quality evaluation • Data integrity • Annotation • Data profiling • Data augmentation
Req-DP-05	Adaptability of data cleaning mechanisms	M	REC	1.1, 1.2, 4.1	• Data quality evaluation • Energy efficiency • Data augmentation • Data profiling • Data orchestrator
Req-DP-06	Ground truth for data quality metrics	L	OPT	1.2, 2.2, 3.3	• Data profiling • Data adapter • Data quality evaluation • Data curation
Req-DP-07	Data cleaning modules extendable definitions	M	REC	1.2, 2.2, 3.3	• Data curation • Data processing dashboard • Data adapter • Data quality evaluation • AI orchestrator

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-DP-08	Automate execution of tasks	L	OPT	1.2, 1.3, 2.2, 3.3, 4.1	• Data orchestrator • Data processing dashboard
Req-DP-09	Dataset augmentation	M	REC	1.2, 1.4, 2.1, 2.2, 4.1, 4.2	• Data orchestrator, Data processing dashboard
Req-DP-10	Anonymisation of private information	H	REC	1.2, 3.1, 3.3	• Data profiling • Data anonymisation • Data encryption • Data processing dashboard • Data orchestrator

5.4 Artificial Intelligence

This section presents the analysis of the Artificial Intelligence related requirements, which were presented in Section 6.4 of SEDIMARK deliverable D2.1. As discussed also there, Intelligence is one key pillar of SEDIMARK and the project platform should support the efficient training of machine learning models both at local level and distributedly. Also, SEDIMARK should support an effective recommendation system that will provide recommendations to the users when they discover offerings and assets that are shared through SEDIMARK.

Table 5. Artificial Intelligence requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-ML-01	Model input data cleaning and formatting	H	REQ	1.4, 3.4, 4.1, 4.2	• Data curation, • Data profiling, • Annotation, • Data adapter, • Feature engineering, • Semantic enrichment • Data augmentation, • Local model training, • Distributed model training
Req-ML-02	Decentralised ML	M	REC	1.1, 1.3, 1.4, 4.1, 4.2	• Distributed model training, • Formatting, • Model inference, • AI as a service

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-ML-03	Trusted participation in decentralised training	H	REC	1.1, 3.1, 4.1, 4.2, 5.1, 5.2	• Distributed model training, • Trust management, • Identity management, • Data encryption, • AI as a service
Req-ML-04	Models agnostic to platforms	M	REC	4.1, 4.2,	• Model optimisation, • Local model training, • Distributed model training, • Frugal AI, • Formatting, • AI orchestrator
Req-ML-05	Models persistence mechanisms	H	REC	1.1, 1.3, 4.1, 4.2	• Distributed model training, • Local model training, • Data storage, • Formatting, • Data analytics, • AI as a service
Req-ML-06	Event generation from pattern extraction	H	REC	1.3, 1.4, 4.1, 4.2	• Data analytics, • Semantic enrichment, • Model optimisation, • Model inference
Req-ML-07	Synchronous and asynchronous training of models	L	OPT	1.1, 4.1, 4.2	• Distributed model training, • AI orchestrator, • Model optimisation, • Formatting, • AI as a service
Req-RS-01	User profiling	H	REQ	1.2, 1.3, 3.1, .3.4	• User profiling, • Logging, • Recommendations,
Req-RS-02	Rich item information	H	REQ	1.2, 1.3, 2.1, 3.4	• Recommendations, • Offering discovery, • Offering statistics, • Monitoring, • Ratings

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-RS-03	Decentralised Recommender system	H	REQ	1.1, 1.2, 3.1, 4.1, 4.2, 5.1, 5.2	- Distributed model training, - Recommendations, - Data encryption, - Data anonymisation, - Trust management
Req-RS-04	Cold start problem	H	REC	1.1, 1.2, 1.3, 4.1, 4.2, 5.2	- Recommendations, - Distributed model training, - AI model optimisation

5.5 Energy efficiency

The SEDIMARK platform puts a strong emphasis on energy efficiency. Modules will be designed to be as lightweight as possible while still aiming for a top performance. This will be applied to data processing modules and machine learning models, both for the training and inference parts.

Table 6. Energy Efficiency requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-EE-01	Lightweight and energy efficient DP modules	H	REQ	1.2, 2.2, 4.1, 4.2	- Data curation - Data augmentation - Data profiling - Data orchestrator - Data processing dashboard - Model optimisation - AI orchestrator
Req-EE-02	Lightweight and energy efficient AI/ML models	M	REC	4.1, 4.2	- Local model training - Distributed model training - Frugal AI - Model inference - Model optimisation
Req-EE-03	Energy efficient decentralized training of ML model	H	REQ	4.1, 4.2	- Model optimisation - Model inference - AI orchestrator
Req-EE-04	Usage of compiler optimizations for ML model	L	OPT	4.1, 4.2	- Energy efficiency - Frugal AI - Model optimisation - AI orchestrator

5.6 Interoperability

SEDIMARK aims at integrating data and related actors and assets of very heterogeneous origins. Hence, ensuring interoperability among such heterogeneous entities is of vital importance. An essential element towards this is introducing a set of common information models that represent all SEDIMARK entities (participants, data, services, AI models, etc.) in a unified way, thus allowing their identification, processing and interaction inside the SEDIMARK platform. In the following, we detail the information model requirements and associate each one of them to SEDIMARK objectives while precising the related functional components.

Table 7. Interoperability requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-INT-01	Information model for interoperability	High	REQ	1.2, 2.1, 3.4	• Semantic enrichment • Annotation • Data processing dashboard • Data orchestrator • Validation
Req-INT-02	Information model for data and their metadata	High	REQ	2.1, 3.3, 3.4	• Data quality evaluation • Formatting • Semantic enrichment • Annotation • Data processing dashboard • Data orchestrator • Validation
Req-INT-03	Metadata fields	High	REQ	1.2, 2.1, 3.1, 3.4	• Data quality evaluation • Semantic enrichment • Annotation • Data anonymisation • Validation
Req-INT-04	Data compliance with the information model	H	REQ	1.2, 2.1, 2.2, 3.4	• Data quality evaluation • Formatting • Semantic enrichment • Annotation • Data anonymisation • Validation

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-INT-05	Enforcing data compliance with the information model	M	REC	1.2, 1.3, 1.4	• Annotation • Formatting • Validation • Data adapter • Data processing dashboard • Data orchestrator • Data quality evaluation
Req-INT-06	Information model for AI models	H	REC	1.2, 1.3, 1.4, 2.2, 4.1, 4.2	• Distributed model training • Formatting • Annotation • Model optimization • Model inference • Validation

5.7 Data Storage

The Data storage component plays a fundamental role to support the function of the other components in the SEDIMARK architecture. The requirements identified relate to the location and hosting of data and metadata belonging to a data provider. It also highlights the need to support the temporal storage of intermediate assets within processing pipelines, and the employment of the appropriate type of persistence used for each asset and ensuring final assets are stored.

Table 8 Interoperability requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-STR-01	Default dataset storage domain	H	REQ	1.1, 3.1, 4.2	• Data storage • Distributed storage
Req-STR-02	Storage of offering descriptions on distributed catalogue	H	REQ	1.1, 1.2, 1.3, 1.4, 3.1	• Data storage • Offering description • Local catalogue • Distributed catalogue • Offering registration • Offering discovery
Req-STR-03	Temporary storage of intermediate artefacts with pipeline	H	REQ	1.2, 2.2, 3.4, 4.2	• Data storage • Feature engineering • Data orchestrator

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-STR-04	Storage of post-processed data in consumable manner	H	REQ	3.1, 4.2	• Data storage • Model inference, • Formatting, • Validation
Req-STR-05	Management of data for distributed storage	L	OPT	1.2, 2.2, 4.1	• Data storage • Distributed storage • Data processing dashboard • Data orchestrator • Offering sharing
Req-STR-06	Storage Service for constrained data providers	M	OPT	1.1, 3.1, 4.2	• Distributed storage • Service provisioning • Offering sharing
Req-STR-07	Storage for knowledge domain services	M	REC	1.3, 2.1, 2.2, 3.2, 3.4	• Data storage • Semantic enrichment • Distributed catalogue
Req-STR-08	Storage for offerings other than datasets	M	REC	1.2, 1.3, 2.2, 3.4, 4.2	• Data storage • Feature engineering • Local model training • Model inference • Formatting • Model optimisation • Validation

5.8 Publication and discovery

In order to enable a proper exchange of information among participants in the SEDIMARK marketplace, several requirements for publication and discovery of the different assets provided in the marketplace need to be fulfilled. Table 9 details these requirements and associate them with the SEDIMARK objectives and the different functional components.

Table 9. Publication and discovery requirements.

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-P&D-01	Assets described as part of offerings	H	REQ	1.3, 2.1	• Offering description • Offering discovery

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-P&D-02	Offerings' registry	H	REQ	1.1, 1.3, 3.1, 3.2	- Offering registration - Offering sharing - Offering discovery - Registry - Distributed storage - Local catalogue - Data storage - Distributed catalogue
Req-P&D-03	Generic offering metadata	H	REQ	1.3, 2.1	- Offering description
Req-P&D-04	Open Data portal discovery	H	REQ	1.1, 1.2, 1.3	- Open data enabler
Req-P&D-05	Offerings' catalogue for queries	H	REC	1.1, 1.3, 3.1, 3.2	- Offering discovery - Distributed catalogue

5.9 User requirements (Marketplace User Interfaces)

The SEDIMARK platform will offer a handful of graphical interfaces to simplify its interactions with various users, ranging from simple visitors wishing to browse the offerings catalogue, to participants willing to buy/sell services. Through these graphical interfaces, users should be able to login, discover and manage their offerings, as well as accessing the SEDIMARK toolbox for data processing. Consequently, these requirements span over a large set of the objectives defined in Table 10.

Table 10. Marketplace & User Interfaces requirements.

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-UI-01	Logging in UI	H	REQ	1.1, 1.3, 1.4	- Identity management, - Frontend
Req-UI-02	Offerings discoverability	H	REQ	1.3	- Offering discovery, - Frontend
Req-UI-03	Users' identity management	H	REQ	1.3, 5.1, 5.2	- Identity management, - Frontend
Req-UI-04	Offerings management	H	REQ	1.2, 1.3, 5.1, 5.2	- Offering registration, - Trust management, - Frontend - Payment/Billing

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-UI-05	Offering description page	H	REQ	1.3	• Offering discovery, • Data visualisation, • Frontend
Req-UI-06	SEDIMARK toolbox access in UI	M	REQ	1.3	• Offering discovery, • Data visualisation, • Frontend
Req-UI-07	Rating of offerings	L	REQ	3.3	• Ratings, • Data visualisation, • Frontend
Req-UI-08	Offerings statistics	L	OPT	3.4	• Open data enabler, • Data visualisation, • Frontend

5.10 Smart contract and tokenisation

The underlying decentralized infrastructure of the SEDIMARK project enables the adoption of a special form of transactions built on top of the Distributed Ledger. The transactions of assets and services in the SEDIMARK Marketplace are governed through a special form of agreements, namely the Smart Contracts. The related requirements to fulfil the trading functionalities of the marketplace in the SEDIMARK project are reported in Table 11.

Table 11. Smart Contract and tokenisation requirement analysis

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-SCT-01	Smart Contracts support	M	REQ	1.1	• Smart contracts • Transactions
Req-SCT-02	Tokenisation of Assets	M	REQ	1.1	• Smart contracts • Tokenisation
Req-SCT-03	User Digital Wallet	M	REQ	1.1	• Tokenisation • Payment

5.11 Non-functional requirements

This section presents the analysis of the non-functional requirements, which were presented in Section 6.1 of SEDIMARK deliverable D2.1. Although these are non-functional requirements which define general system behaviour, some of them can be mapped to functional components. The following table shows this mapping for most of the non-functional requirements. The rest of the requirements present general expected behaviour either for the platform as a whole or for all of the modules to be developed.

Table 12. non-functional requirements.

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-NF-01	Decentralisation	H	REQ	1.1, 4.1, 4.2, 5.1	• All DLT layer modules, • Distributed model training • Distributed storage • Distributed catalogue
Req-NF-02	Security, Privacy, Trust	H	REQ	1.2, 1.3, 1.4, 3.1, 3.3, 4.1, 4.2, 5.1, 5.2	• All security/trust layer modules
Req-NF-03	Interoperability	H	REQ	1.2, 1.3, 1.4, 2.1, 2.2	• Annotation • Data adapter • Semantic enrichment • Validation • Formatting
Req-NF-04	Data availability and quality	H	REQ	1.2, 1.3, 1.4, 3.2, 3.3	• All data processing modules
Req-NF-05	Intelligence	H	REQ	3.4, 4.1, 4.2	• All AI layer modules
Req-NF-06	Energy efficiency	H	REQ	4.1, 4.2	• Energy efficiency (module in data processing) • Frugal AI
Req-NF-07	Resilience and Reliability	H	REQ	1.1, 1.4, 2.2	• N/A (mapped to all components and the platform as a whole)
Req-NF-08	Scalability	H	REQ	1.1, 3.4, 4.1, 4.2	• N/A (mapped to all components and the platform as a whole)
Req-NF-09	Openness, Extensibility	H	REQ	1.1, 5.1, 5.2	• N/A (mapped to the platform as a whole)
Req-NF-10	Usability	H	REQ	1.2, 1.3	• All marketplace service modules
Req-NF-11	Maintainability	H	REQ	1.1	• N/A (mapped to all components and the platform as a whole)
Req-NF-12	Adaptivity to data types and fast processing	M	REC	1.2, 1.3, 1.3, 2.1, 2.2	• All data processing modules

Identifier	Short Name	Priority	Req. Level	Objectives	Functional Components
Req-NF-13	Reusability	H	REQ	1.2, 1.3, 3.4	• All asset sharing and discovery modules.
Req-NF-14	Flexibility	H	REQ	1.2, 1.3, 2.1, 2.2, 3.1	• N/A (mapped to all components and the platform as a whole)

6 SEDIMARK Architecture

6.1 Overview – High Level view

6.1.1 Overview

This section presents the first draft of the SEDIMARK functional architecture, designed to address the key objectives of the project for a decentralised, intelligent, trustworthy and interoperable data and services marketplace. Compared to most existing data marketplaces, which are centralised, SEDIMARK proposes to move towards a fully decentralised solution (as shown in Figure 11), without any single point of data collection or single point of failures. SEDIMARK enables the data providers to keep their data locally, at their own premises and only share them with the consumers they want to. The fully decentralised architecture of SEDIMARK is built on the EU strategy for Data [24], aiming to allow data to flow easily within the EU, while users get FAIR access to data and data are protected, being shared only in a secure and trustworthy way.

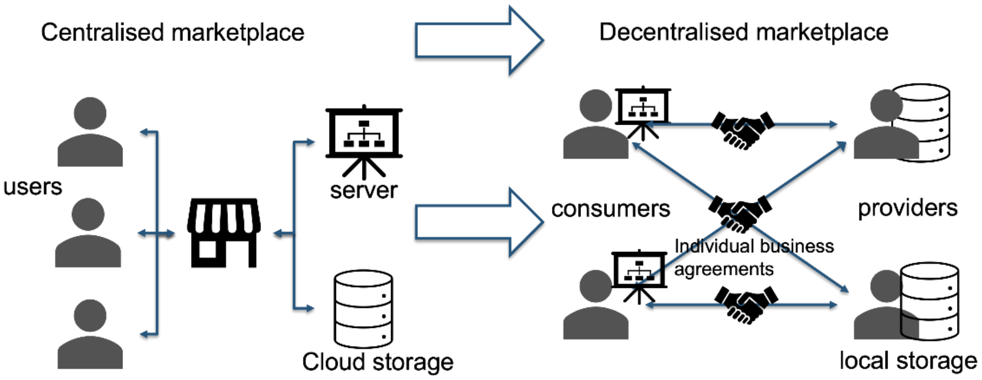


Figure 11: Centralised vs decentralised marketplaces

The conceptual decentralised architecture of SEDIMARK depicted in Figure 12 is designed so that the communication between the various participating nodes (providers, consumers, etc.) is done on a peer-to-peer basis, without intermediate nodes or proxies, enabling consumers to get direct access to the assets of providers. The architecture is leveraging on DLT as a means for providing trust in the overall marketplace, verifying the credentials of the communicating peers and the transactions between them.

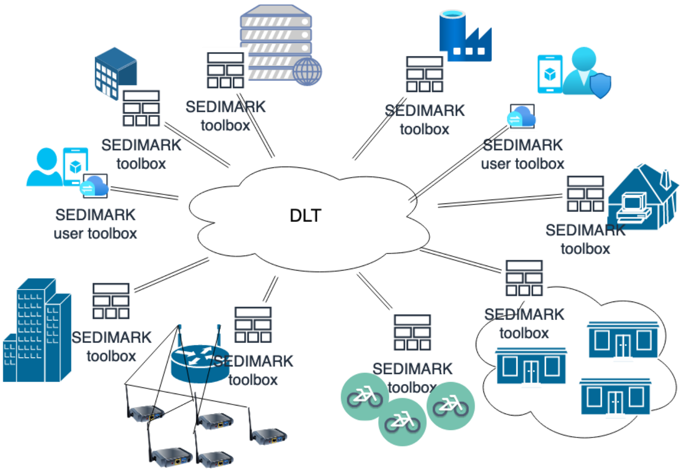


Figure 12: SEDIMARK conceptual decentralised network

The SEDIMARK architecture aims to support the SEDIMARK objectives by providing a set of tools that will be utilised both by providers and consumers to seamlessly exchange assets in a secure and trusted way:

- **Providers**: providers will get all the necessary tools to manage their assets and convert them to offerings for being shared in the marketplace. In this respect, data providers will be able to get insights about the quality of their datasets, process them and improve their quality, train and run AI models and analytics on them and manage which type of consumers will be allowed to get access to them. Service providers will be able to manage their services and set the access policies.
- **Consumers**: consumers will be able to discover assets that are of interest or relevant to them, get access to these assets, request datasets in various formats, access services or request model training and analytics results, while also getting automated recommendations based on their preference and interaction history.

6.1.2 High level view of the SEDIMARK architecture

The fully decentralised nature of the SEDIMARK project can be seen in the high level view of the SEDIMARK architecture, which is depicted in Figure 13. As it is clear in the figure, the main actors of the SEDIMARK architecture are the providers, the consumers and the supporting infrastructure for the DLT and for providing trust. Since SEDIMARK is a fully decentralised architecture there is not a central node that stores information, data or manages the overall SEDIMARK network. The communication between the consumers and the providers is done directly, on a peer-to-peer basis, without intermediary cloud infrastructure.

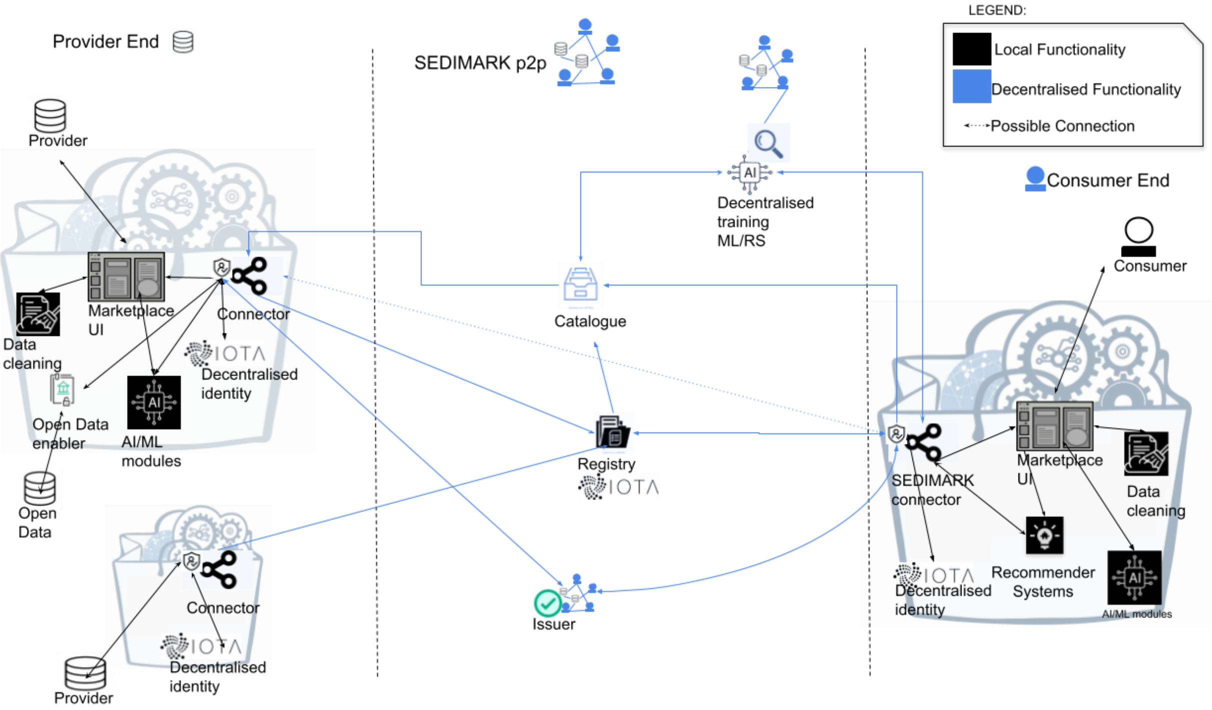


Figure 13: High level view of the SEDIMARK architecture

Any participant in SEDIMARK will download and install on their premises/servers a version of the SEDIMARK toolbox that will include the necessary components that correspond to the participant's role in the marketplace (provider, consumer or both). It is assumed that each role will include different components of the toolbox for providing the necessary functionalities for that specific role, while other optional components will also be available for download to provide additional extra functionalities. For example, the provider toolbox will include components for data processing and cleaning, which might not be of interest for a consumer. On the contrary, the consumer toolbox will include recommendations and asset discovery functionalities, which might not be of interest for a provider. Thus, to achieve this, the SEDIMARK architecture will have to be modular and extensible so that modules can be added or removed without significant (or any at all) user interaction. More details about the integration of the SEDIMARK architectural components and the creation of the different instantiations per participant role will be given in the deliverables of WP5.

As one can see in the Figure 13, the main two roles in the SEDIMARK architecture is the provider and the consumer. A provider that has some assets to offer through SEDIMARK will use the toolbox to manage these assets, create the offering descriptions and register the assets to the Registry, so that they can become available for purchase by the consumers. More specifically, a Data Provider will be able to use the Data Curation pipeline (part of the Data Processing pipeline) of SEDIMARK in order to process their data assets, clean them, see details about the quality of the data, convert them to the SEDIMARK format and then register them to the DLT registry. The key point to note here is that SEDIMARK aims to provide a simplified version of the data curation pipeline that will require minimum user intervention, so that it can be used even by non-expert providers. Providers also have access to an AI pipeline, which enables them to train and run ML models on top of their data in an energy efficient way. Additionally, it allows them to train ML models distributedly, exploiting other providers that have similar datasets, so that the ML models are more generalisable and not just biased towards the dataset that this provider has. Nevertheless, the pipelines of SEDIMARK will be fully customisable so that expert users can configure them as much as they want in order to extract the full potential of their data assets.

Similar to GAIA-X, the SEDIMARK registry is a distributed, non-repudiable, immutable, and permissionless database based on a decentralized infrastructure that keeps track of any transaction that takes place within the SEDIMARK decentralised platform. Providers register their assets there, storing the description of their offerings, so that they can be discovered. The decentralised infrastructure includes also the Catalogue, which can be considered as the federation the Self-listings (see Section 4), keeping a record of all the available offerings in an indexed and easily semantically searchable database. The Catalogue is constructed by crawling the registry and identifying new offerings being registered, adding then these offerings into the publicly searchable database.

The infrastructure includes also the Issuer, which is a federation of servers issuing the verifiable credentials to the SEDIMARK participants, so that they can participate in the platform.

The Consumer interacts with the Issuer to get their credentials and then using their standalone locally run user interface they are able to login to the marketplace, access the Catalogue to discover available offerings, receive recommendations based on their history and profile and request access to assets. Additionally, the consumer is also able to access services offer by providers for running analytics or also for training machine learning models and receive the results of model inference on specific data, instead of accessing datasets.

One important component of the SEDIMARK architecture is the connectivity of the platform with external open data repositories through an open data enabler. This allows SEDIMARK consumers to get access to not only SEDIMARK providers but also external providers through the SEDIMARK system. Consumers will be able to identify through their interface which datasets are provided by SEDIMARK providers and which are external open datasets.

6.1.3 Functional view of the SEDIMARK architecture

To address the goals of SEDIMARK and inspired by the BDVA Architecture Reference model, the functional architecture is split into six (6) architectural layers, as depicted in Figure 14 and detailed below:

- **Data layer**: this is one of the core layers of the SEDIMARK functional architecture and handles the processing of datasets, assessing their quality, cleaning them, annotating and enriching them, ensuring they are interoperable and validating their format, so that they can be shared within marketplace with improved quality.
- **Intelligence layer**: this layer manages the machine learning and artificial intelligence aspects of the SEDIMARK functional architecture, allowing the training of ML models both locally and distributedly, the execution of analytics and the exchange of interoperable ML models between providers and consumers.
- **Interaction layer**: this layers handles the connectivity of the SEDIMARK nodes providing the backbone infrastructure managing the DLT, the distributed registry, the catalogue that stores the offering information and all the transactions between the participants.
- **Services layer**: this is the layer that manages the services provided within SEDIMARK, especially related with the marketplace, i.e. for the registration, discovery and sharing of offerings, payment, provision of functionalities as services (i.e. ML training, analytics, etc.), contracting, recommendations, etc.
- **Trust layer**: this layer manages trust and security throughout the SEDIMARK marketplace, focusing on verifiable credentials, decentralised identities, and data integrity.
- **Distributed storage layer**: this layer manages the distributed storage facilities within the network of participants, ensuring that any offered asset is discoverable, securely stored and available for purchase by consumers.

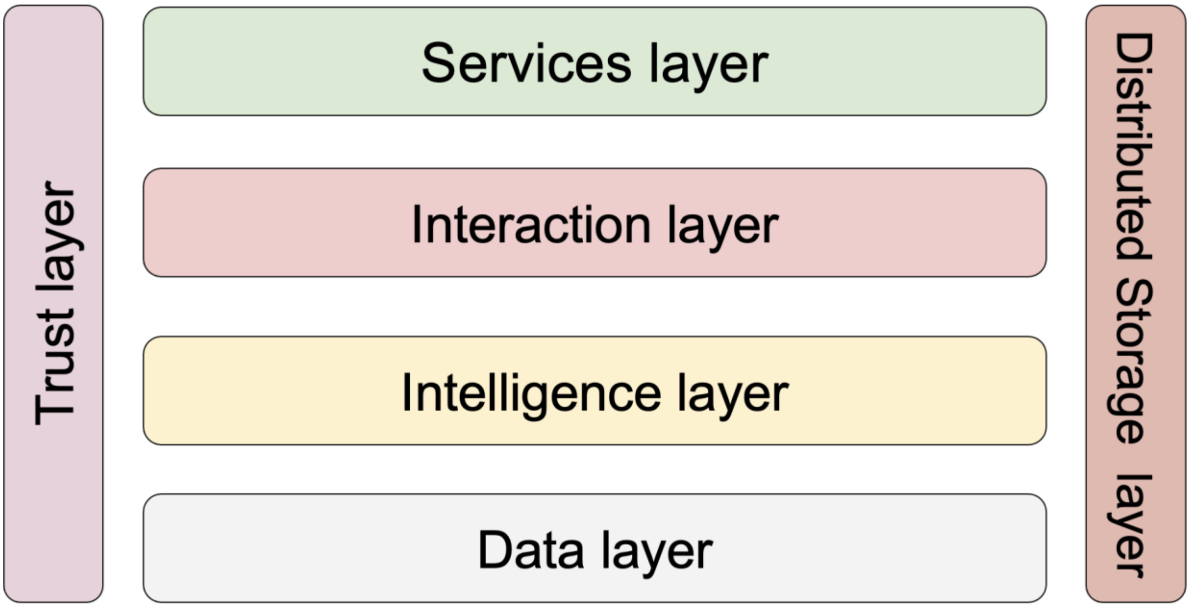


Figure 14: SEDIMARK architecture layers

The more detailed functional view of the SEDIMARK architecture is given in Figure 15 below, which also depicts the initial high-level interaction between the modules of the different layers. As it can be seen, SEDIMARK puts significant emphasis on the Data layer, including many tools for improving the quality of the data, cleaning them, processing them, annotating them and ensuring they conform to the SEDIMARK information model, so that they are highly interoperable. The Data layer components interact with the Distributed Storage layer in order to enable the secure storage of the data assets distributedly. in a secure way and their registration in the catalogue. The data components also interact with the components of the Intelligence layer, to allow the training of ML models and analytics and to convert data points to rich information. The Interaction layer components interact with the components of the data layer to add the registered data assets to the registry and to tokenise the data assets, and with the components of the Intelligence layer to convert AI assets to tokens. The components of the Service layer interact with all other layers in order to provide datasets as services, to support data processing, to manage AI assets, etc. The Trust layer components also interact with all layers to ensure cross-layer trust in all activities.

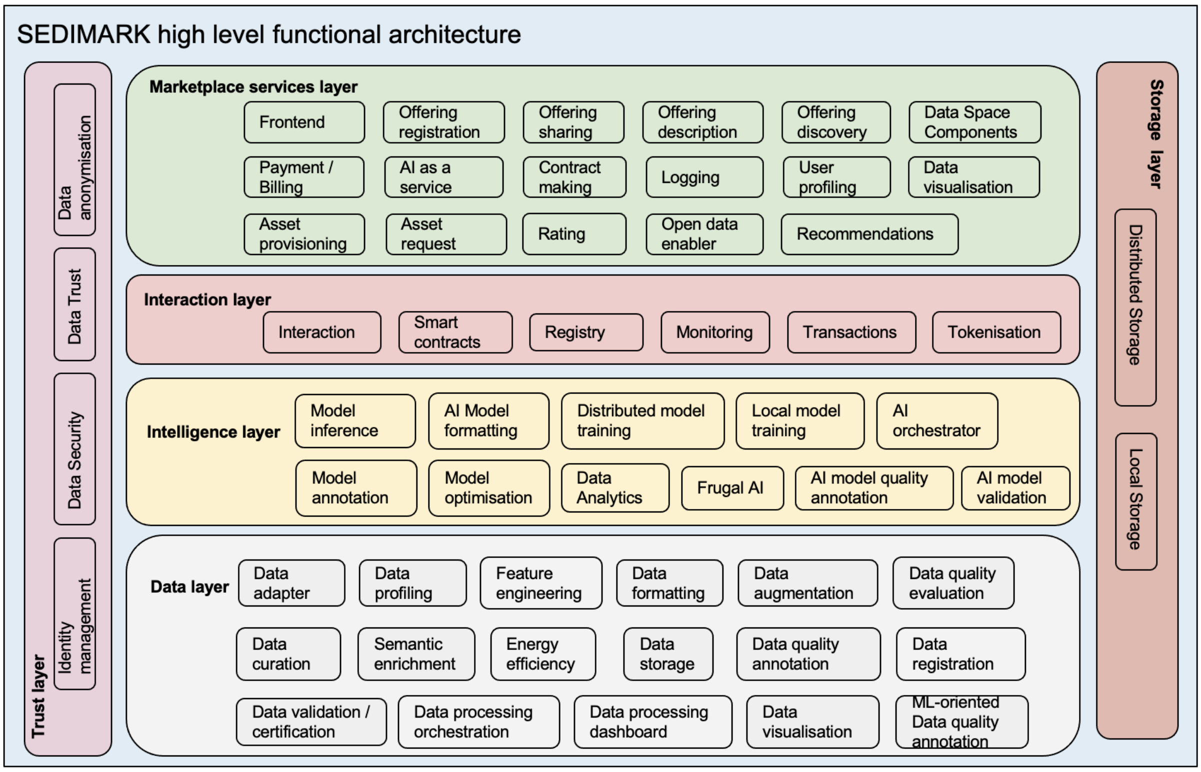


Figure 15: Functional view of the SEDIMARK architecture

More information about the components is given in the following subsections.

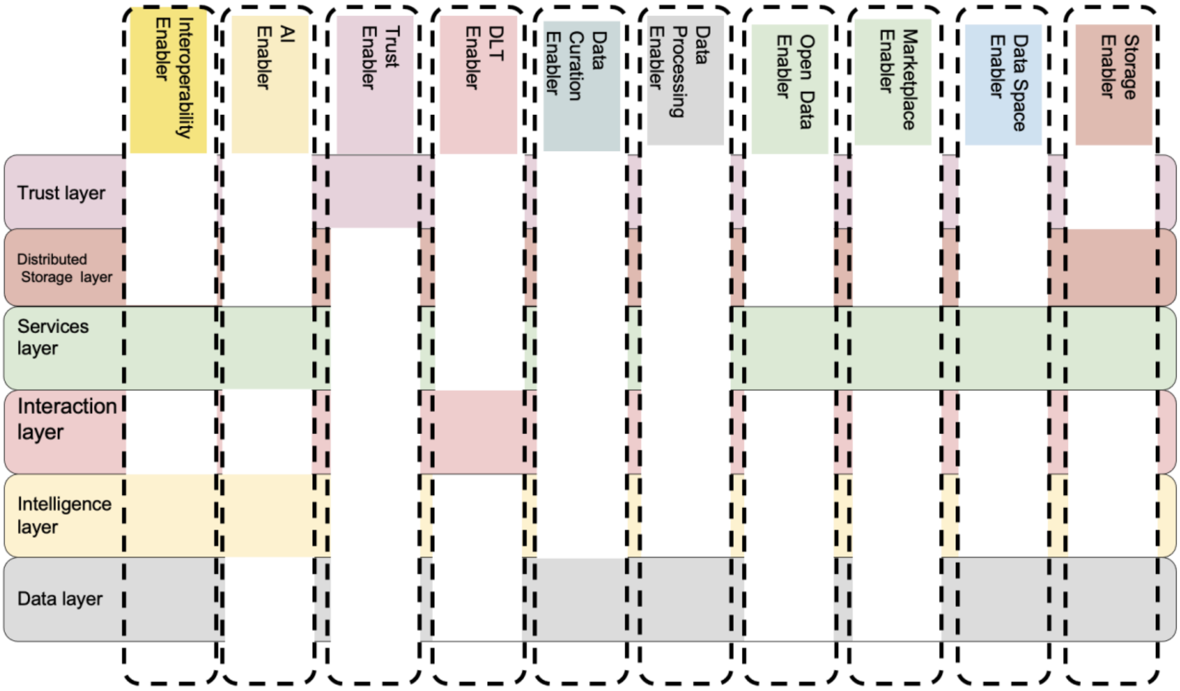


Figure 16: SEDIMARK functional entities mapped to the architectural layers

6.2 SEDIMARK Functional entities

The architectural components of SEDIMARK are grouped into functional entities, which are basically functionality groups of components that jointly perform related functionalities and can be considered as a higher layer abstraction, in order to abstract the architecture and ease its development. In SEDIMARK, the functional components have been grouped into the following ten (10) functional entities as depicted in Figure 16, following vaguely the colour coding of the functional layers, considering at which layer corresponds the majority of the components that are part of the enabler:

- **Data curation enabler**: this entity handles the functionalities for data cleaning, data quality assessment and data profiling.
- **Data processing enabler**: this entity manages the whole data processing pipeline, including the data curation, energy efficiency, data processing orchestration, etc.
- **AI enabler**: this entity manages the whole ML training pipeline, the data analytics, the model inference, etc.
- **Interoperability enabler**: this entity includes the functionalities for transforming the data to the SEDIMARK information model, for data annotation, model interoperability, data validation, etc.
- **Storage enabler**: this entity handles the functionalities for data storage locally and distributedly, as well as for logging and for creating the Catalogues.
- **DLT enabler**: this entity includes the functionalities for handling the DLT connectivity, the transactions, the tokenisation, the smart contracts and the registry.
- **Data Space enabler**: this entity handles the whole life-cycle of offerings, from managing the offering descriptions, to registering the offerings, for offering discovery and sharing.

- **Marketplace enabler**: this entity manages the functionalities for the SEDIMARK marketplace services, including service provisioning, service request, marketplace user interface, logging in, payments, user profiling and recommendations.
- **Trust enabler**: this entity manages the trust-related functionalities throughout the whole SEDIMARK platform.
- **Open data enabler**: this entity is responsible for connecting the SEDIMARK marketplace with external open data repositories, to allow them to be discoverable through SEDIMARK.

Figure 16 shows also the mapping of the functional entities into the architectural layers of SEDIMARK, showing that some of the entities are cross-layer, covering functionalities from multiple layers. For example, SEDIMARK aims to provide cross-layer interoperability for data, AI models and services, so it spans on all these three layers. Similarly, AI enabler assumes that AI models and AI training will also be offered as services.

The interactions between the different functional entities are shown in Figure 17. These interactions and the related interfaces between the entities will be described in more detail in Section 8. Nevertheless, it can be noted here that two vertical entities (Trust and Storage) have connections with most other entities, as also has the DLT which is a central entity in the SEDIMARK architecture.

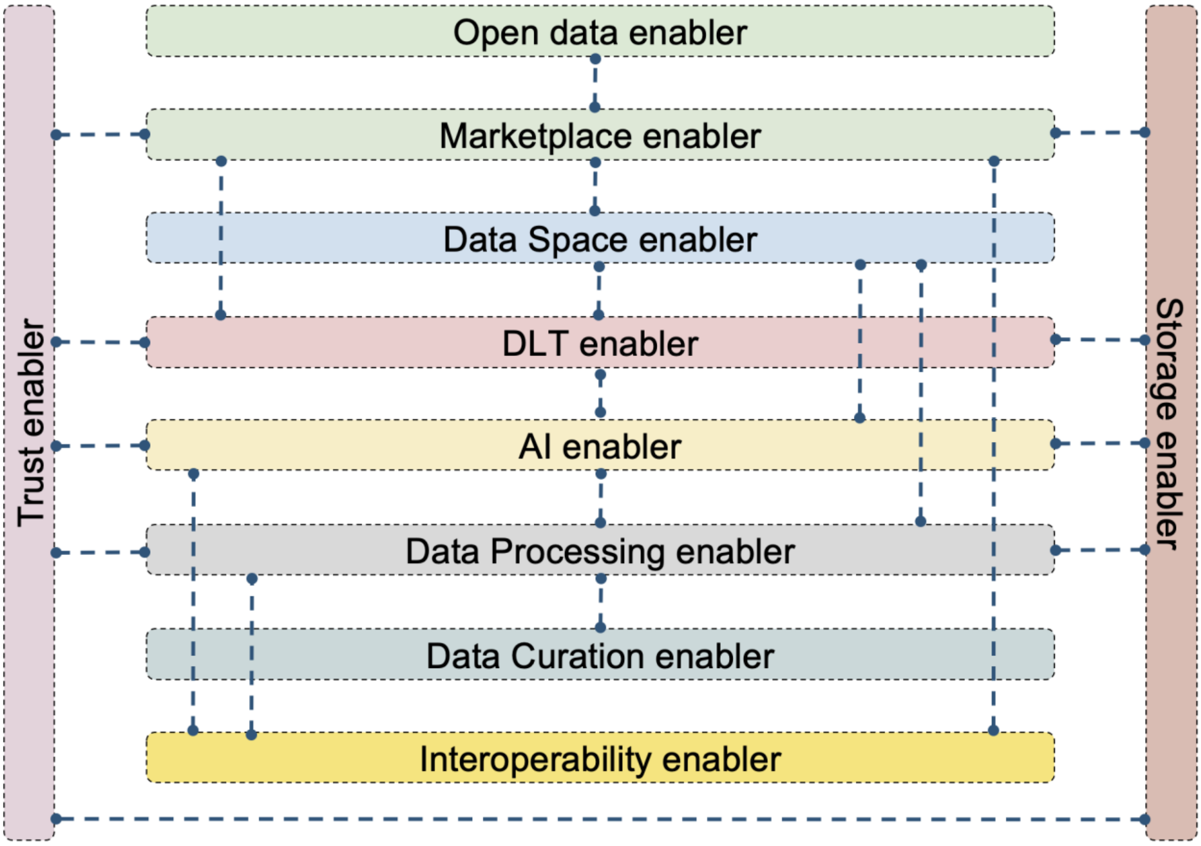


Figure 17: SEDIMARK functional entities interconnectivity

6.3 Data curation enabler

Data quality within SEDIMARK is considered for both consumers and providers. On the one hand, the consumers will have access to high quality data and will know the statistics of quality for the datasets that are being offered through the marketplace, allowing them to discover and purchase datasets based on their assessed quality. On the other hand, the providers will get access to tools that will help them assess and improve the quality of their datasets, increasing value both within their own organisation and within the marketplace. This will also save them loads of man hours dedicated to data processing and allow them to train better quality machine learning models. Additionally, the providers will also be able to request higher prices for their datasets, since they will be cleaned and assessed using the SEDIMARK data curation functionalities.

These functionalities will be developed in SEDIMARK as part of the Data Curation Enabler (DCE). The internal structure of the entity and the interactions of the internal components are depicted in Figure 18. As it can be seen, the DCE includes the following six (6) functional components:

- **Data profiling**: this module is the foundation of data curation and handles the process of analysing and examining the data, creating summaries and insights about the data, including some simple statistics, i.e., number of records, number of fields, data types, min/max values, etc. These statistics and insights are the first step to assess the quality of data and can be of high value for the data providers to get some early indications about issues with their data.
- **Data quality evaluation**: this module provides the assessment about the quality of the dataset in its original form, without being curated or improved within the DCE. In this respect, SEDIMARK will define data quality metrics that go beyond the standard ones used in the literature and will also focus on assessing the data quality with respect to their usefulness to train machine learning models.
- **Data deduplication**: this module handles the detection of duplicate records within the data. Based on the user preference it can identify and label duplicate records or remove them completely. SEDIMARK will build data deduplication modules that can work for both static datasets and data streams, as well as for time series data.
- **Error/outlier detection:** this module handles the detection, labelling and/or removal or errors, noise and outliers in the dataset. Similar to the data deduplication, it requires user input with respect to how the module will handle the outliers, i.e. to keep them and label them as outliers or to remove them completely.
- **Missing value imputation**: this module will react to missing and null/nan values in the dataset, aiming to impute them, filling out the blanks with various methods depending on the type of the data field and dataset.
- **Data augmentation**: this module serves various purposes within SEDIMARK. It is basically a module that creates synthetic data based on the input data. Users can select the amount of data to be generated or they can provide some basic configuration with respect to the augmentation goals and the module will identify automatically how many records and what type of augmentation will be done. Data augmentation is applied within SEDIMARK for (i) increasing the size of small datasets, (ii) balancing datasets that are unbalanced towards some labels, (iii) increase fairness and (iv) remove biases.

Figure 18 also shows the interactions of the different internal functional components of the DCE. The first two steps are for profiling the data and performing the initial quality evaluation. Then, depending on the needs of the dataset and/or the preferences of the user (provider), any of the other curation modules can follow or the dataset can go directly to data augmentation. Data augmentation should only take place in “cleaned” datasets in order to avoid creating more outliers, duplicates or errors during the augmentation process. In the end, to assess the improved quality of the dataset, it can go through the data quality evaluation process again to compute the final quality statistics.

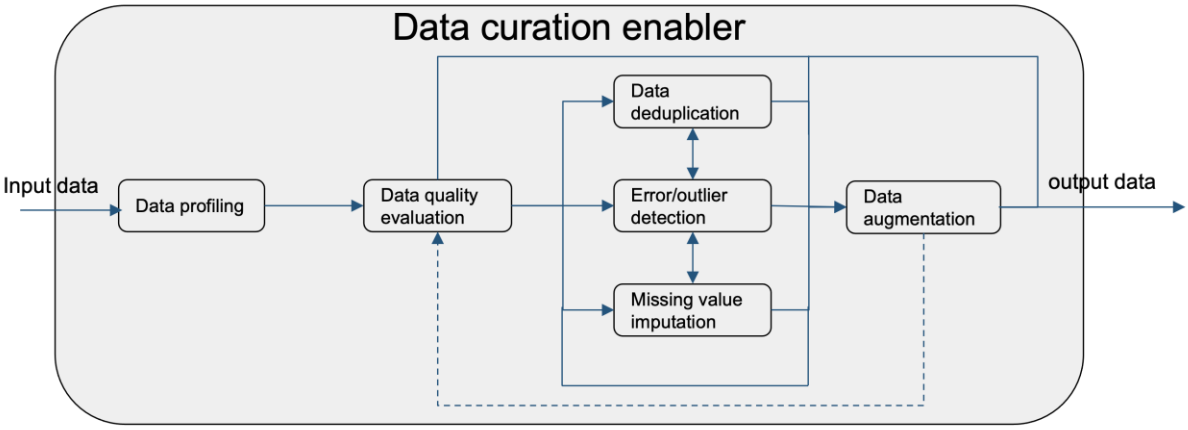


Figure 18: Internal structure of the Data curation enabler.

6.4 Data processing enabler

The data processing enabler functional entity includes data processing orchestration, to register and run each data processing step, data adapter, to convert external data to internal SEDIMARK data format and convert it back to the external data format. It also includes other functionalities to enhance the processed data, such as semantic enrichment and feature engineering, as well as energy efficiency for all the processing steps.

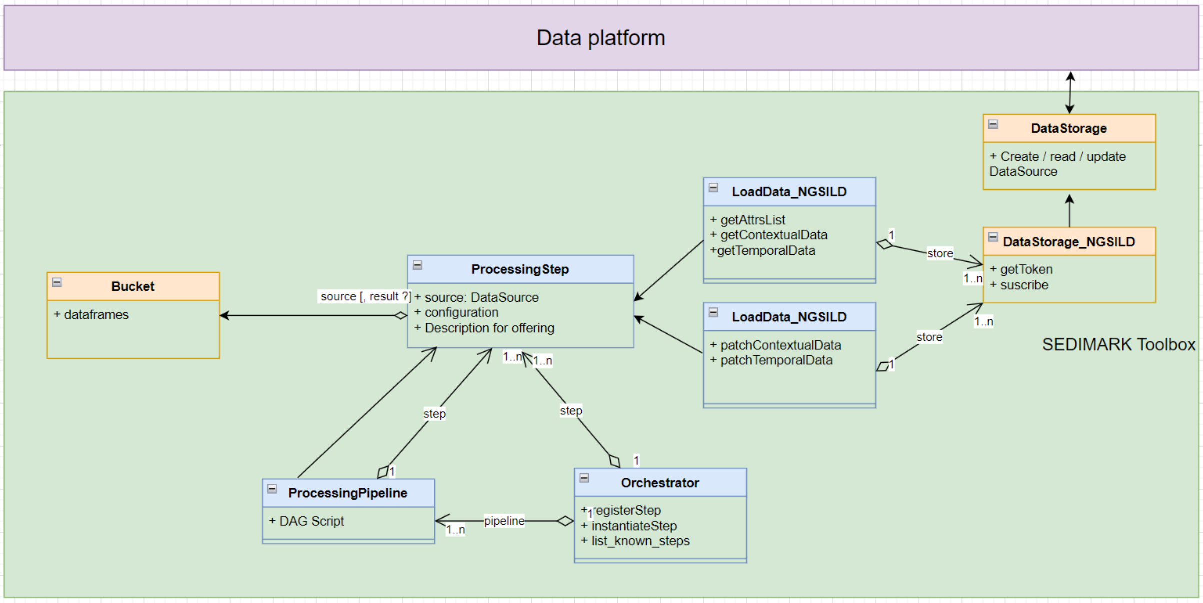


Figure 19: SEDIMARK Data processing pipeline.

6.4.1 Data processing orchestration

The data processing pipeline will be orchestrated by an orchestrator class that can register and instantiate processing steps. The orchestrator uses a “bucket” dictionary that is initialized when a pipeline is built and that will be given as parameter to each processing step. Each processing step can then modify values within the bucket or add some elements, to be passed on to the next processing step (the bucket is the SEDIMARK internal data format).

A class diagram for the SEDIMARK processing pipeline is shown in Figure 19.

6.4.2 Data adapter

The SEDIMARK toolbox will provide means to authenticate to an NGSI-LD context broker, to load data from it, converting the NGSI-LD format to an internal format of the data processing pipeline (i.e. represented by a custom data class including a Pandas dataframe) and save data back to the NGSI-LD format. The loaded data can then be sent through the data curation pipeline, used in AI models, shared in the marketplace, saved back to the broker.

LoadData processing step adds to the bucket a list of all the attributes in the entity, a pandas DataFrame containing all the temporal data of the entity and a dictionary containing all contextual information of the entity. The contextual data dictionary allows to keep some semantic information that is lost when flattening the temporal data into a table. The goal of the SaveData processing step is to reform the NGSI-LD entity using the DataFrame and the dictionary.

6.4.3 Semantic enrichment

The process of augmenting the source of metadata concerning assets (i.e., data, data stream, AI model) with additional terms in an automatic or semi-automatic way is called semantic enrichment. Typically, such metadata is represented as a set of tags or annotations to enhance the whole asset.

By enriching our understanding of the underlying data, we can better interpret the data to provide more complete information, and also improve data interoperability. The development of semantic enrichment is motivated by the information interoperability problem. The enrichment process of data providers' metadata within SEDIMARK can be obtained through data processing, such as data labelling after prediction or clustering, if a data instance is an anomaly or a normal instance through an anomaly detection process, studying the correlation between feature through feature engineering, and so forth.

So the enrichment consists in the annotation which is a metadata associated to a part of a data such that its semantics obtained through data processing.

6.4.4 Feature engineering

The data processing pipeline is composed by a set of components to train machine learning models and provide inference and analytics using Data Assets within SEDIMARK. Feature engineering is one of the principal components in the AI pipeline. It refers to the preprocessing steps that select and transform the most relevant features, aka variables, from raw data to be used in machine learning algorithms, such as predictive models. Whence the goal of simplifying and speeding up data transformations while enhancing models accuracy.

In the following, we detail the processes of feature engineering that mainly consists of feature creation, feature extraction and feature selection that would lead to an accurate machine learning algorithm:

- **Feature creation** involves creating new features which will be most useful in the predictive model. This can be adding or removing features where existing feature are mixed via addition, subtraction, multiplication, and ratio to create new derived ones that have greater predictive power.

- **Feature extraction** involves reducing the number of features to be processed using dimensionality reduction techniques. It consists of extracting features from a dataset or data stream to identify useful information. Without distorting the original feature space, this compresses the set of features into manageable quantities for algorithms to process. E.g., constructing from a set of input features in high-dimensional space, a new set of features in a lower dimensional space.
- **Feature selection** is the process of selecting a subset of the input features, i.e., the most relevant and non-redundant features, without operating any sort of data transformation or extraction. To do so, feature selection techniques mainly analyse and rank various features to determine which ones are irrelevant and should be removed, which ones are redundant and/or correlated, and which ones are more relevant and useful for the model and should be prioritized.

In order to make machine learning work well on data within SEDIMARK, feature engineering techniques will be used in the data processing pipeline for better model overall performance and data visualization.

6.4.5 Energy efficiency

Since data processing pipelines will be handled at the edge, energy efficient techniques need to be adopted to take into consideration varying computational constraints that can reside at the edge, in comparison to resource-abundant cloud-based solutions. These techniques will include adopting efficient data structures when processing data assets, separating the metadata of the data asset from the pipeline but maintaining semantics throughout, parallelisation of batch/stream processing depending on the complexity of the data asset, and aggregation based on domain-specific attributes when energy efficiency is selected as a priority over granularity. Additional techniques like data pruning or replacing “heavier” ML models algorithms with convex optimization or sparse models where this is possible, could be investigated further in order to reduce the computation costs.

6.4.6 Data Curation Enabler

A significant part of the Data Processing Enabler is the Data Curation Enabler, which is presented in Section 6.3. It is presented as a different enabler because of its importance and the fact that it has a specific target for curating and cleaning the data.

6.4.7 Data Processing Dashboard

This module is responsible for playing the role of connecting the back-end modules of the data processing enabler with the real user. It is actually a user interface that provides easy-to-use functionalities for the user to view, analyse and process their data, allowing them to get insights about the usefulness and the quality of their data and giving them direct access to the data processing tools for processing their data. The Dashboard is mainly connected with the Data Processing Orchestrator, which has the overall coordination of the processing modules and with the Data Visualisation module that is used to visualise the data and provide easy access to statistics, graphs and metrics for data quality.

6.4.8 Data Visualisation

This module provides the necessary tools to users for representing their data in a graphical way using visual elements such as graphs, charts, timelines, so that users can get an easy view on their data and be able to identify issues with their data. The results of the data curation modules will also be visualised in a user-friendly way, so that users can see outliers and duplicates or find patterns in their data.

6.5 AI enabler

SEDIMARK's main objective is to build a marketplace for both data and services, with a major emphasis on services that exploit AI techniques as a service, i.e., for automating analytics, reasoning and event detection. The marketplace provides users with the necessary tools to run these techniques on the distributed data and infer knowledge without having access to the actual data, which can be confidential in some cases or too large and expensive for purchase. SEDIMARK will use distributed ML techniques for greater scalability and energy efficiency, minimising the server costs for running ML models on huge amounts of data or using computing power and datasets from other providers so that all together build a common machine learning model. Distributed ML architectures such as Federated Learning or Distributed SGD can be exploited, using tools such as Tensorflow-Federated, PySyft and Apache Spark. In this respect, only the model parameters will be sent to the server to be averaged and sent back to the devices for the model update during training.

Finally, Green AI techniques for minimising the energy consumption of AI models using i.e. model compression and inference latency reduction will be exploited. In the case of model compression, neural network models are compressed to smaller sizes with minimal loss of performance, requiring fewer floating operations and less time to train (and predict), thus consuming less energy. In the case of latency reduction, model latency reduction techniques (PyTorch, TensorFlow) for quantization and pruning will be used to minimize the required time for a model to make inference/predictions, also reducing the energy consumption.

The following Figure 20 summarizes the main components of AI enabler which lead to the AI services.

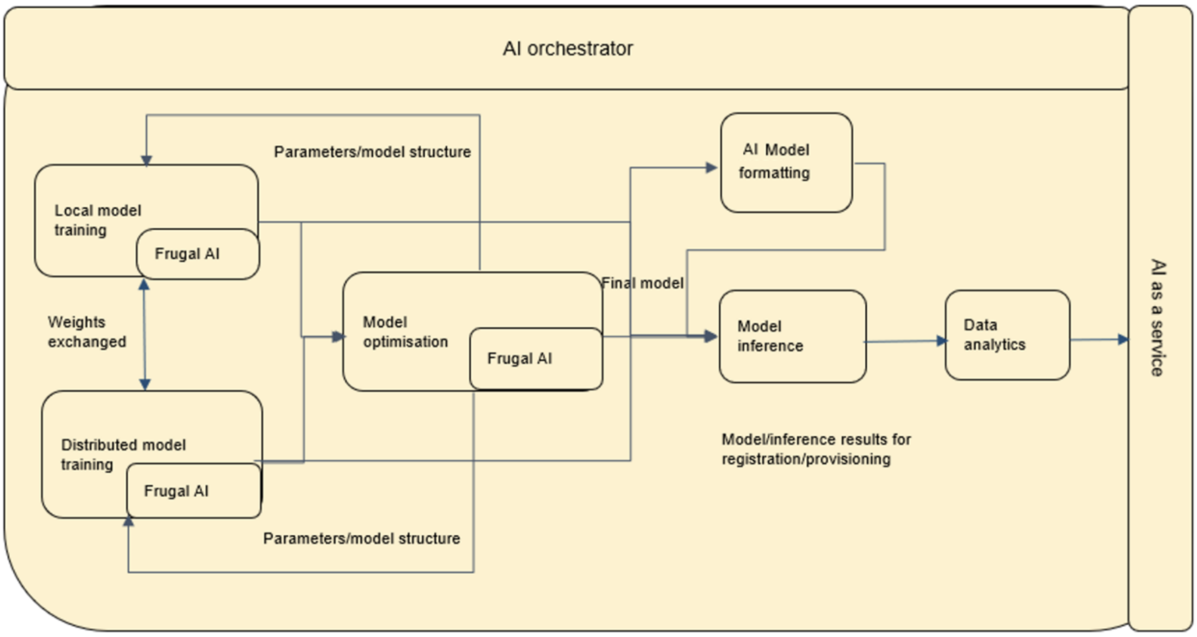


Figure 20: AI Enabler

More specifically, the AI enabler comprises the following functional components:

6.5.1 Local model training

Local model training module comprises techniques for model training locally on a SEDIMARK node with local access to training data, in order to enable a parameterised machine learning algorithm to output a model with optimal learned trainable parameters that minimize an objective function. Local models use all the available data for training, but internally separate the data into local groups and train a different model for each group. This contrasts with global models, that train a single classifier based on all training data.

6.5.2 Distributed model training

Training machine learning models using only local data can create bias and result in models that are not generalised well. Distributed training can help towards exploiting datasets from other providers and without needing to share datasets (that might contain private information) can help jointly build common models. Additionally, distributed model training can help in cases where a provider wants to train a huge model with large amount of parameters on a huge dataset, by exploiting other nodes that offer computation power and can help alleviate the load on the provider's servers by training parts of the model. Distributed model training is a module that distributes training workloads across multiple SEDIMARK nodes. These SEDIMARK nodes, referred to as worker nodes, work in parallel for the joint training process. Their parallelism can be achieved by data parallelism or model parallelism, both of which are described below.

- **Data parallelism**: In this type of distributed training, data is split up and processed in parallel. Each SEDIMARK node trains a copy of the model on a different batch of training data, communicating its results after computation to keep the model parameters and gradients in sync across all nodes. These results can be shared synchronously or asynchronously.
- **Model parallelism**: In model parallelism, the model itself is divided into parts that are trained simultaneously across different worker nodes. All workers use the same dataset, and they only need to share global model parameters with other workers, typically just before forward or backward propagation. This type of distributed training is much more difficult to implement and only works well in models with naturally parallel architectures, such as those with multiple branches.

Distributed training can be used for traditional ML models but is better suited for compute and time intensive tasks. The component provides users with the necessary tools to run these techniques on the distributed data and infer knowledge without having access to the actual data, which can be confidential or massive and expensive to purchase. Secure and privacy preserving training of such models is also investigated.

6.5.3 Frugal AI

Frugal AI is a module that provides techniques to reduce the usage of data and computer power while guaranteeing robustness within the intended field of use for a given AI model. AI systems are built using logic to implement an algorithm and data used to train the algorithmic system. Simply put, frugal AI is training AI systems with little resources. For AI, the need for frugality is usually triggered when there is little data available to begin with, or to align to internal sustainability (and cost efficiency) objectives. To move the dial on AI frugality, input frugality and/or learning frugality can be adopted.

With input frugality, which can be motivated by resource constraints and by privacy constraints, focus is placed on the data and using fewer training datasets or fewer features than needed to achieve the prediction quality achievable in a non-frugal setting. With learning frugality, emphasis is placed on costs associated with the learning process, specifically computational and memory resources. Learning process frugality is primarily motivated by resource constraints, including limited computational power and limited battery capacity.

For companies with small datasets and a mandate to move beyond experimentation, frugal AI promises to be a way to overcome this challenge. Conversely, companies with large datasets may leverage frugal AI to reign in AI scaling costs, lower their carbon footprint, and align to internal mandates. Regardless of the motivation, AI models trained with smaller datasets have the potential to reduce:

- Computational resource requirements and needs.
- Storage infrastructure and data collection and processing costs.
- Energy costs associated with training and operationalising AI systems.

6.5.4 Model inference

ML model inference is the process of inferring results based on trained and deployed ML models. The model can be processing new and unseen input data. When a model performs inference, it is producing a result based on the trained algorithm. This means that model inference is part of the machine learning lifecycle's deployment phase. The results that are inferred are usually observed and continuously monitored, at which point the model can be retrained or optimised as a separate phase of a model lifecycle.

The main considerations for model inference include:

- The data flow of input/output data.
- How the model embeds within the system architecture. This could be containerised machine learning models which draw from different system resources, or a server-based pipeline.
- Transforming input data into processable data by the model. This may require a pre-processing step.
- Transforming the output data or result into information that’s interpretable (e.g. a numerical result may need transformation into defined labels)

6.5.5 Data analytics

Data analytics module represent the final stage of the AI modelling lifecycle. It is the process of transforming, processing, and organizing raw data to gain valuable insights for decision making and provide maximum value to enterprise data. This process also helps in measuring the decisions made reducing the risks associated with making decisions based on data. Furthermore, it has become automated to a great extent, by using various known algorithms. Data analytics has gained important in recent times and data driven associations have adopted this process to better understand their target users and reduce losses by making informed decisions.

The types of data analysis are presented below:

- **Quantitative data analysis**: This data analysis technique focuses mostly on the statistical aspects of the enterprise data. This determines the pattern in the data and finds the rise and fall of data in a timeline. There are two types of quantitative data analysis: the descriptive analysis and the inferential analysis.

  - Descriptive analysis is used to find patterns that exist in particular data. This technique deals with aspects like percentages, frequencies and central tendencies to describe the data.
  - Inferential analysis is used when there is a correlation between the datasets and is required to find the differences. This includes statistical techniques like z-tests, t-tests, chi-square, etc.
- **Qualitative data analysis**: Is the opposite of quantitative analysis, this analysis technique usually deals with non-numeric data like audio, video, images, and many more. It also determines the changes that take place in data.

6.5.6 Model optimisation

Model optimisation module is the process of improving the efficiency and performance of a machine learning model through techniques such as hyperparameter tuning, careful model architecture selection, model compression, data pre-processing, and performance optimization strategies such as using a GPU, concurrency, caching, and batching. Model optimisation seeks to maximise a model's performance while minimising the computational resources and time required to train and evaluate the model. Other ways to improve the final model performance can include iteratively improving the accuracy of an AI model, lowering the degrees of error, and reducing the size by applying special optimisation methods.

6.5.7 AI orchestrator

AI orchestrator is a high-level management module of the AI/ML pipeline and acts as a single point of connection with the modules outside the AI layer. All connections to the other enablers should be made through the ML orchestrator. The main functionalities of the AI orchestrator are:

- **Local model training organisation**: Provide users with the available options to train their models based on their data. Afterwards, the users select the desired options and the orchestrator interact with the other ML modules to run the training and create the pipeline.
- **Model optimisation**: Provide users with the available options to optimise their models mainly for energy efficiency purposes or for optimizing inference based on the target deployment.
- **Organise model interoperability**: Allow users to transform the model in various formats and trigger the respective module for doing the conversion.
- **Organise model publication as an asset**: Contact the offering description/registration modules, providing the necessary information for transforming the model into an asset and details to create the offering.
- **Distributed training organisation**: Act as facilitator of the overall process (e.g. proxy between the distributed model training and the rest of the modules).

6.5.8 AI as a service

AI as a service, also called AIaaS, is a cloud-based service offering AI outsourcing. AIaaS enables individuals and businesses to experiment with AI, and even take AI to production for large-scale use cases, with low risk and without a large up-front investment. Because it is easy to start with, it makes it possible to experiment with different public cloud platforms, services, and machine learning algorithms. Another important aspect of AIaaS is that a cloud provider can offer specialized hardware and software, packaged together with the service, e.g. in a

SEDIMARK node, packaging the hardware together with the AI technique/service. Also, computer vision applications are computationally harsh and rely on hardware such as graphical processing units (GPUs) or field-programmable gateway arrays (FPGA). Buying and operating the hardware and software needed to get started with AI can be prohibitive for many organizations. With AIaaS, a company can get the AI services together with the complete infrastructure needed to run them.

The main types of AIaaS are listed below:

- **Bots and Digital Assistants**: These are a popular type of AIaaS, they allow companies to implement functionality like virtual assistants, chatbots, and automated email response services. These solutions use natural language processing (NLP) to learn from human conversations.
- **Application Programming Interfaces (APIs)**: AIaaS solutions provide APIs that allow software programs to gain access to AI functionality. Developers can integrate their applications with AIaaS APIs with only a few lines of code and gain access to powerful functionality.
- **ML frameworks**: These are tools that developers can use to build their own AI models. However, they can be complex to deploy, and do not provide a full machine learning operations (MLOps) pipeline. In other words, these frameworks make it possible to build an ML model but require additional tools and manual steps to test that model and deploy it to production.
- **No-code or Low-code ML services**: Fully managed machine learning services provide the same features as machine learning frameworks, but without the need for developers to build their own AI models. Instead, these types of AIaaS solutions include pre-built models, custom templates, and no-code interfaces

6.6 Interoperability Enabler

The SEDIMARK ecosystem will include various artefacts, such as data, AI models, offerings, and so forth. Interoperability within and across this ecosystem is thus an important functionality which ensures that each such artefact should conform to the SEDIMARK information model for that artefact.

Data interoperability is defined as the ability of systems and services that create, exchange, and consume data to have clear, shared expectations for the content, context, and meaning of that data. Thus, it allows access and processing of data from multiple sources in diverse formats without losing sense and then integrates that data for mapping, visualization, and other forms of representation and analysis. Similarly, with AI models, interoperability between various environments and platforms would create a common understanding and interconnectable AI models. To ensure interoperability between these data assets, SEDIMARK will use a formatting process to convert input data/AI models from the different providers to a common data/AI model format within SEDIMARK.

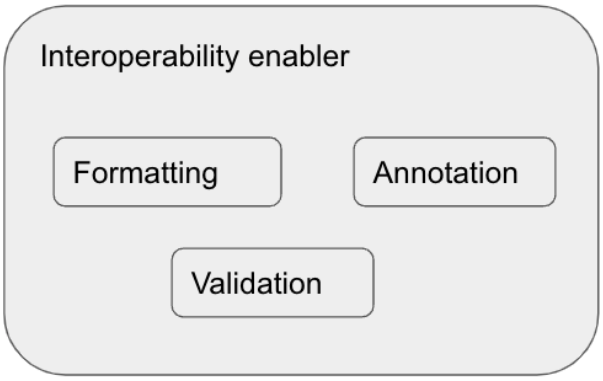


Figure 21: Internal structure of the Interoperability Enabler.

In order to enrich these data assets, an annotation process will take place to add metadata, e.g., about the quality of data and of AI models, based on the analyses performed on these assets in the other enablers, mainly the Data processing enabler and the AI enabler. Besides, to integrate the SEDIMARK marketplace, these assets need to be validated by considering a set of criteria and an information model for each one of them.

The SEDIMARK Interoperability Enabler groups these aforementioned functionalities and is illustrated and detailed in the following Figure 21 and subsections, respectively.

6.6.1 Formatting

Formatting within SEDIMARK concerns principally the data asset format considered within the marketplace in order to ensure its compatibility with multiple software and environments. This is discussed below and illustrated in the following Figure 22.

- **Data formatting**: The Data formatting component will translate the data expressed in various formats provided by providers into the SEDIMARK format. The NGSI-LD format is the one adopted within SEDIMARK, which will make the heterogeneous data easier to process mainly within the Data processing enabler and in interaction with the AI enabler.
- **AI model formatting**: The interoperability enabler will also provide tools to improve the interoperability of AI/ML models, acknowledging that not all data providers/sources will use the exact same models and the same software to train/run the models. In the decentralized environment of SEDIMARK, ensuring that all users will use the exact same ML platform for training the model and the same machines is unrealistic, so, SEDIMARK models should be agnostic to underlying platforms. SEDIMARK will provide tools for AI model formatting that aim to convert models to and from a reference format and support models to run on machines of various capabilities and platforms. Interoperability between and across different platforms also helps ML models to be more flexible. SEDIMARK considers the use of ONNX formatting that enables the seamless transfer between AI frameworks and ML models. The benefits of ONNX are numerous; it allows to move models between different frameworks without having to rewrite code, which makes easier to develop and deploy models across different platforms. This helps to ensure that models are interoperable across different frameworks and platforms.

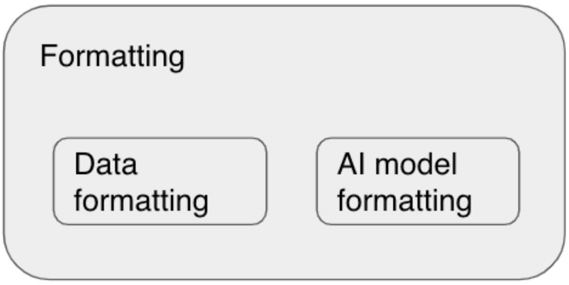


Figure 22: Internal structure of the Formatting module of the Interoperability enabler.

6.6.2 Annotation

The annotation component will add annotations principally resulting from the Data processing enabler and the AI enabler. These annotations are presented in the following Figure 23 and detailed as follows:

- **Data quality annotation**: Data quality annotation consists in adding annotations to the data within SEDIMARK based on some set of data quality metrics. Considering these metrics, annotations regarding the data quality will be added.
- **ML-oriented data quality annotation**: AI and ML models are made to discover patterns and relationships in data, and they can only perform well if the data is of good quality and has been correctly labelled and annotated, because the quality of data utilized to train these algorithms has a significant impact on their performance and accuracy. The quality of data to be used by ML algorithms will thus be investigated. So, high-quality data and data annotation would lead to multiple benefits, notably faster data training, fewer mistakes, and better accuracy and precision. In this context, building accurate models and ensuring their effectiveness requires good data quality. On the other hand, poor data quality can produce biased or erroneous models, which will have poor performance and unreliable predictions.
- **Semantic annotation**: The semantic annotation process of data will lead to significantly better interoperability and linkage of the data from various sources. The process of annotation enhances the data quality by, inter alia, adding information in the form of metadata.
- **AI model quality annotation**: The AI model quality annotation consists of annotating the AI model depending on its performance and accuracy obtained from the AI enabler. In this context, AI and ML models can be optimized and enhanced by taking into account the AI quality model annotation.

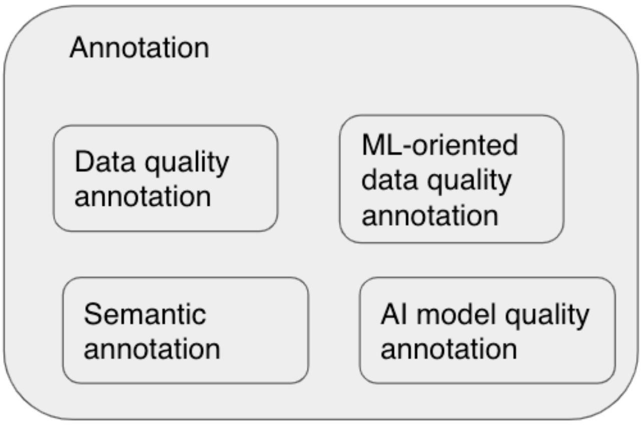

Annotation
Data quality annotation
ML-oriented data quality annotation
Semantic annotation
AI model quality annotation

Figure 23: Internal structure of the Annotation module of the Interoperability enabler.

6.6.3 Validation

Validation is necessary to ensure compliance of artefacts inside SEDIMARK with respect to their corresponding information models. In this context, we distinguish (see also the following Figure 24):

- **Data validation/certification**: SEDIMARK will ensure that datasets or data streams made available through the marketplace are validated against a set of criteria based on essential requirements for syntactic and semantic interoperability, and domain-specific knowledge that can be used to detect irregularities in relation to value ranges and data annotations. In the case of data streams, routine auditing will be applied to check the consistency of the data quality and conformance. Once the data asset has been validated, a certification will be issued. Certification will also provide a rating of the quality of the data asset based on the analysis of the data processing applied to the data asset. The rating will be set relative to the number of validation tests applied to the data asset.
- **AI model validation**: SEDIMARK will ensure that an AI model's design, format, and description made available through the marketplace conforms to a reference AI information model.
- **Offering validation**: Similarly to Data validation and AI model validation, offerings within the SEDIMARK Catalogue will be validated using a common information model.

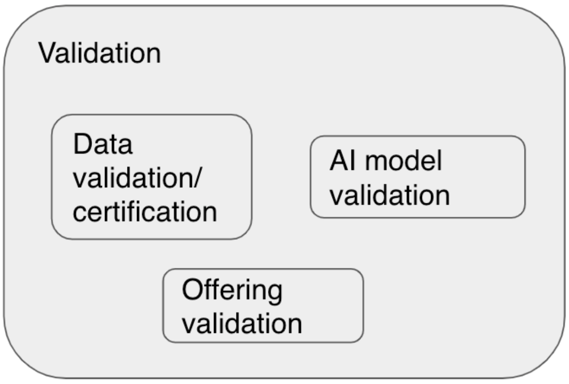


Figure 24: Internal structure of the Validation module of the Interoperability enabler.

6.6.4 Interactions with other enablers

Principal interactions between the functional components of the Interoperability enabler and other SEDIMARK enablers are showed in the following Figure 25 and Figure 26.

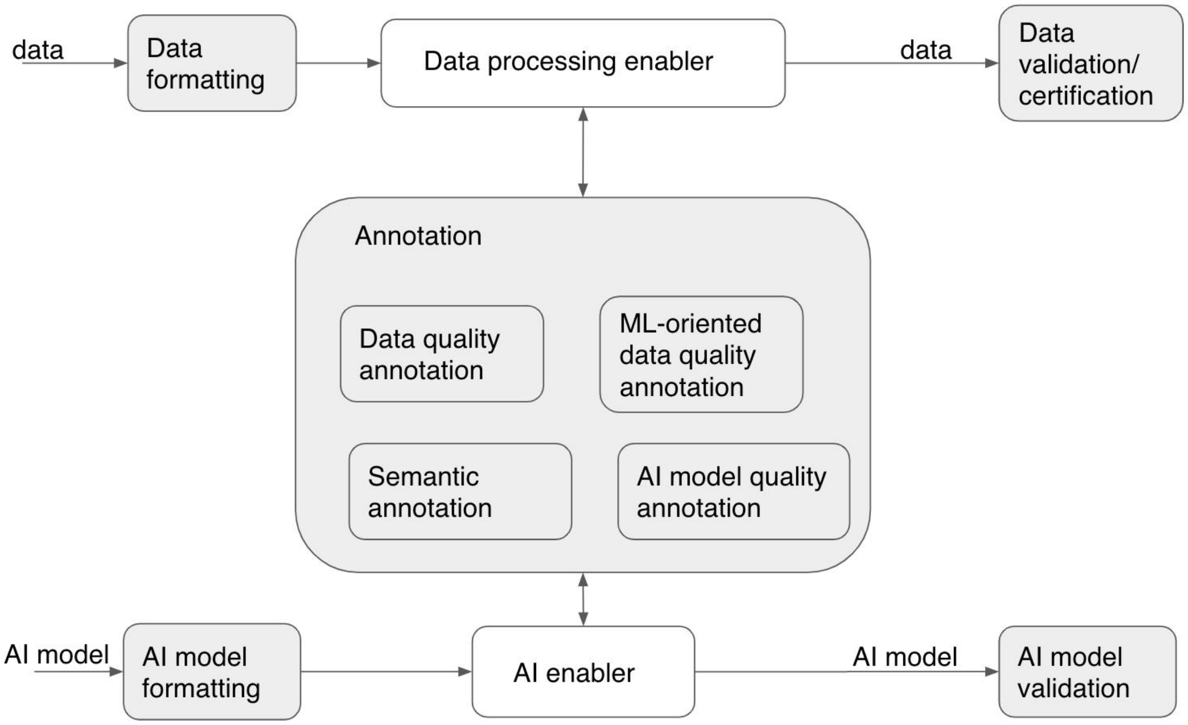


Figure 25: Interconnection of the Interoperability enabler with the Data Processing enabler and the AI enabler.

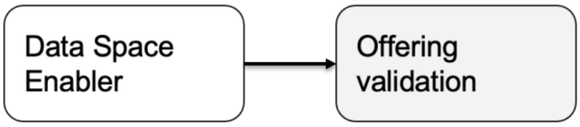


Figure 26: Interconnection of the Offering validation module with the offering enabler

6.7 Data Space enabler

The Data Space Enabler (DSE) is the main gateway for any participant to interact with a Marketplace. This functional entity provides the technical means to ensure a secure and trustworthy asset exchange between participants, enabling them to become either Consumers or Providers. Hence, the functional components that are part of the DSE deal with most of the interactions in which either an offering or an asset is involved. That is, the DSE handles all of the phases from an Offering life-cycle, which includes the creation, registration and discovery, as well as the agreements between participants to exchange assets. Furthermore, it is also in charge of handling the asset exchange once two parties agree under a defined set of terms and conditions.

As most of the interactions carried out within the Marketplace are meant to be done between DSEs, this functional entity must provide the corresponding interfaces to interact between them, as well as with a participant, either directly or indirectly (i.e. through other functional entities such as the Marketplace Enabler through an API). On the other hand, most of the functional components are meant to support one participant role, Consumer or Provider. Table 13 shows the relation between the components and the participants' role they support.

Table 13. Relationship between functional components and participants role

Functional Component	Consumer	Provider
Asset Request	X	
Asset Provisioning		X
Offering Discovery	X	
Offering Registration		X
Offering Description		X
Offering Sharing		X
Contracting	X	X

The DSE is divided in two different planes: the Asset and Offering Control planes. The Offering Control plane deals with offerings, and provides the functionalities to generate (i.e. describe an asset or a set of assets using the Marketplace Information Model) and register (i.e. store the offering in the self-description and the DLT) an offering. Besides, it also includes the interfaces to receive any request and deliver existing offerings, and implements the means to enforce the rules established by the agreement for the shared assets. On the contrary, the Asset Control plane is in charge of requesting and providing the set of assets described in the offering from/to another participant.

The interactions between components are shown in the diagram from Figure 27. In here, the interactions with other entities, such as the DLT Enabler, Marketplace Enabler and Storage Enabler, as well as other Data Space components, are shown. As it can be seen, the Data Space Enabler is in charge of any interaction the participant has to carry out with other entities. In this sense, the DSE does not provide any GUI, but a backend and a set of APIs that other components can use.

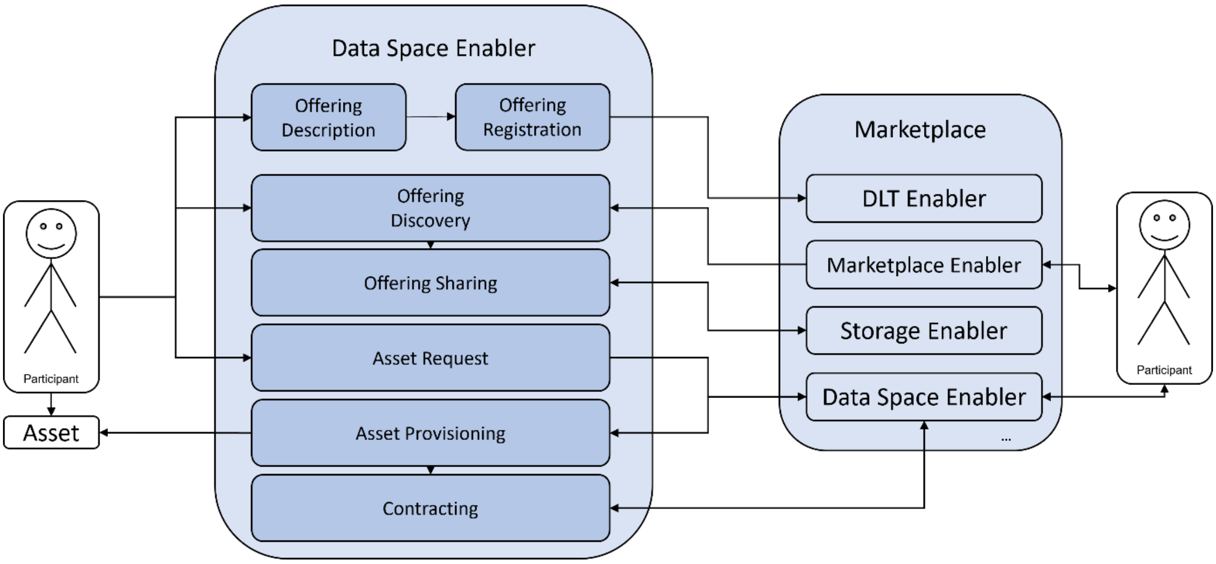


Figure 27: Interaction between the components of the Data Space Enabler and the Marketplace

6.7.1 Asset Request

The Asset Request functional component enables any participant to request an asset once an agreement has been reached. It provides two interfaces: one of them to communicate with another DSE; and an additional one to enable a participant or other functional entity to request an asset.

6.7.2 Asset Provisioning

It is the counterpart of the Asset Request component, and enables any provider to deliver the requested asset. Any request must be validated (i.e. a valid agreement has been reached) before being provided. Hence, it has to communicate with other functional components, including the Contracting, to ensure that an agreement has been reached, and it is still valid.

6.7.3 Offering Discovery

Enable the offering discovery functionality, providing an interface to request information from a Distributed Catalogue. It provides an interface that can be used directly by a participant to communicate with the Distributed Catalogue, or by other functional entities providing a GUI to ease the offering discovery process.

6.7.4 Offering Registration

This functional component is in charge of registering an offering locally (including it as part of the Self-Description), and into the Registry. Offerings are received by the Offering Description component; thus, no validation of the content is being performed, but to carry out any process on top of the offering (e.g. signing a Verifiable Credential), to ensure its trustworthiness and follow the SEDIMARK guidelines for the DLT enabler.

6.7.5 Offering Description

The Offering Description is a helper to build offerings based on existing assets. Offerings must follow the Marketplace Information Model to ensure the interoperability within the marketplace. Therefore, this component also communicates with different functional entities to validate that the offering is compliant with the information model.

6.7.6 Offering Sharing

The Offering Sharing functional component provides the functionalities to support the offering description sharing, exposing the corresponding interfaces to allow other authenticated and authorised functional entities to obtain the Participant's Self-Description and the existing offerings. Accessing this information can be done directly by other DSEs, or a Distributed Catalogue to index existing offerings.

6.7.7 Offering Statistics

The Offering Statistics provides an interface for other functional entities to obtain statistical details about the offerings from a provider, including among other the number of times an offering has been requested, how many offerings are being provided or the number of agreements reached per offering.

6.7.8 Contracting

The Contracting functional component provides the interfaces to negotiate the contracts and reach an agreement regarding an existing offering between participants. It also implements the mechanisms to ensure that the conditions of an agreement are being met.

6.8 Storage Enabler

The Storage Enabler handles and manages the storage of the various artefacts produced by the other enablers of the architecture. It handles functionalities in the Data layer, Services layer and Distributed Storage layer. It mainly interacts with the Data Processing enabler, AI enabler, DLT enabler, and Marketplace enabler.

6.8.1 Storage Artefacts

The first step in defining a Storage Enabler is to identify the artefacts that will be produced with the SEDIMARK marketplace ecosystem in relation to the other Enablers that will depend on persisting these artefacts.

The main artefacts that need to be stored in relation to the Enablers are shown in Table 14.

Table 14. Storage for Assets and Artefacts

Architectural Layer	Enabler	Asset/Artefact	Persistence
Data	Data Processing	Dataset	Relational (Time-series)
			Relational (Document)
		Data Stream	Buffered, In-Memory
		Processing Artefacts	Object Storage
		Configuration	File-based
Intelligence	AI	Models	Model Registry
		Configuration	File-based
		Training Artefacts	File-based, Object storage
Interaction	DLT	Asset Hash	Relational
		Offering Hash	Relational
Services	Data Space	Offering Description	Document
			Graph
	Marketplace	Logs	Document

More detail regarding the artefacts are given below based on which enabler they belong:

- **Data Processing**: This mainly relates to the storage of Data Assets. These assets can be advertised as unprocessed i.e. original, or processed, through a pipeline handled by the Data Processing enabler. The Storage enabler needs to ensure assets are stored in a consumable manner before consumption by a Participant or a Tool from the Data Processing enabler. Data Assets manifest in mainly two forms, Datasets and Data Streams. Data Streams can be consumed or processed atomically, i.e. by individual data points, or by windows or batches of data points, which could be a prerequisite for further processing, e.g. aggregation. Another subset of artefacts here relates to structured configuration information that is used for processing data assets in a specific manner. This structured information caters to the domain of interest the data is originating from (i.e. the use case scenario), attributes that indicate normal ranges and limits, and also the dynamicity of the data. Configurations can be either generic or domain-specific.

- **AI:** These artefacts mainly relate to the AI Model Assets generated from an AI Service. AI Models are usually stored in Model Registries that can support the loading as well as storage of models for on-demand inference. Another set relates to the storage of intermediate datasets and other artefacts that are produced during the Data Processing or AI Model training pipelines, which are consumed by intermediate or cascading components. Prior to storage, artefacts could require serialisation for it to be consumable by other components.
- **DLT:** These are artefacts related with the SEDIMARK Registry and include items that are stored on the DLT, i.e. asset and offering hashes.
- **Data Space**: The artifacts here relate to the administration of the marketplace. This includes the registration of participants such as Marketplace Operators, Data Producers, Data Providers, Data Consumers and Service Providers. The second set of artefacts relates to the advertisement of offerings by Data Producers and Service Providers in the form of structured descriptions that comply with the information model that describes the assets and services available.
- **Marketplace**: The artifacts here relate to the logs that are being kept in the marketplace with respect to user's activities, history, preferences, etc.

6.8.2 Local and Distributed Storage

It branches into two main sub-modules, local and distributed. Local storage handles the basic storage needs for the data producer/provider to persist their data asset, local catalogue descriptions and artifacts created throughout the data processing pipeline. The distributed storage element handles the storage of artifacts generated from data producers and AI/ML services on other participants' connectors/domains.

Figure 28 illustrates the storage modules within a Participant node for local and distributed storage.

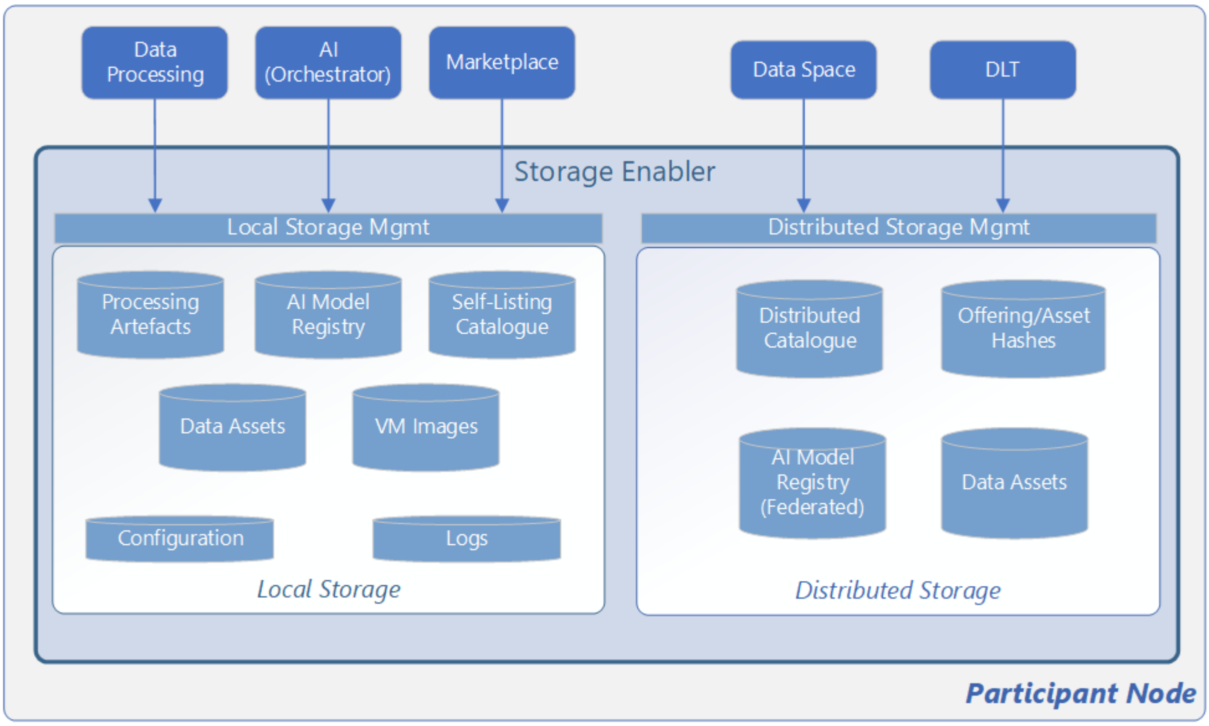


Figure 28: Local Storage and Distributed Storage Modules

6.9 DLT Enabler

SEDIMARK intends to design, develop, and test a decentralized marketplace. The basic functional entity for enabling such a kind of interaction is the Distributed Ledger Technology (DLT) with its decentralized technical capabilities and data immutability feature. SEDIMARK DLT Enabler is composed of the building blocks reported in Figure 29.

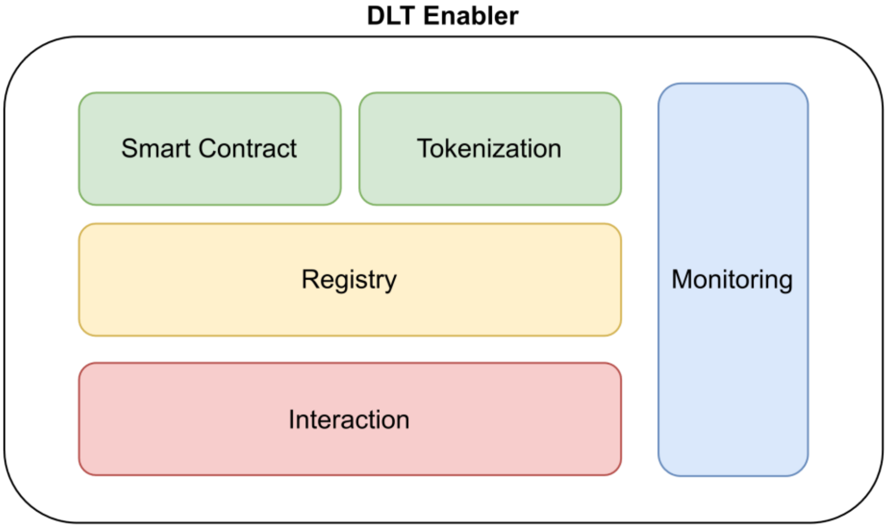


Figure 29: Block Diagram of the DLT enabler.

More in detail, the DLT Enabler groups the following functionalities:

- **Registry**: the distributed ledger to provide trust, non-repudiable and immutable information about Participants and Offerings. SEDIMARK leverages the IOTA DLT solution. IOTA overcomes known scalability bottlenecks in its distributed ledger by using partially ordered data structures based on Directed Acyclic Graph (DAG). The Two essential layer 1 (L1) components are the IOTA Ledger and the IOTA Tangle [25]. The former is (unspent transaction output) UTXO ledger providing scalability and high throughput, whereas the latter is the permissionless, feeless and miner-less consensus protocol based on DAG. The Tangle consensus protocol is based on validation of the ledger by the users of the ledger themselves, not by miners. The IOTA DLT solution adds to the L1 a new layer 2 (L2) called IOTA Smart Contract (ISC) [26]. It is a framework that extends the base protocol of L1 of the IOTA DLT by introducing multiple programmable ledgers on top as L2. The result is a multi- chain environment where all chains are anchored on the IOTA Ledger at L1. Each chain is a separate blockchain with smart contracts on it, functionally equivalent to Ethereum smart contracts and fully composable between each other. The ISC environment also enables the thrustless transacting and composability of smart contracts between different chains over L1 with high throughput and high scalability.

- **Interaction**: the software component at any Connector that enable interaction with the IOTA DLT. The overall component is formed by the IOTA Client to interact with L1 to issue data transactions and by the Wallet to interact with L1 to issue value transactions (i.e., native IOTA tokens) and with L2 ISC to exchange value (i.e., native and data tokens) and interact with the smart contracts.
- **Smart contract**: is a software application that operates on the L2 decentralized network of validators who execute and validate the same code reaching a consensus on the same valid output. One notable characteristic of such applications is their deterministic execution, ensuring that running the same code on multiple validators results in identical outcomes. Smart Contracts are deployed on the immutable L2 ledger which means that once they are published, nobody can tamper with the code.
- **Tokenization**: is the key feature implemented through ERC-721 [27] and ERC-20 [28] Smart Contracts to tokenize assets and made them tradable over the marketplace. ERC-721 is responsible for minting an NFT representing a specific Offering of a Provider whereas ERC-20 Smart Contract is responsible for minting a certain number of ERC-20 tokens (i.e., datatokens) for the owner of the NFT (i.e., the owner of the related asset). Providers gives access to an asset against proof of purchase of the related Datatoken.
- **Monitoring**: is a software component that constantly check the IOTA DLT (both L1 and L2) to collect the evidence (i.e., in principle all kind of transactions) to describe the behaviour of participants and of the key elements of the marketplace.

6.10 Trust enabler

SEDIMARK intend to design, develop, and test a secure, decentralized marketplace where a participant can enter into a trusted interaction with any other participant. In principle, there isn't *a priori* trust among the participants, hence the marketplace must ensure such an important feature. The Trust Enabler, shown in Figure 30, is devoted to such target.

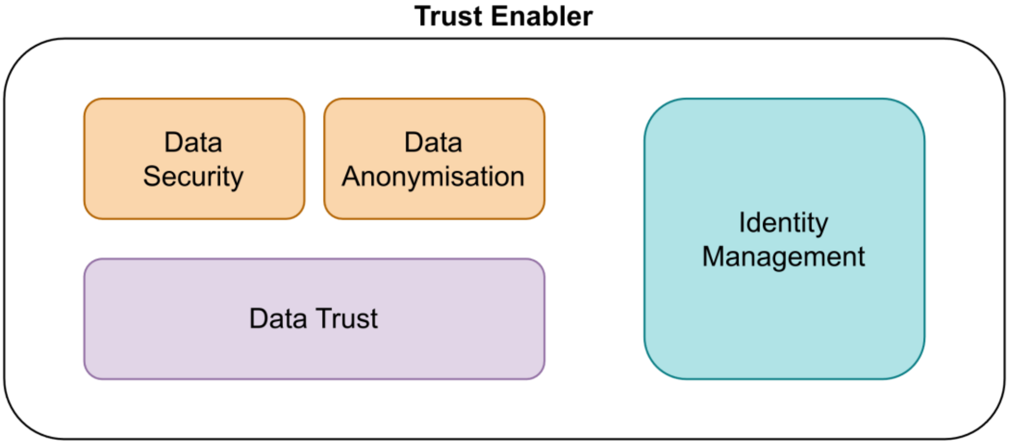


Figure 30: Block Diagram of the Trust Enabler

The Trust Enabler groups the following basic functionalities:

- **Identity Management**: digital identity is compliant with the Self-Sovereign Identity (SSI) [29] model standardized by W3C [30] [31] [32]. SSI is a decentralized digital identity paradigm that gives both human beings and things full control over the data they use to build and to prove their identity. SSI leverage any DLT acting as the Root-of-Trust (RoT) for identity data. In fact, DLTs are distributed immutable means of storage by design.
- **Data Security**: provides all cryptographic primitives needed for building security all over the marketplace. Basic crypto operations are digital signatures (RSA, ECDSA and EdDSA), hashes (SHA2 and SHA3) and asymmetric (RSA) and symmetric encryption/decryption (AES128 and ChaCha20).
- **Data Trust**: provides an algorithm to calculate and verify a proof of authenticity of data assets. It leverages the previous discussed primitives. This algorithm is intended to provide different element of the marketplace to prove and/or verify the data they are handling are the intended ones. The main purpose is to build trust in data all over the marketplace.
- **Data Anonymisation**: The anonymization process in SEDIMARK focuses on safeguarding sensitive information while maintaining data integrity and relationships. Anonymisation can be split in the following phases: In the first phase, entity names are anonymized by replacing them with generic labels ('client') followed by unique numerical codes. This step ensures that the identity of individuals or entities remains hidden while preserving uniqueness. The second phase involves mapping the anonymized entity names to corresponding entities based on similarity, effectively concealing the actual entity names while retaining linkages. Subsequently, other sensitive numeric attributes are anonymized by converting them into ranked integer values. This transformation obscures the original values, making it challenging to identify specific individuals or entities. Finally, the anonymized values are mapped to their counterparts, preserving anonymity and enabling secure data analysis and sharing without compromising data privacy. This anonymization process enhances data protection and confidentiality while maintaining the usefulness of the datasets for various analytical purposes.

6.11 Marketplace enabler

The marketplace is the main entry-point for users to interact with the various components of the SEDIMARK platform. It aims at providing intuitive graphical user interfaces enabling the key core functionalities of SEDIMARK, which can be split in two categories: the data spaces functionalities, consisting of the essential features shared by all data spaces, and the SEDIMARK specificity, gathering all original features proper to SEDIMARK ensuring its added value compared to other projects listed in section 3. In more details:

- **Data Space functionalities:**
  - New participants registration: ensuring the delivery of verifiable credentials from their digital identity, but also ensuring participants' self-descriptions are compliant with SEDIMARK's standards.
  - Offerings catalogue browsing.
  - New offerings registration: the marketplace will provide tools to ease the process of writing compliant descriptions of offerings and ensure their secure and trustful access.
  - Contracts negotiation between offering providers and consumers.
  - Offerings management dashboard: where participants can have an overview of their offerings and transactions.
- **SEDIMARK added value:**
  - Offerings monitoring and statistics: augmenting the offerings management dashboard with insightful data about transactions' status and offerings' usage.
  - Offering recommendations: highlighting specific offerings based on users' interests.
  - Access to AI and data processing toolboxes: providing a friendly graphical user interface to test or run data processing pipelines, to enhance datasets before providing them as offerings.

Because of this wide set of functionalities, the marketplace enabler interacts with all other functional entities, and therefore needs a substantial number of components constitute its backend as shown in Figure 31 (marketplace architecture), in order to handle these interactions as loosely as possible. For most of the data space core components, SEDIMARK leverages existing solutions in line with the Data Space Business Alliance, especially the Gaia-X portal architecture [33], in order to ensure interoperability with other data spaces. These modules are:

- **Onboarding service**: it takes care of the registration of new users, and therefore interacts with the trust services to register their digital identities and provides them with verifiable credentials.
- **Administration service**: the entry point for federators to perform admin level operations.
- **Discovery service**: responsible of the offerings' discovery.
- **User account service**: to manage users' data.
- **Self-description service**: used to create/edit descriptions of offerings.
- **Lifecycle management service**: it manages the lifecycle of users' provided or consumed offerings.
- **Dashboard service**: it keeps track of the provided and consumed offerings of users and allows them to edit their descriptions and manage their lifecycle via the lifecycle management service.

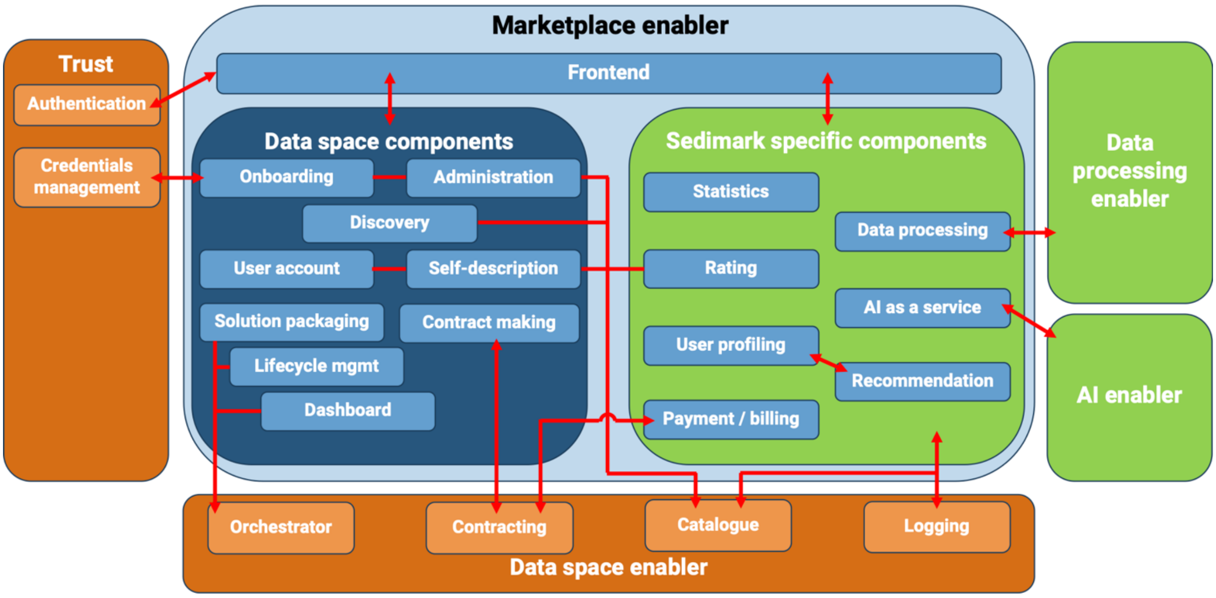


Figure 31: Marketplace architecture

In addition, the marketplace will rely on modules dedicated to specific SEDIMARK features (shown in green in Figure 31 marketplace architecture):

- User profiling: it is a service that logs information about users' navigation within the marketplace, in order to feed the recommendation service.
- Recommendation: it is a service that selects specific offerings the users may be interested to consume. Such recommendations are built upon users' catalogue browsing pattern.
- Statistics: it reports the activity around provided/consumed offerings of users. For instance, a provider may consult how many views her/his offerings attracted.
- Rating: it is a service that enables users to rate the offerings they consumed, and augments the catalogue displayed in the marketplace with ratings from other users.
- Payment/billing: it is a service that provides an interface for participants to pay for the offerings they consume, based on the payment policies defined in them.
- Data processing service: this component interacts with the data processing orchestrator to enable users to manage their processing pipelines and monitor their status.
- AI as a service: similarly to the data processing service, this component interfaces with the AI orchestrator to manage machine learning tasks the users may want to perform as a service, or on its offerings.

The marketplace is designed to cover the needs of three types of users persona: visitors, offering providers and offering consumers.

- Visitors: such users are not registered within the SEDIMARK ecosystem. Consequently, they do not need to log in the marketplace, and therefore do not have access to restricted offerings, and cannot publish/consume offerings. However, they can browse the catalogue of publicly available offerings. Despite their limited access to a subset of the marketplace features, the consideration of this persona is key in its design. It stresses out how the marketplace must be able to showcase the SEDIMARK platform to prospective participants. Moreover, the activity of such users may be monitored by the user profiling service in order to identify popular offerings.

- **Offering providers**: participants registered in the SEDIMARK ecosystem, willing to share datasets or services as offerings within the marketplace, either publicly or restricted to designated participants, either free of charge or as a paid service. The AI and data processing components empower such users with tools to improve the quality of their offerings prior to their publication. These participants' activities may cover all components of the marketplace, yet they emphasize a special care needed on the offering management in the dashboard, and on the AI and data processing toolboxes access.
- **Offering consumers**: participants registered in the SEDIMARK ecosystem, interested in buying offerings of other participants. Their primary interest focuses on the offering discovery, requiring an intuitive catalogue browsing experience, therefore a special attention on search filters and recommendations. The dashboard is also relevant to such persona, enabling them to monitor their active contracts. AI and data processing tools may also be of interest to such users, although not to the extent of offering providers.

6.12 Open data enabler

The Open data enabler aims at making datasets from open data portals available to SEDIMARK ecosystem participants. To fulfil this goal, it acts as a participant, enriching the catalogue with offerings corresponding to each portal. As shown in Figure 32 (Open data enabler working principle), this open data enabler participant (in orange) hosts in its premises a set of adapter modules, implemented following an abstract interface defined to adapt the external portals' API to the internal SEDIMARK protocols.

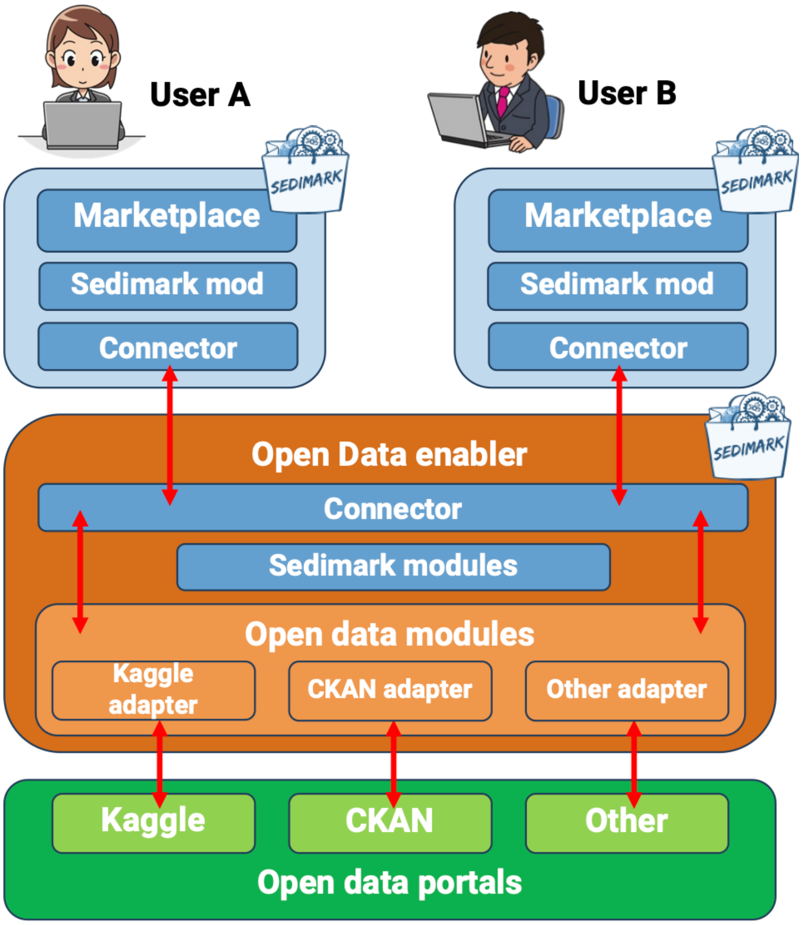


Figure 32: Open data enabler working principle.

Each adapter maps to an open data source (in green), corresponding to a provided offering in the catalogue, free and accessible to every participant with no restriction. Any participant in the SEDIMARK ecosystem can consume these offerings, by accepting the contracts defined by the open data enabler. The signing of such contract triggers the transaction, where the relevant adapter is used to transfer the data from the relevant open data portal API to the open data enabler's connector, so the latter can transfer data to the consumer's connector.

This approach has the following benefits:

- **Scalability**: new portals can easily be added simply by implementing their corresponding modules.
- **Flexibility**: even if adapter modules all follow the same interface, they are independent, so any update or modification of a specific adapter does not affect the others.
- **Consistency**: open data offerings are no different compared to any other provided offerings.
- **Self-contained**: since the open data enabler is a participant, with its own premises, updating it does not require to propagate these updates over all participants premises. It can also use a lightweight version of the SEDIMARK platform, focused solely on providing the open data offerings. For instance, it does not require the AI or data processing enablers, nor the marketplace enabler.

6.13 SEDIMARK relation to other EU initiatives

SEDIMARK is closely related to different data space and marketplace initiatives from other projects. The following subsections shows the similarities between the concepts and the functional architecture defined in SEDIMARK and some of the most important initiatives nowadays. It has to be noted that in all the figures in this subsection, the SEDIMARK functional components are depicted with a bold red outline in order to differentiate them from the components of the other initiatives/projects.

6.13.1 SEDIMARK relation to IDSA

IDSA aims to establish an open standard for the exchange and sharing of data, ensuring data sovereignty and a secure exchange, named IDS. SEDIMARK is aligned with the vision of IDS to implement a secure and decentralized data exchange marketplace. However, while most of the underlying concepts of data exchange remains similar, there are significant differences between the scope of both initiatives.

One of the main aspects is that IDS does not consider or establish any guideline on data formatting and the communication interfaces needed for their provision, beyond the models associated with the metadata and data exchange control aspects. Hence, there is no definition for a Data Asset Information Model in IDS, but only for the metadata associated to the exchange. While in SEDIMARK there is no Data Asset Information Model enforcement, it is required in case the provider wants to leverage the functionalities offered by the Data Processing Enabler (e.g. data curation). Finally, Data enrichment for data assets and AI model generation, crucial aspects of SEDIMARK, are not addressed by IDS.

On the other hand, IDS standard is solely focused on the provision of data, while SEDIMARK foresees the provision of other type of services, such as the ones bases on AI, including distributed learning and ML models.

Finally, it is worth highlighting one of the main differences in the support of Identity an Access Management (IAM) in SEDIMARK and IDS, as per the current version of the IDS-RAM, security relies on a central component as identity provider, which delivers temporal tokens to be used in transactions. Conversely, in SEDIMARK, IAM relies on a decentralized infrastructure leveraging DLTs and trusted anchors to support IAM mechanisms within a marketplace.

So as to show how the components relates to each other between the functional architecture of SEDIMARK and IDS, Figure 33 shows the system view of the IDS architecture and their relationship to SEDIMARK functional entities. It is worth mentioning that the Trust and DLT Enabler are not represented as the system view of the IDS architecture does not include the Identity Provider for readability. The relationships between SEDIMARK terms and the ones proposed by IDS can be found in Section 4.

In recent proposals from IDSA, such as the Data Space Protocol, a decentralised infrastructure to support security mechanisms is introduced. Nonetheless, this is still under development and there is not a consolidated version yet published.

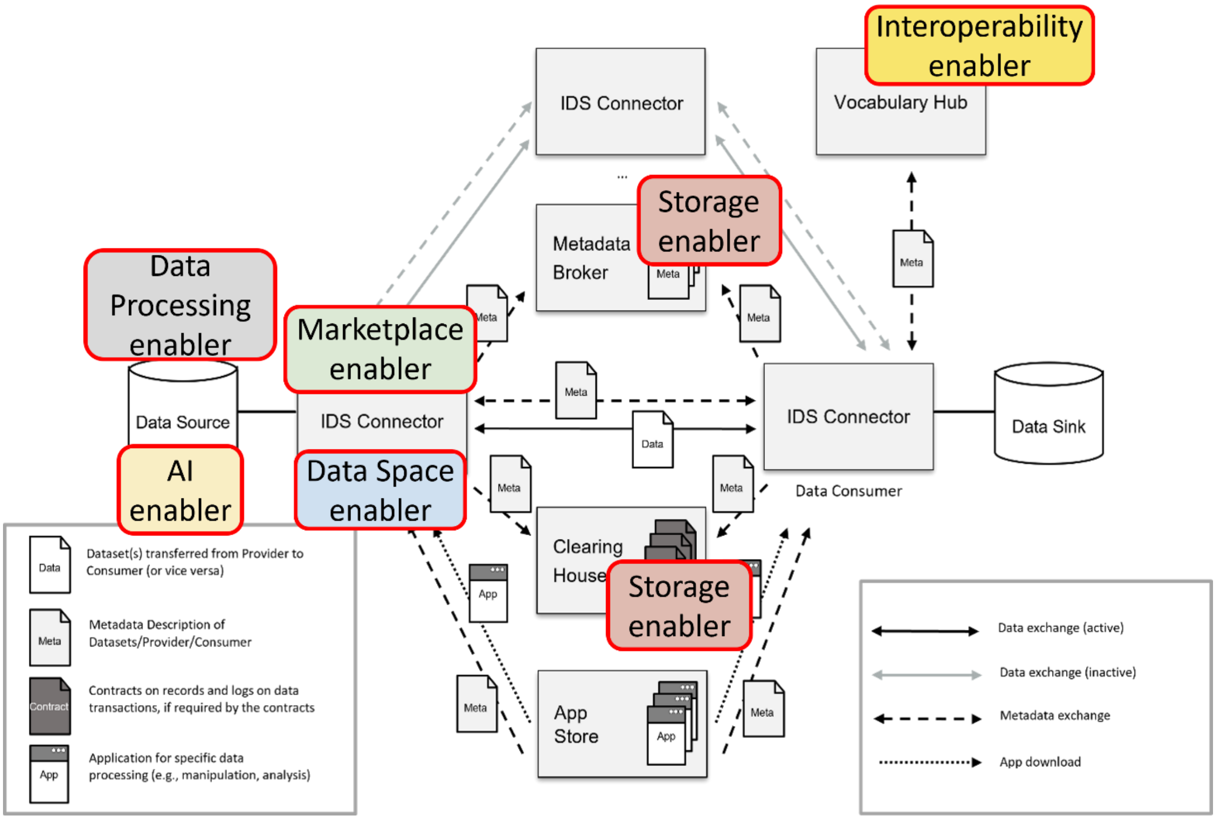


Figure 33: IDS standard system view of the architecture and SEDIMARK functional entities relationship

6.13.2 SEDIMARK relation to GAIA-X

GAIA-X is a European initiative established to develop a comprehensive framework to foster the creation of an efficient, competitive, safe, and trustworthy data infrastructure for Europe, based on a federation of cloud services. One of the main aspects of the proposed framework, similarly to IDS and SEDIMARK, is related to the sovereignty principle in any data exchange, which is also pursued under the SEDIMARK framework proposal. Besides, Gaia-X also introduces the concept of service provision, such as the one from cloud infrastructures, that is foreseen in SEDIMARK.

SEDIMARK and Gaia-X also share the vision of a decentralised trust framework, leveraging decentralised infrastructures, such as the DLT infrastructure proposed in the functional architecture, to support IAM in SEDIMARK.

On the other hand, Gaia-X does not define the technical components in charge of providing access for data and service exchange beyond the ones being part of supporting the federation and trust framework, but is open to any other implementation as long as it follows the guidelines introduced by Gaia-X related to trust and their conceptual model (equivalent to the Marketplace Information Model proposed in SEDIMARK). Likewise, there is no proposal for an Asset or AI Model Information Models, nor for the functional entities to leverage AI models. Therefore, future implementations of SEDIMARK can become part of deployed Gaia-X ecosystems for data and service exchange.

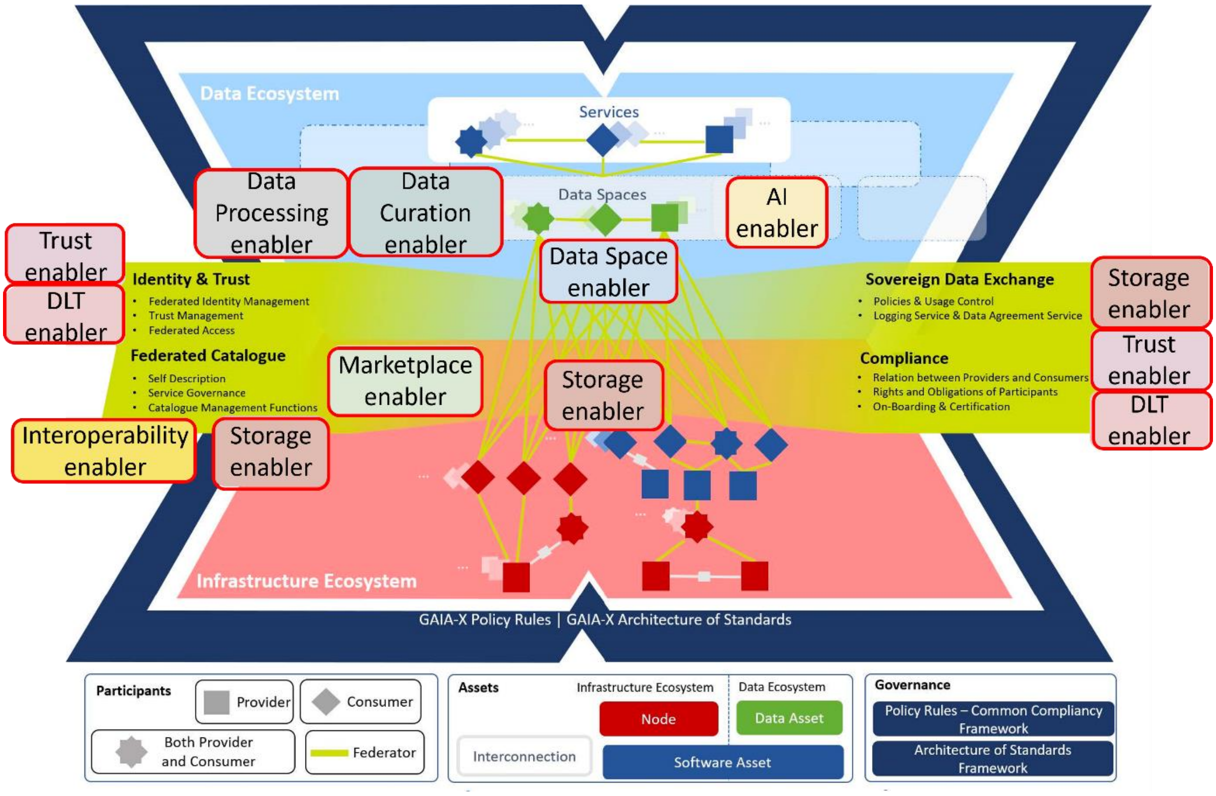


Figure 34: Gaia-X architecture view from Gaia-X Architecture document [34]

Figure 34 shows the relationship between SEDIMARK functional entities and Gaia-X architecture. It is worth highlighting that several of the components that are interrelated in the figure for the GAIA-X architecture are not present, such as the Gaia-X Portal, which is mirrored in SEDIMARK through the Marketplace Enabler functional entity. Other functional entities, such as the Data Space enabler, Data Processing and AI enablers are not part of the Gaia-X architecture, although they are crucial for the marketplace in SEDIMARK. Finally, several components such as the Storage enabler provides some of the functionalities envisioned in Gaia-X (e.g. transaction logging, data storage provision or the Federated Catalogue).

6.13.3 SEDIMARK relation to BDVA

The Big Data Value Association (BDVA) [35] is a public-private partnership representing the European Big Data Value ecosystem. The goal of BDVA is to boost data-driven digital transformation across industries in Europe. SEDIMARK's efforts of improving data quality and interoperability at the edge, implementing distributed AI, and creating a marketplace for efficient data/service discovery ties closely with BDVA's objectives.

The Reference Model of BDVA is depicted in Figure 35. The BDVA Reference Model aims to address concerns and aspects of Big Data value systems, outlining features that are the core of BDVA, while also including aspects that are aligned with other EU activities. The Reference Model is split into horizontal and vertical concerns, where the horizontal ones cover specific aspects related with a specific functional area, while the vertical ones are cross-cutting activities spanning in multiple or all areas and can also be non-technical aspects.

Figure 35 shows also the mapping of SEDIMARK's Functional Enablers to the horizontal and vertical aspects of the BDVA Reference Model. It is evident that SEDIMARK addresses all concerns of BDVA, with a high overlap on the horizontal aspects, since SEDIMARK is a data-driven framework that deals with data curation, processing, data analytics, AI, Interoperability and data sharing.

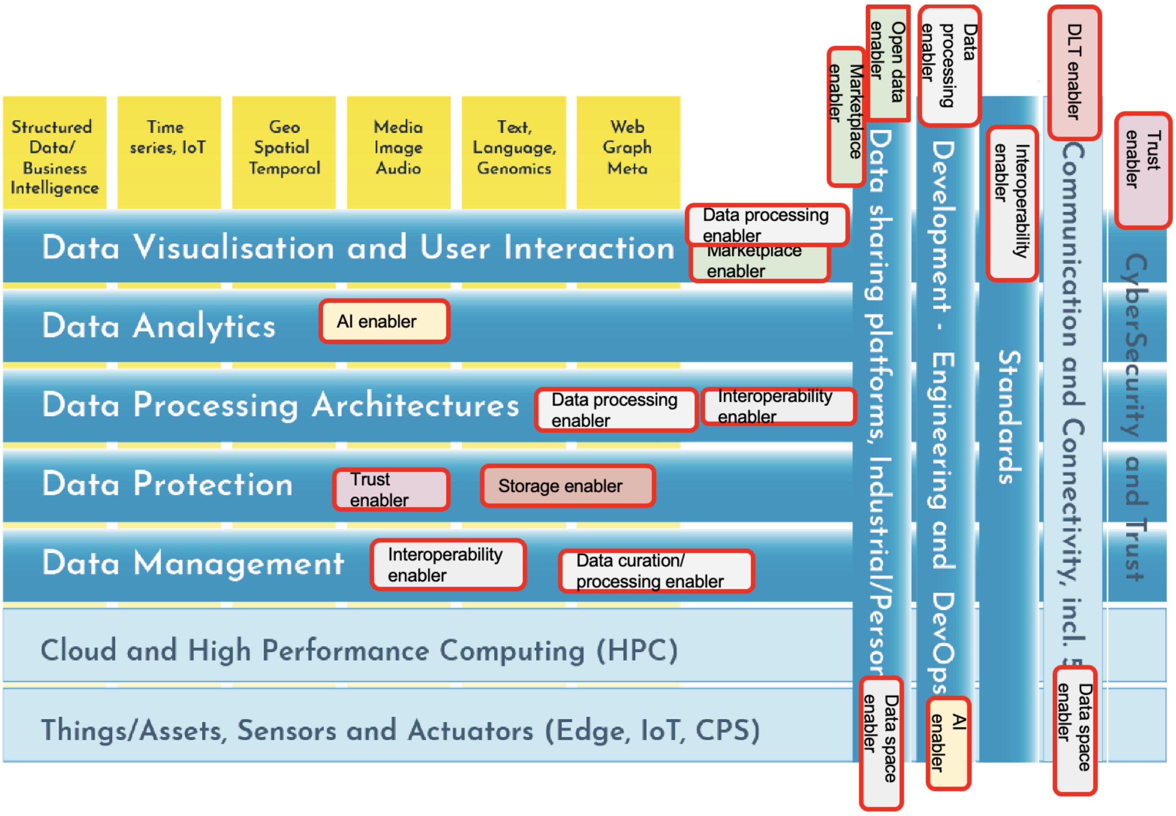


Figure 35: Mapping SEDIMARK Functional Enablers to the BDVA reference architecture

6.13.4 SEDIMARK relation to OpenDEI

OpenDEI [36] is an EU Horizon project that deals with creating common data platforms on a unified architecture, implementing the Digital Transformation strategy of the European Union. In doing so, OpenDEI targets to align reference architectures and open platforms for digitising the European Industry. OpenDEI has published its Reference Architecture [37] (see Figure 36) which promotes reusability and interoperability, framed around reusable Model Building Blocks (MBBs). SEDIMARK fits well with the approach followed by OpenDEI, with its Functional Enablers being aligned to the MBBs of OpenDEI. This mapping between the FEs and the MBBs can be seen also in Figure 36. As it can be seen, SEDIMARK has Functional Enablers covering one or more OpenDEI MBBs, apart from the Human Systems MBB, because SEDIMARK does not deal with human interactions with IoT systems.

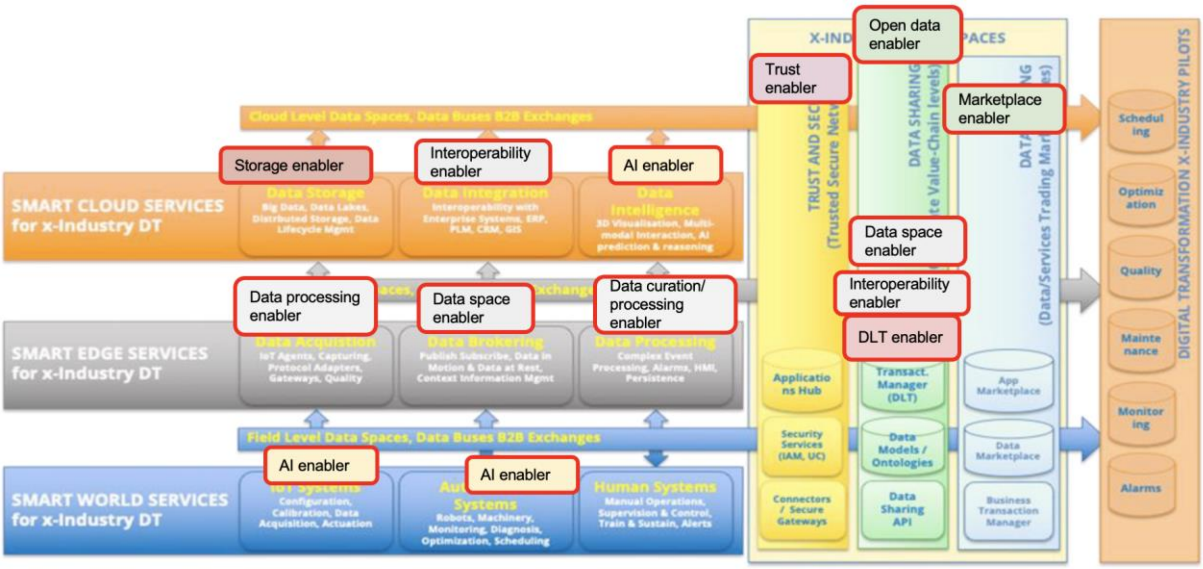

X-IND
PACES
Open data enabler
Trust enabler
Cloud Level Data Spaces, Data Buses B2B Exchanges
Storage enabler
Interoperability enabler
AI enabler
Marketplace enabler
SMART CLOUD SERVICES for x-Industry DT
Data Storage
Big Data, Data Lakes, Distrbuted Storage, Data Lifecycle Mgmt
Data Integration
Interoperability with Enterprise Systems, ERP, PLM, CRM, GIS
Data Intelligence
3D Visualisation, Multi-modal Interaction, AI prediction & reasoning
TRUST AND SEC
(Trusted Secure Netw
DATA SHARING
te Value-Chain levels)
DATA
NG
(Data/Services Trading Mar
es)
Data space enabler
Data processing enabler
Data space enabler
Data curation/ processing enabler
Interoperability enabler
DLT enabler
SMART EDGE SERVICES for x-Industry DT
Data Acquistion
IoT Agents, Capturing, Protocol Adapters, Gateways, Quality
Data Brokering
Publish Subscribe, Data in Motion & Data at Rest, Context Information Mgmt
Data Processing
Complex Event Processing, Alarms, HMI, Persistence
Applicatio ns Hub
Transact. Manager (DLT)
App Marketplace
Field Level Data Spaces, Data Buses B2B Exchanges
Security Services (IAM, UC)
Data Models / Ontologies
Data Marketplace
AI enabler
AI enabler
SMART WORLD SERVICES for x-Industry DT
IoT Systems
Configuration, Calibration, Data Acquisition, Actuation
Au Systems
Robots, Machinery, Monitoring, Diagnosis, Optimization, Scheduling
Human Systems
Manual Operations, Supervision & Control, Train & Sustain, Alerts
Connectors / Secure Gateways
Data Sharing API
Business Transaction Manager
Schedul ing
Optimiz ation
Quality
Mainte nance
Monitor ing
Alarms
DIGITAL TRANSFORMATION X-INDUSTRY PILOTS

Figure 36: Mapping SEDIMARK Functional Enablers to the OpenDEI architecture

6.13.5 SEDIMARK relation to DSSC

The Data Spaces Support Centre (DSSC) [15] is an initiative funded by the European Commission as part of the Digital Europe Program and aims to assist companies and the public sector to build interoperable and interacting Data Spaces. SEDIMARK is participating in its activities and contributing to the building of the concepts and the architecture of DSSC towards creating common data spaces within the EU.

As it has been described in Section 3, the DSSC has already circulated some quite stable sketches of its functional architecture, which is organized in three pillars, and defines a set of, so-called, Building Blocks, per each of these pillars.

As it is shown in Figure 37, the three pillars in the DSSC's architecture are Interoperability, Trust and Data value.

In terms of Interoperability, the DSSC is looking for supporting the complete decoupling of data providers and consumers, making data services FAIR (Findable, Accessible, Interoperable and Reusable). This requires the adoption of interoperable APIs for data exchange and the definition of compatible data models. Interoperable mechanisms for the traceability of data exchange transactions and data provenance are also needed.

In terms of Trust, DSSC is proposing that data spaces should bring technical means for guaranteeing that participants in a data space can trust each other and exercise sovereignty over the data they share based on user-controlled consent. This requires the adoption of interoperable standards for managing the identity of participants, verifying their trustworthiness, and enforcing policies agreed upon for data access and usage control.

Finally, in terms of Data value, the DSSC is promoting the generation of value from sharing data. Therefore, data spaces can contain multi-sided markets to trade, buy, and sell data services. This requires, not only the adoption of interoperable mechanisms enabling the description of the terms and conditions for accessing to the available data service offerings, but also the publication and discovery of such offerings, and the management of all the necessary steps supporting the lifecycle of contracts that are established when a given participant acquires the rights to access and use data.

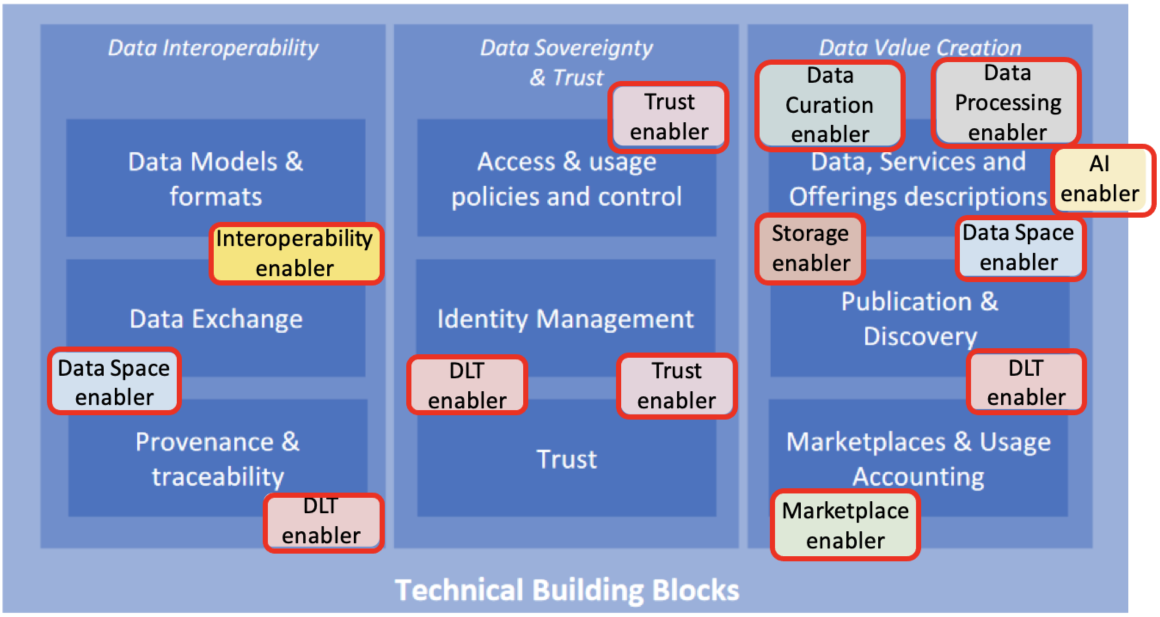


Figure 37: Initial mapping of SEDIMARK Functional Enablers to the DSSC Building Blocks

As it can be seen in Figure 37, the already identified SEDIMARK's Functional Enablers cover the three pillars in the DSSC's functional model and provide support to the respective Building Blocks in each of these pillars.

Most interestingly, the DLT enabler, which is, among other aspects at the root of the decentralized nature of the SEDIMARK's marketplace, spans its influence over the three pillars.

Moreover, data enrichment and quality enhancement aspects, which are crucial in the SEDIMARK model, are fostered through the development of several enablers that are meant to go beyond the FAIRness of data is referred to as a towards data quality and further on to creating data value chains.

Last, but not least, the Data Space enabler supporting most of the data lifecycle in the Marketplace shows the alignment between the SEDIMARK functional model and the data space ecosystem that DSSC is trying to create.

In any case, the DSSC aims to publish the first public version of its architectural building blocks in September 2023. Thus, a thorough mapping of the SEDIMARK's Functional Entities to the DSSC Building Blocks will be presented in the next version of this Deliverable, D2.3 due in Month 24 (i.e. September 2024).

7 Examples of data flows

This section presents a set of examples to show the different data flows between functional entities of the architecture described in Section 6. In particular, the data flows represent the following cases:

- The first three data flows are related to the whole Offering lifecycle, which includes the Offering publishing (i.e. create and make available an Offering within the Marketplace), Offering discovery (i.e. enable the discoverability of Offerings and how to discover them), and Offering consumption (i.e. that includes the purchase and data access process).
- The following two data flows are related to the Identity and Access Management within the functional architecture, and include the identity generation and registration to the Marketplace, which describes the onboarding process to obtain an account in a Marketplace; and the identity verification flow, to validate all the interactions carried out in the Marketplace.
- Finally, there are three data flows related to data processing and AI management, including:
    - data processing to curate, augment, visualize and enrich data before it is published into the Marketplace;
    - a recommender system data flow based on personal preferences for a consumer; and
    - the distributed learning process, to train AI models collaboratively between participants in the Marketplace, including Federated and Gossip Learning strategies.

7.1 Offering Publishing data flow

The "Offering Publishing" data flow presents the interactions and processes involved when a participant provider publishes a new offering within the marketplace. This is shown in Figure 38.

In this data flow there are three main functional entities involved, including the Data Space Enabler, from the provider side; the DLT enabler, that will support the decentralized infrastructure to guarantee the trustworthiness, provenance and traceability of the registered offering; and the Interoperability enabler, to provide the means to ensure the offerings follow the Marketplace Information Model. The main actor in this scenario is the provider, and no interactions will be carried out from the consumer side.

In this data flow, the provider creates an offering from existing assets (or the ones resulting from some additional processes, like the ones provided by the Data Processing enabler). Once it is created, a new Verifiable Credential signed by an Issuer is created, and later on stored locally. Finally, it is registered into the Registry, including the pointer to the location of the complete offering, and a hash to ensure the offering has not been tampered after its publication. Therefore, any modification of any existing offering must be registered again into the Marketplace.

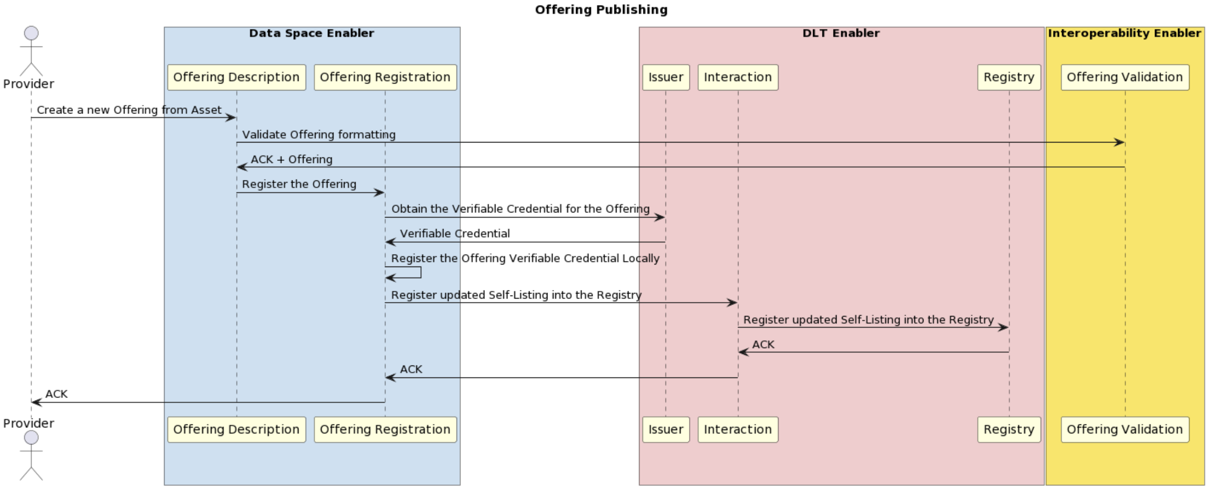

Offering Publishing
Data Space Enabler
DLT Enabler
Interoperability Enabler
Provider
Offering Description
Offering Registration
Issuer
Interaction
Registry
Offering Validation
Create a new Offering from Asset
Validate Offering formatting
ACK + Offering
Register the Offering
Obtain the Verifiable Credential for the Offering
Verifiable Credential
Register the Offering Verifiable Credential Locally
Register updated Self-Listing into the Registry
Register updated Self-Listing into the Registry
ACK
ACK
ACK

Figure 38: Offering Publishing data flow example

7.2 Offering Discovery data flow

The "Offering Discovery" data flow, depicted in Figure 39, plays a pivotal role in the secure decentralized SEDIMARK Marketplace architecture, enabling consumers to search for and discover offerings that align with their requirements. This data flow involves interactions and processes initiated by a Consumer, that search throughout the catalogue for offerings.

This data flow emphasizes the critical role of the Distributed Catalogue in maintaining an accurate and up-to-date inventory of data offerings. It ensures that Consumers can efficiently discover and access the data they require. In this sense, the Distributed Catalogue is in charge of querying and indexing the list of offerings from the Registry. While now the workflow is based on querying operations, in a future it is foreseen the use of a subscription-based mechanism to maintain the catalogue.

It is important to highlight that the Offering (in the form of Verifiable Credential) will not be fully published into the Registry, thus the Distributed Catalogue will have to query the offering provider to obtain all the details. In this sense, identification and authentication flows are not included in this data flow, and can be found in the following subsections.

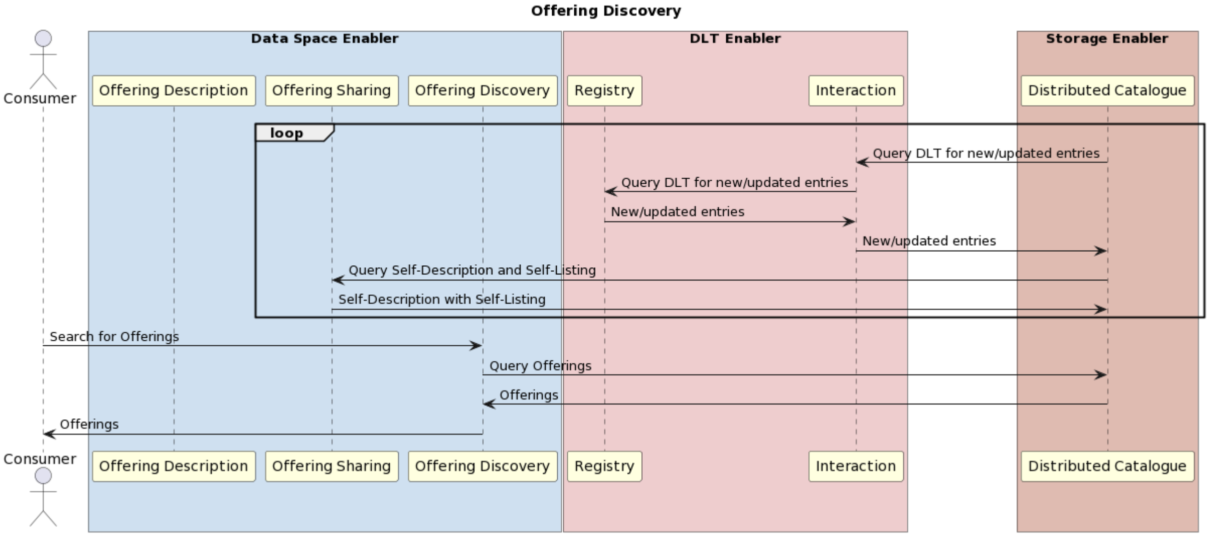


Figure 39: Offering Discovery data flow example

7.3 Offering Consumption data flow

The "Offering Consumption" data flow encapsulates the interactions and processes that occur when a consumer consumes an offering published by a provider. In this data flow, as shown in Figure 40, the consumer always initiates the interaction, and can be divided in two main phases: the contract signing and the asset exchange.

During the first phase, that is only needed once, the consumer requests the details of the offering, thus the details of the contract established by the provider can be obtained. This process is of utmost importance as such contract must be signed by both participants, and establishes the access and usage control rules under which the assets that belongs to the offering can be used. To sign the contract and come to an agreement, the platform leverages the use of Smart Contracts from the DLT, that can be queried by both participants to ensure the details are still met.

The second phase involves the asset exchange, and can follow several communication patterns. The first one follows a request-reply pattern, in which the offering assets are requested by the consumer. The second one follows a publish-subscribe pattern, in which the consumer is notified for any update or change in an asset (e.g. data streaming). Therefore, this phase can last as long as the agreement is valid.

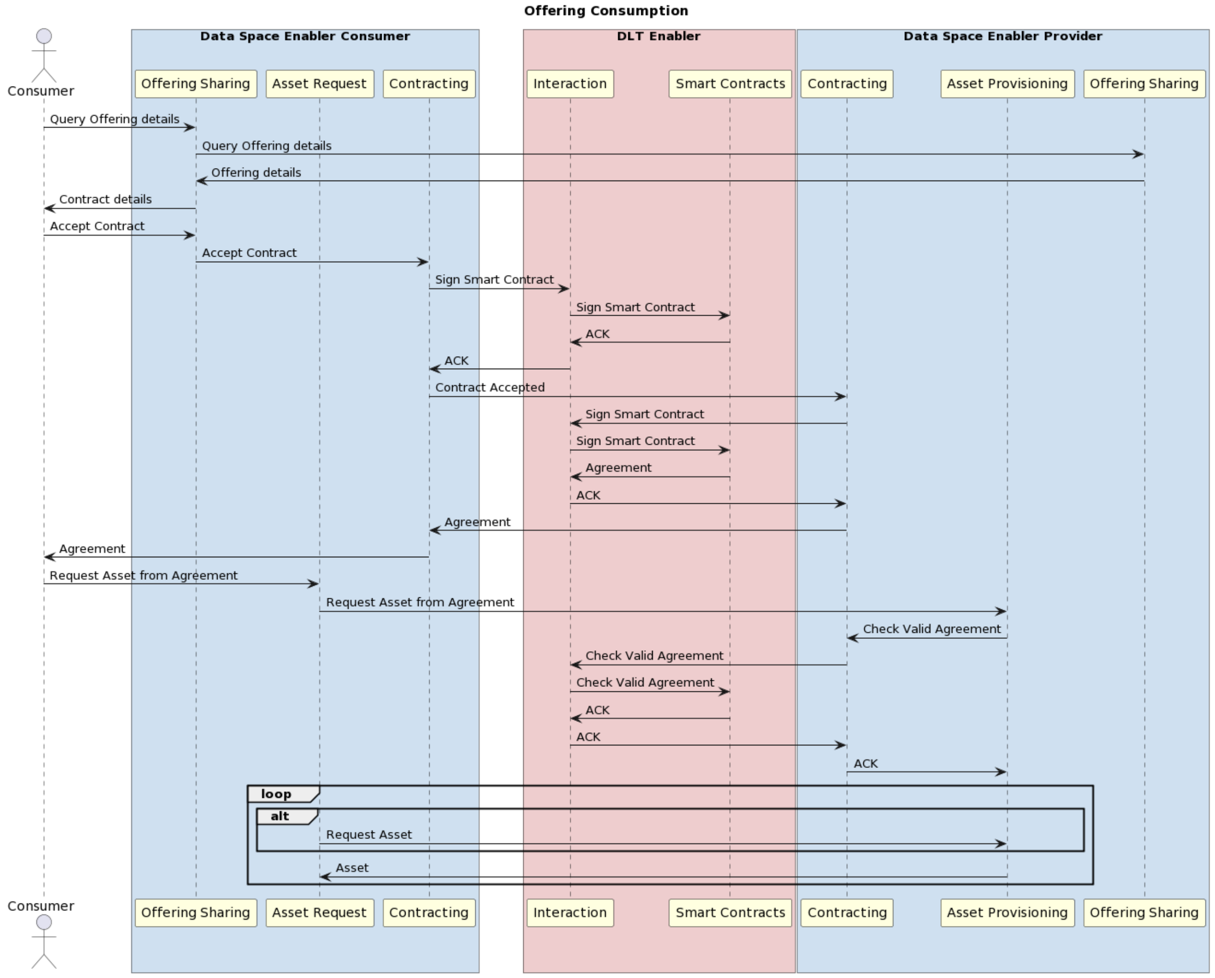


Figure 40: Offering Consumption data flow example

7.4 Identity Generation and Registration to the marketplace data flow

The UML Sequence Diagram, shown in Figure 41, represent the flow for the creation of a digital identity to be employed in the SEDIMARK Marketplace.

A generic Connector (which can be any actor) creates its own DID and DID document and then publish it onto the Distributed Ledger. Then, it asks to the Issuer for the SEDIMARK Marketplace to issue a VC. The Issuer generates and returns the VC. All the interactions with the DLT are mediated by the Interaction functional block. Eventually, the User's Connector that requested the VC securely stores the VC locally.

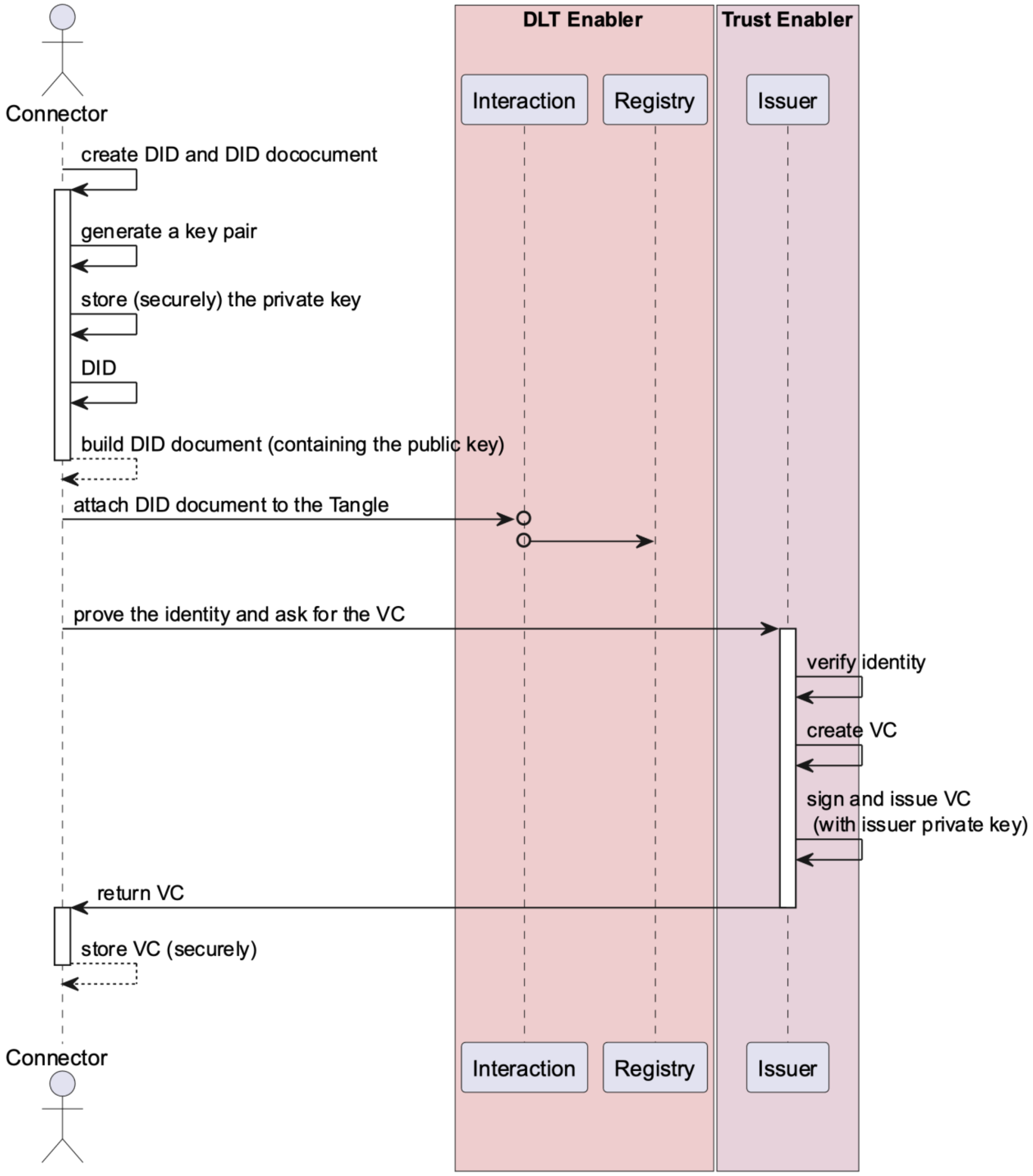

DLT Enabler
Trust Enabler
Connector
Interaction
Registry
Issuer
create DID and DID dococument
generate a key pair
store (securely) the private key
DID
build DID document (containing the public key)
attach DID document to the Tangle
prove the identity and ask for the VC
verify identity
create VC
sign and issue VC
(with issuer private key)
return VC
store VC (securely)
Connector
Interaction
Registry
Issuer

Figure 41: Identity generation and registration to the marketplace data flow example

7.5 Identity verification data flow

The UML Sequence Diagram, shown in Figure 42, represent the flow for the verification of an identity of an actor in the SEDIMARK Marketplace.

A generic Connector (which can be any actor) uses its own (previously created) digital identity to generate a VP and presents it to the Verifier where it is requesting a service. The Verifier performs all the necessary security and cryptographic verifications (Trust Enabler) on the VP, also interacting with the Registry (DLT Enabler). Depending on the results, the Verifier decides to grant (or deny access) to the service.

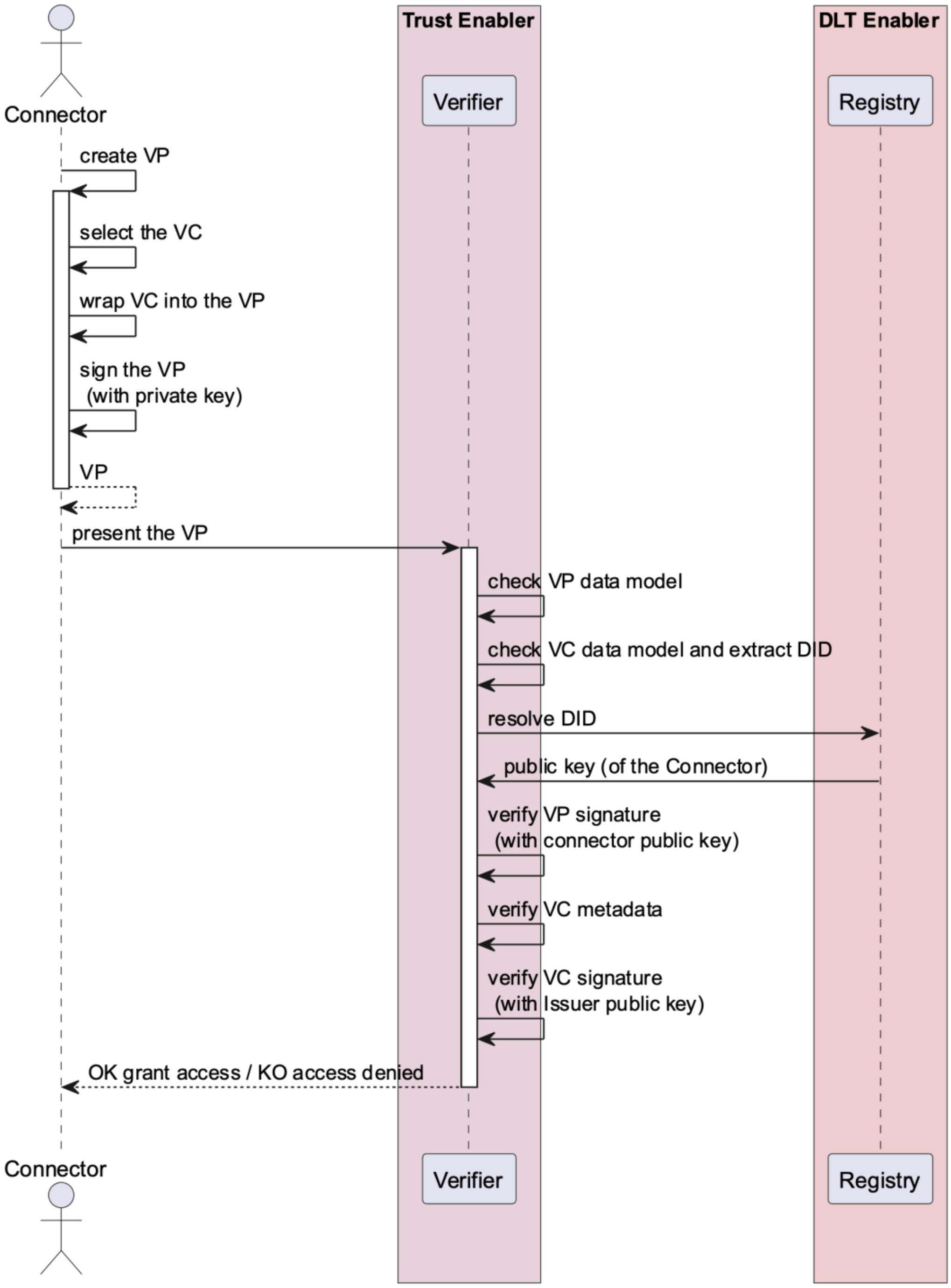


Figure 42: Identity verification data flow example

7.6 Open data discovery

As described in section 6.12, SEDIMARK aims at providing access to some open data portals to all of its participants. To achieve this, the Open Data enabler itself will be a participant in the ecosystem, acting as a provider whose offerings exposes the APIs of selected open data portals such as Kaggle [38] or CKAN [39]. Therefore, the workflow to access such open data from a consumer perspective is identical to the one depicted in Figure 40, where the offering provider is the open data enabler.

7.7 Data processing data flows

An example of the message sequence for processing and curation a dataset at the provider's side is given in Figure 43. The process assumes that the provider interacts with the Data Processing Dashboard, which is the main component that allows any user to interact with the data processing pipeline. The provider then selects the configuration of the processing request i.e. which dataset to process/curate, what type of processing/curation will be performed, what modules will be used and the configuration for each module. The Data Processing Dashboard then forwards all this information to the Data Processing Orchestrator, which is the module responsible for managing the data processing pipeline and invoking the respective processing and curation services. The Data Processing Orchestrator receives the dataset from the data source and forwards it to the data adapter, so that the dataset is converted to the SEDIMARK internal format for being processed and then forwards the formatted dataset to the Data Profiling module for extracting the initial statistics about the dataset. Then, the dataset is forwarded to the rest of the curation modules (Deduplication, Outlier Detection, Missing Value Imputation) and then forwarded to the Data Quality Evaluation module for computing the final quality statistics about the dataset. After the dataset being curated, it can be forwarded to the Feature Engineering module to extract features and then to the Data Annotation to be converted to the external SEDIMARK information format for interoperability. After that, the dataset is validated through the Data Validation module and presented to the provider through the Data Visualisation module, where the provider can see details about the statistics of the dataset and the various processing steps that have been used. At this point, the dataset is ready to be registered as an asset to the marketplace.

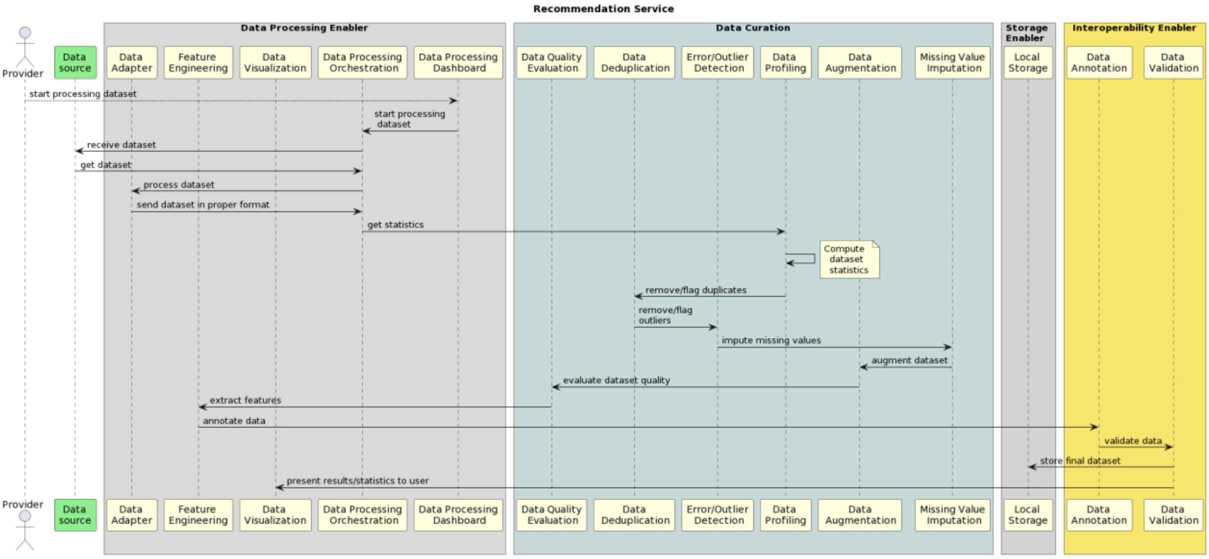


Figure 43: Data processing data flow example

7.8 Recommendation service data flows

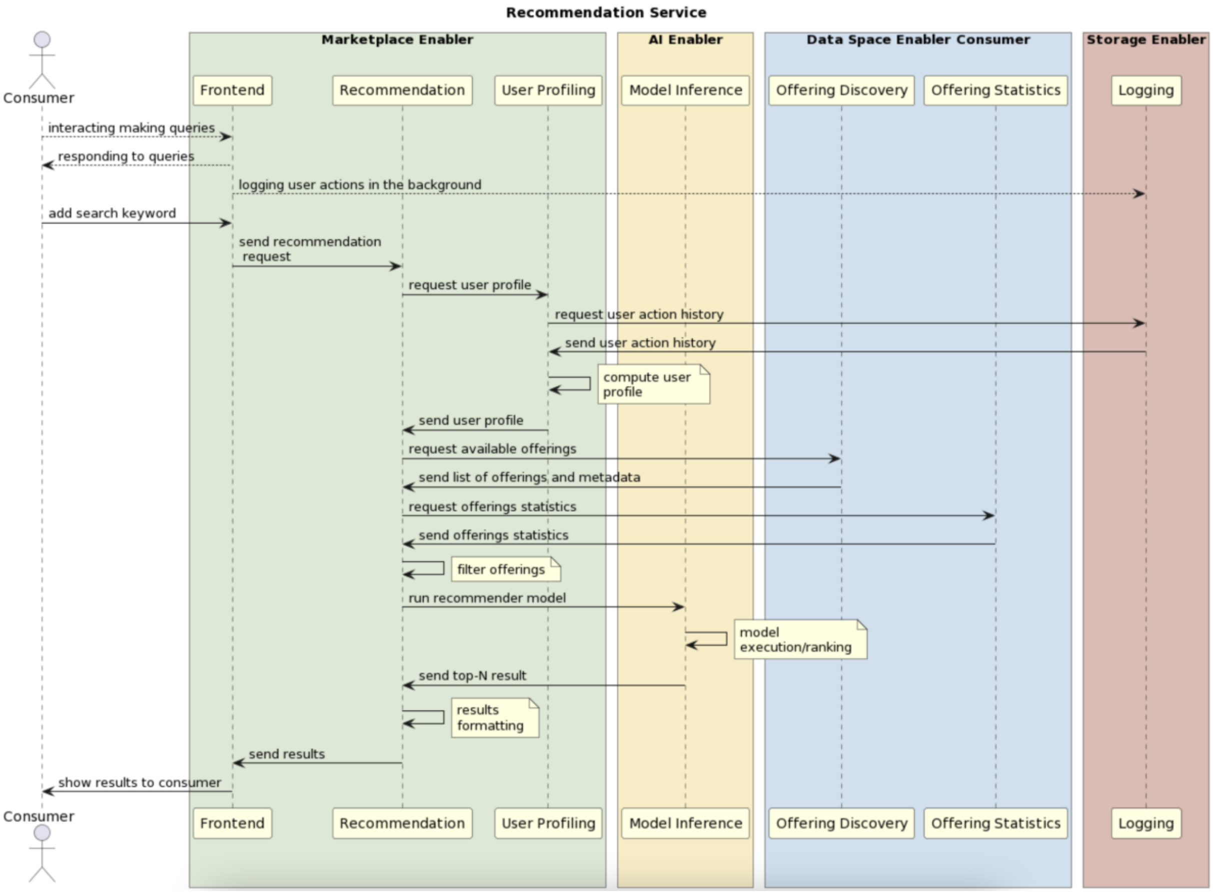


Figure 44: Recommendation service data flow example

In this section we present an example of the data flow for serving recommendations for assets to users of the SEDIMARK platform. The recommendations can be served either due to direct user request or as a side process when users perform discovery queries for assets/offerings. The example data flow is presented in Figure 44. The process starts with the consumer interacting with the Marketplace Frontend performing queries about discovering offerings. The Frontend has a background process that logs user actions locally (with the user's consent) using the Logging module. This is used in order to be able to track the activity history of the user and construct a user profile to be able to identify the users preferences for assets/offerings. The Frontend forwards user query metadata to the Recommendation module that is responsible for providing the recommendations to the user. The Recommendation module responds each time a user makes a new query, presenting personalised results about offerings that might be of interest for the consumer. To do this, the Recommendation module forwards requests to the User Profiling module for getting the user profile and preferences. To do so, User Profiling interacts with Logging to get the user history log for past queries, purchases, likes, etc. and then computes the profile and sends it to the Recommendation module. In parallel, The Recommendation module interacts with the Offering Discovery and the Offering Statistics modules in order to receive the list of available offerings and their related metadata including statistics (i.e. trending, most purchased, highest rated, etc.). After that, the Recommendation module does a first filtering for removing completely unrelated offering, reducing the possible options and forwards all the details to the Model Inference module, which runs the Recommender Model in order to extract the top-N results for the offerings to be recommended to the consumer (where top-N is the `N` offerings with the highest recommendation score). Then, the Recommendation module converts the results into a proper format for display and forwards them to the Frontend to be presented to the consumer.

7.9 Distributed machine learning training data flows

In this section we present two examples of data flows for distributed training of machine learning models, based on the two main categories of distributed training considered within SEDIMARK: (i) Federated Learning and (ii) Gossip Learning. These are presented in the next two subsections.

7.9.1 Federated learning

One example of Federated Learning is the process shown in Figure 45. In this process, we assume that the Provider plays the role of the Server in the Federated Learning and has some dataset on which they have trained a machine learning model and they want to improve it using the knowledge of similar datasets of other providers/users. Thus, the Provider/Server interacts with the AI Orchestrator submitting a command for starting a Federated Learning training, specifying all the required configuration regarding the model to be trained, the dataset to be used, the policies for the process, etc. The AI Orchestrator interacts with the Distributed Model Training module to start the process. The Distributed Model Training module (DMT) then sends the model details to the Local Model Training (LMT) module to load or initialise the model and start the local training. The LMT then sends back the model weights to the DMT, which forwards them to the AI Orchestrator in order to register the Federated Training process as a new SEDIMARK Offering getting the Offering description from the respective module (not depicted for simplicity) and sending the details to the Offering Registration module for registering the offering to the Registry and update the entries to the Catalogue.

Then, at the Client side (which can be another provider that has a similar dataset), the Client interacts with their Offering Discovery module (through their Frontend – not depicted for simplicity) to discover offerings. The Offering discovery module interacts with the Catalogue to get the list of Offerings and presents them to the Client, who selects the Federated Learning offering and interacts with the Offering Sharing module to get this Offering (the contracting and agreement processes are not depicted and the sharing process is simplified for simplicity). The Client's Offering Sharing module interacts with the Server's Offering Sharing module to download the offering details, i.e. training process, model weights, dataset description, etc. Then, the Client's Offering Sharing is forwarding these details to the Client's AI Orchestrator to manage the local process. The Client's AI Orchestrator forwards the details to the Client's DMT to start the local Federated Learning process. The Client's DMT forwards the model details to the local LMT to build and initialise the model. The Client's DMT interacts directly with the Server's DMT to register the Client as a new client to the Federated Learning process and gets back the latest model weights, which are then forwarded to the local (Client's) LMT to start a new round of local training. Then, the rest of the process is in a loop of local model training in rounds, sending the Client's model weights to the Server, which aggregates the weights received from all clients and its own weights and sending back to the Clients the updated weights for the new round.

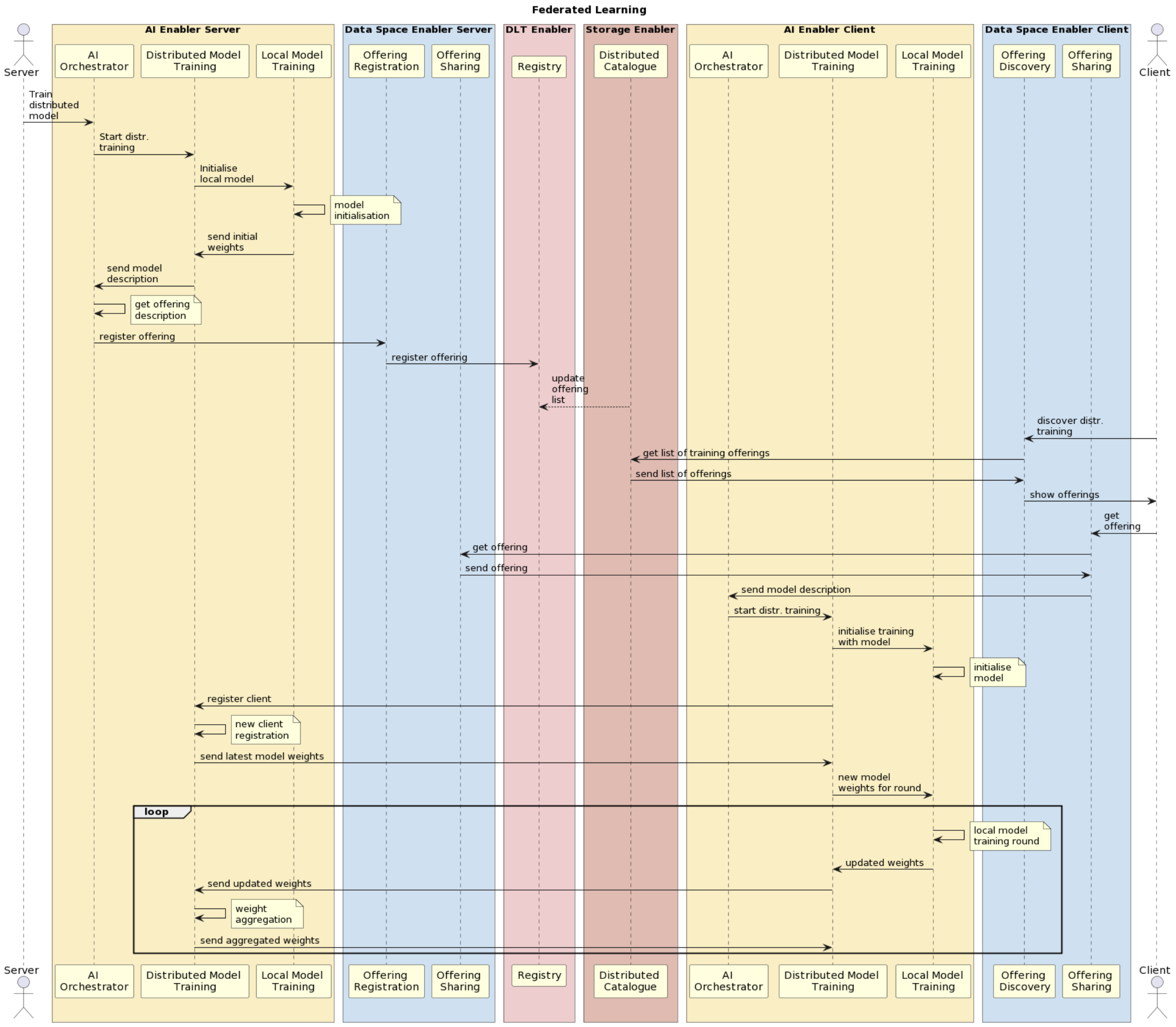


Figure 45: Federated learning data flow example

7.9.2 Gossip learning

The example of the data flow for Gossip Learning within SEDIMARK is depicted in Figure 46. This process is different compared to the Federated Learning one due to the fact that Gossip Learning is a completely decentralised process without the need for a server, allowing participants to send the updated weights directly to each other. It can be considered as each participant having both the roles of a Server (doing weight aggregation) and Client (doing local training and sending weights to other “servers”). However, considering that the process will need to start in some way, we assume that one participant will be the “Initiator” of the process and will be the provider that has a dataset and wants to train a decentralised model on it. The process starts in a similar way as in the Federated Learning data flow, with the exception that the Initiator also initialises the Network Graph (a graph that shows which are the participants in the Gossip Learning process) through their Distributed Model Training Module and stores this to a Distributed Storage area setting the respective policies so that only the participants in the Gossip Learning process will be allowed to access the Network Graph. Then, the Initiator continues to build the local model and register the process as a new offering in the Registry/Catalogue.

At the Participant's side, the discovery process is similar to the Federated Learning one, getting the offering details from the Initiator and starting the local model. However, here there is no process to register as a client to a server, but instead the Participant accesses the distributed storage to get the network graph and update it using their details to add themselves as another participant in the process, so that other participants can send/receive weights from them. Then, each round of Gossip Learning includes the local training of the model, the update of the network graph, the selection of target peers to send the weights, the reception of weights from other peers and the aggregation of all weights in order to start the next round of training.

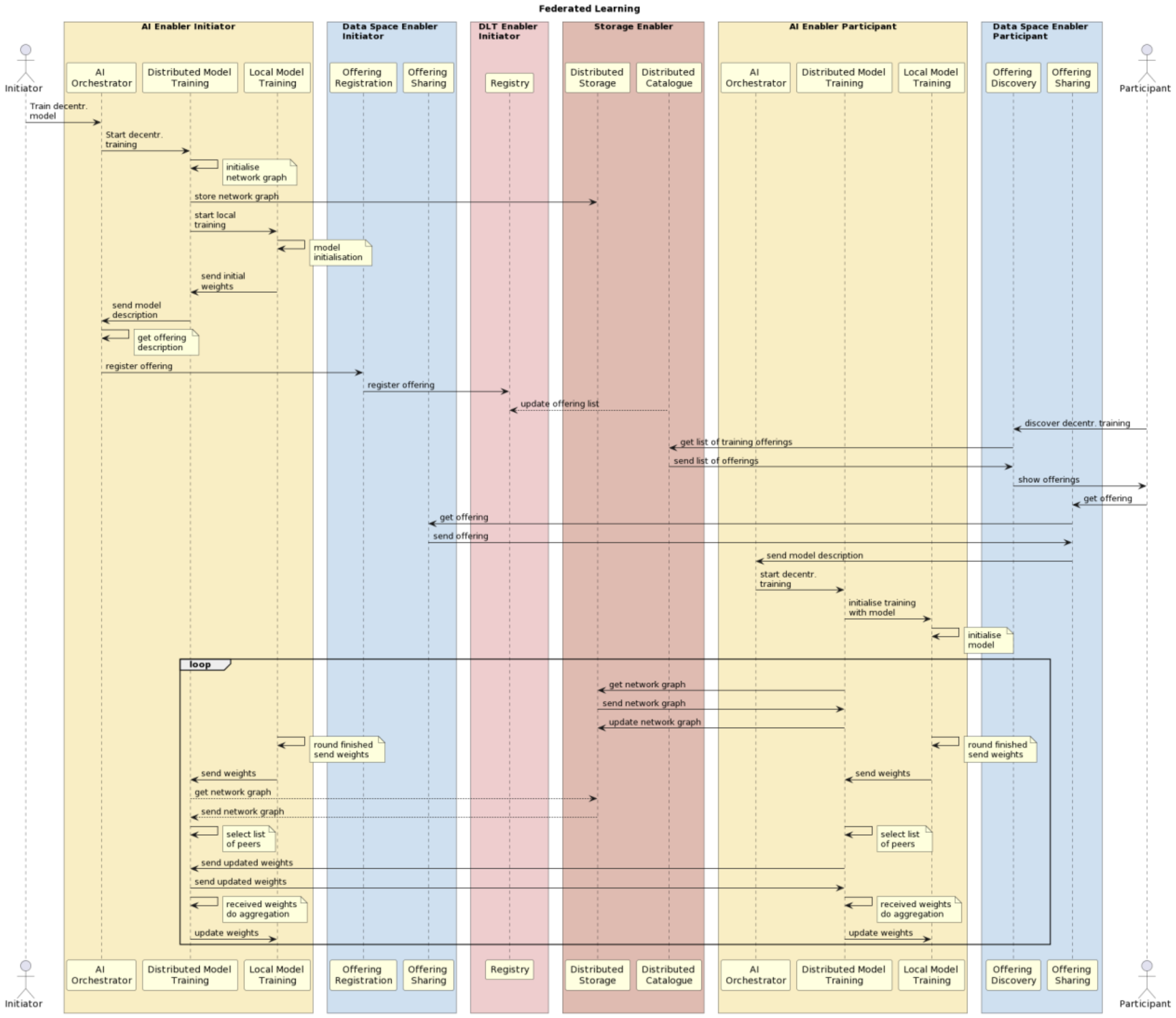


Figure 46: Gossip learning data flow example

8 Initial Interface description

The previous chapter gave a detailed description of SEDIMARK platform enablers, i.e. groups of components over which all functionalities of SEDIMARK are partitioned. With more than 10 of such enablers, SEDIMARK's architecture encompasses a wide variety of microservices, therefore requiring a significant number of interfaces to ensure well-defined, scalable and robust communication between them. Figure 47 aims at providing an overview of how such interfaces can be classified in three groups: internal, external and graphical user interfaces.

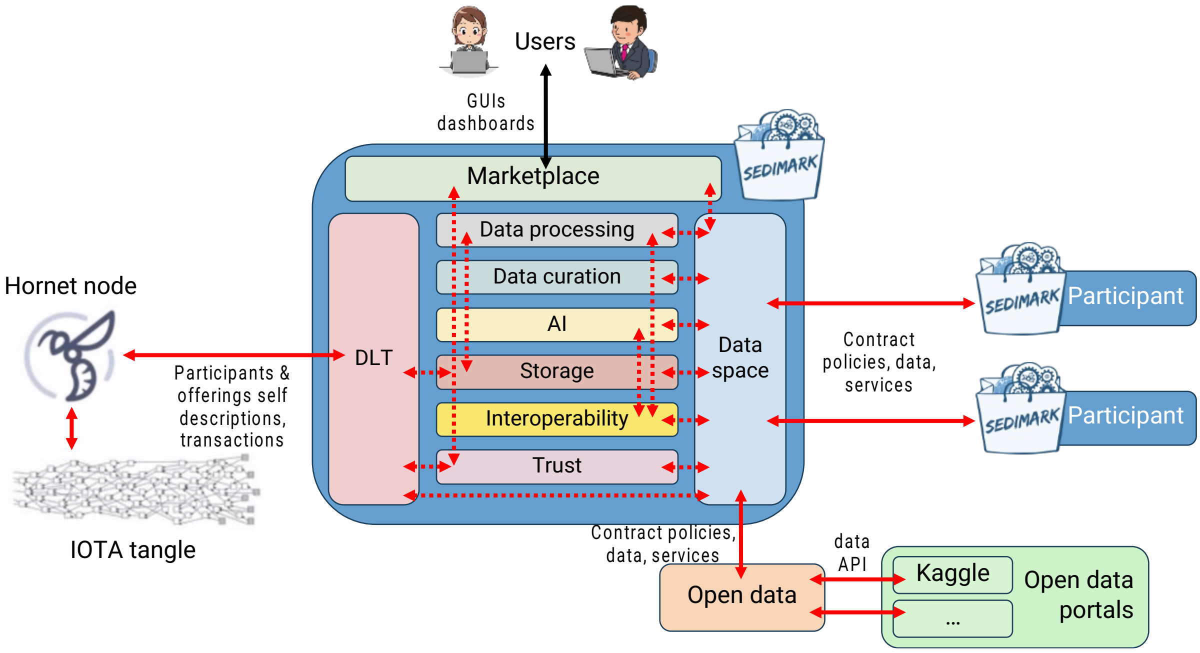


Figure 47: Overview of SEDIMARK platform interfaces

8.1 Internal interfaces

Represented by dashed red lines, they connect internal components of the platform. Within a given enabler, a microservice can communicate with another one from the same enabler, but also interface with a service from another enabler. For example, as shown in Figure 38, the registration of an offering is ensured by the self-description and registration modules within the data space enabler, yet these modules also communicate with the interoperability enabler to validate its description, then the trust enabler to get credentials associated with it, and finally with the DLT enabler to register it to the IOTA tangle.

Because of the vast varieties of components they connect, several technologies may be used to implement these interfaces. While Representational State Transfer (REST) APIs will cover most of them, other technologies may be used in specific cases. For instance, a federated catalogue implementation from Gaia-X uses GraphQL [40]. Remote Procedure Call (RPC) may also be used for orchestrators (such as the data processing manager or offering lifecycle management), as they are more suited for command and action-oriented APIs.

8.2 External interfaces

Indicated in red solid arrows in Figure 47, they enable SEDIMARK platform to exchange information with external resources. They consist of three distinct channels:

- **Between SEDIMARK ecosystem participants**: transactions between participants imply exchanging many data between them, among which the contract agreement, the data usage policies and the data or service offered itself. These exchanges are ensured by the connector component within the data space enabler. Any technology can be used here, depending on the transaction's offering, provided that there is a respective extension in the connector to support that technology.
- **Between participants and open data portals**: as described in section 6.12, the open data enabler is a participant within the SEDIMARK ecosystem, responsible for translating open data portals into offerings directly accessible in the catalogue. Consequently, the open data enabler comprises two set of interfaces: on the one hand, it fetches or exposes data from open portals, while on the other hand it provides such data or access to it via an offering. The latter case is a no more than a particular instance of a transaction between two participants, free of charge, and where the contract policies are indexed on those of the open data portals. For its first version, the Open data enabler will focus on exposing data freely accessible via public REST APIs.
- **Between participants and the IOTA tangle**: SEDIMARK relies on the IOTA tangle to provide a trustworthy, tamper-proof registry of the identity of its participants, offerings and contracts. The DLT enabler sustains this channel to write and retrieve such data from the tangle, namely the self-descriptions of the participants and their offerings, as well as the smart contracts representing the transactions between offerings providers and consumers.

8.3 Graphical user interfaces

SEDIMARK aims at providing a set of user-friendly GUIs to facilitate the creation of offerings and contracts, and therefore foster transactions in its ecosystem. To achieve this, frontends will be implemented for:

- **Data processing orchestrator**: to set data processing pipelines and monitor their status.
- **AI orchestrator**: to create and manage pipelines to train AI models.
- **The marketplace**: the central web app in the platform, where participants can manage their offerings and negotiate their contracts. More precisely, it contains GUIs to:
    - Register new participants;
    - Register new offerings;
    - Browse the offerings catalogue;
    - Create/negotiate a contract to consume an offering;
    - Manage and monitor offerings: a dashboard for users to keep track of their provided/consumed offerings, get some statistics about their usage, and the status of their transactions;
    - Access other SEDIMARK specific orchestrators such as data processing or AI.

To implement such graphical interfaces, a typical web frontend toolset will be used: a React framework such as Nextjs [41] on top of JavaScript.

8.4 System view of the architecture

In the study of complex engineering and systems design, a pivotal perspective emerges when we shift our focus from individual functionalities to the holistic system they collectively create.

This section provides a briefly exploration of this approach within the context of functional architecture, which can also serve as a gateway into WP5.

Functional architecture is the backbone of any well-designed system, and understanding its system-wide view is essential for engineers, designers, and stakeholders alike. In this section, we delve into the interconnectedness of components, subsystems, and the overarching objectives they collectively achieve. In this sense, Figure 48 depicts the envisioned system view of the functional architecture that has been presented throughout this deliverable. It shows the functional view colour code to better identify the functionalities covered by each of the presented components. Still, as several functionalities are spread among different components, only the main ones are shown.

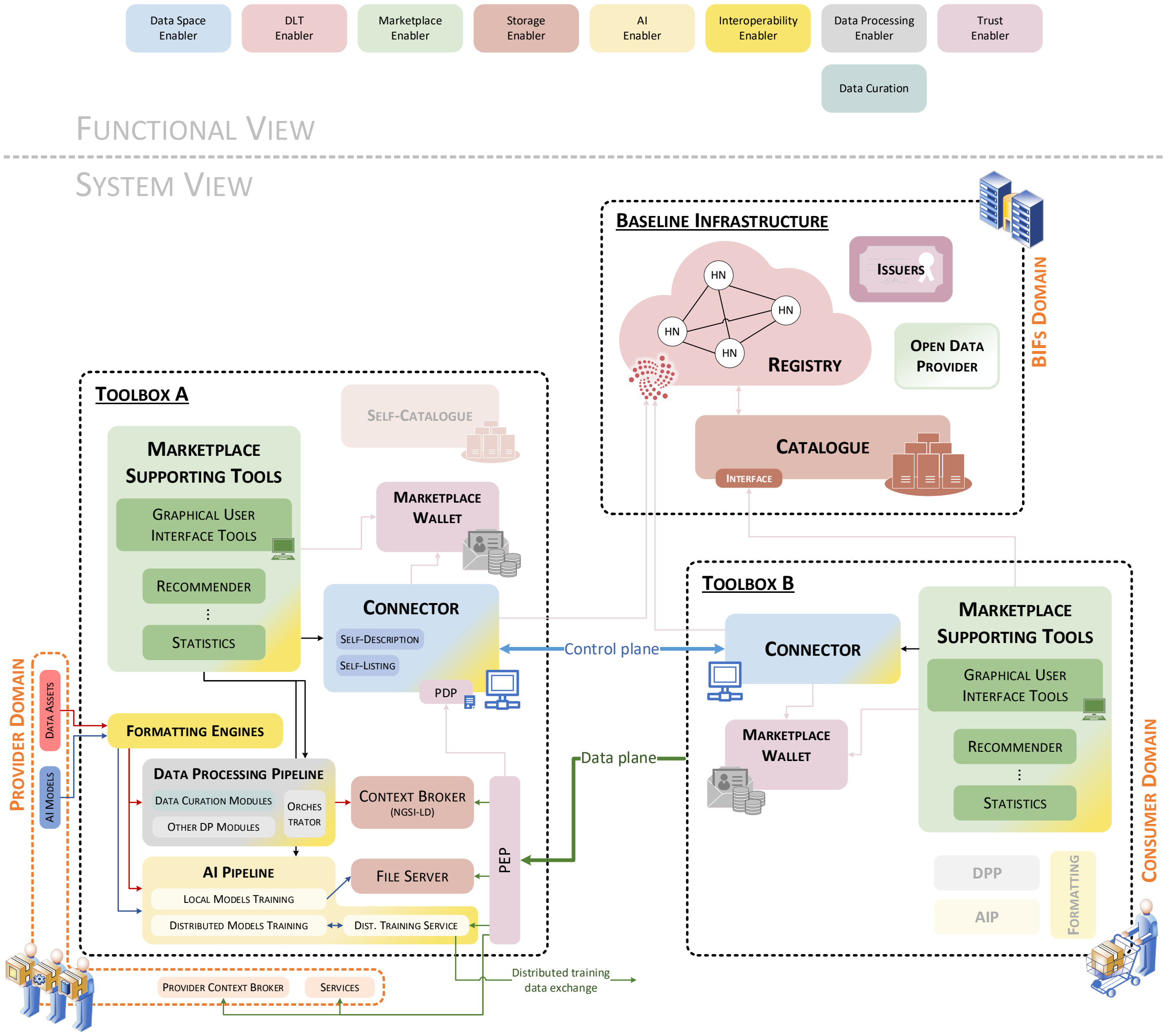


Figure 48: SEDIMARK platform system view

The system view identifies three differentiated domains: the Provider and Consumer ones and the Baseline Infrastructure Facilitators (BIFs) domains. In the SEDIMARK context, the first two are driven by the usage of a common toolbox composed of a set of software components which can be instantiated based on the needs. In contrast, BIFs domain provides all the infrastructure and systems needed to run the Marketplace. The following subsections present the different building blocks being deployed as part of the Baseline Infrastructure and the Toolbox.

8.5 Baseline Infrastructure

Baseline infrastructure includes the following components:

- The **Registry**, which is based on a DLT network with different nodes distributed across multiple stakeholders. This component is used as the basis for the trustworthiness throughout the rest of the marketplace, being both an integral part of the DLT enabler as well as playing a key role in the Trust enabler. Besides, it can also be considered part of the storage enabler and the data space enabler as it is essential in the distributed storage of the whole SEDIMARK offering set.
- **Issuers** are entrusted with the responsibility of validating and confirming the authenticity of individuals' claims and assertions. They act as authoritative sources, ensuring that the information presented in a self-sovereign identity system is accurate and reliable. This role is critical for preventing fraudulent activities and maintaining the integrity of the identity ecosystem. Therefore, Issuers are key part of the trust enabler.
- The **Open Data Provider** acts as any other SEDIMARK provider, exposing offerings in the marketplace to provide access to additional Open Data platforms. Still, in this specific case, the facilitator of this kind of open offerings is the SEDIMARK system itself, acting as intermediate to make such information available to be consumed through the marketplace. It is considered part of the marketplace enabler, although it behaves as a basic provider.
- As defined in Section 4, **Catalogues** are searchable (i.e. indexed) versions of the offering-related information referenced in the Registry, whose purpose is to facilitate Offerings' discoverability in a Marketplace. The baseline infrastructure provides one of these catalogues to serve as the initial contact point for consumers, but there can be multiple catalogues. Catalogues are considered both part of the data space enabler and the storage enabler.

8.6 SEDIMARK Toolbox

Toolboxes can be tailored to suit individual requirements and preferences, hence not all of its components are compulsory. On the one hand, this provides the needed flexibility in customizing the toolkit to meet specific needs. Still, certain components are designated as mandatory, ensuring essential features are always present. Toolboxes include the following components:

- The Marketplace wallet is a crucial component designed for the secure storage of essential credentials required to access and participate in a marketplace ecosystem. It serves as a digital safe for storing authentication tokens, keys, and other sensitive information, ensuring that users can seamlessly interact with the marketplace while maintaining the highest level of security. This component not only safeguards user data but also streamlines the user experience by providing a convenient and protected means of managing access credentials within the marketplace environment. This component is part of the trust enabler and it is mandatory.
- The Connector is responsible for enabling secured peer-to-peer information exchange between Participants. It is a mandatory component, being the central component of the data space enabler and handling most of the control plane interactions between peers. Its role is instrumental in fostering a trusted environment for participants to share data and services seamlessly and securely. Besides, by acting as a bridge for data exchange, the connector enhances communication and collaboration within the network while ensuring

that all information is transmitted and received with the highest levels of security and confidentiality. In this sense, it also takes part on the trust enabler, implementing a security architecture with a Policy Decision Point that allows a Policy Enforcement Point to authorize marketplace requests. Last but not least, it also implements some of the functional components of the interoperability enabler.

- The Data Processing Pipeline comprises a collection of software tools designed to process and increase the quality of data assets. These tools work cohesively to enhance raw data modelled following the Data Asset Information, storing the resulting data assets in an NGSI-LD context broker (part of the storage enabler) for further publication in the marketplace. During the different pipeline steps, heterogeneous functional components from the data processing enabler, the data curation enabler and the interoperability enabler are implemented.
- The AI Pipeline is a system composed of interconnected components designed for training or optimising machine learning models and providing inference and analytics. It takes advantage of existing Data Assets and AI Models, serving as a versatile engine for processing data and generating valuable insights. It can support either local or distributed machine learning models and its results are stored in a file server or object storage system (part of the storage enabler) for further consumption. Besides, in order to enable distributed AI and AI as a service, it can implement a distributed training service to be offered as part of the marketplace. During the different AI pipeline steps, heterogeneous functional components from the AI enabler and the interoperability enabler are implemented.
- The formatting engines are in charge of transforming external information models into more specific ones which can be further processed by either the data processing pipelines and the AI pipelines. They are part of the interoperability enabler.

9 Conclusions

This deliverable presented the first complete version of the SEDIMARK functional architecture that describes the main functional and system components of the SEDIMARK decentralised marketplace. SEDIMARK aims to provide a fully decentralised system for users, providers, companies and researchers to be able to discover and share data, ML models and services in a trusted way. In this respect, SEDIMARK provides a complete set of tools for managing the data through their whole lifecycle, starting from data generation to data processing, data sharing and data exploitation. A major focus of SEDIMARK is on managing the quality of data, thus it provides tools for assessing data quality and for improving the quality of the data. The tools are adaptive to the technical expertise of the user, being able to run in an autonomous way or being fully configurable so that expert users can get the best quality of data in the end. Additionally, SEDIMARK provides tools to build and optimise machine learning models built leveraging the data, either in a fully localised way or in a distributed but privacy preserving way. Furthermore, SEDIMARK adopts well known standards for ensuring interoperability of the data and models that will be shared so that they can get the widest possible adoption.

Towards achieving the highly ambitious goals of the SEDIMARK project, this deliverable presented the main concepts and ideas that govern the activities of the rest of the workpackages. Starting from the definitions of the key terms being used throughout the document, it sets the ground for understanding what the key concepts and project entities are and how all these are interconnected, working together for achieving the project objectives.

Following common methodologies for deriving system architectures, this deliverable presented the requirement analysis, based on the requirements defined in Deliverable D2.1, mapping the requirements to the project objectives and identifying the key functionalities that are needed to meet the requirements. These functionalities were then converted to functional components, which are the main entities that compose the functional view of the architecture. The functional components were then grouped into 10 functional enablers, each one being responsible for a main system functionality (i.e. data processing, interoperability, etc.).

Considering that SEDIMARK does not aim to reinvent the wheel, building a whole system architecture from scratch, in Section 6 this document also presented a mapping of the system architecture to key state of the art architectures of EU initiatives and projects, showing how SEDIMARK reuses concepts and ideas, but adapting them to the specific needs of the project.

As a step forward after presenting the project architecture, this deliverable presented examples of data flows for various main SEDIMARK services, showing how these services can be instantiated using the functional architecture, what components need to interact and what type of information is being exchanged in order to realise the service. Of course, these should be considered as initial ideas and draft data flows, considering that they will be detailed and improved in the technical workpackages of the project during the development phase.

As the final chapters of this document, initial ideas about the system view of the architecture and the functional interfaces are also presented, showing how the system can be instantiated in the real world and how the various entities are interconnected within the platform and with external entities.

Summarising, this deliverable presented the first version of the SEDIMARK architecture and interfaces. It should be considered as a live document, which will be modified, adapted and improved as the technical development and testing activities will progress in the rest of the workpackages, aiming to revise the architecture based on their feedback. Additionally, SEDIMARK is also involved in the Data Spaces Support Community, which aims to build concepts and architecture for data spaces within the EU. SEDIMARK's involvement in the DSSC will also provide a forum for discussions about the SEDIMARK architecture receiving feedback from a wider community. Thus, a new version of the document will be presented in 12 months from now, in September 2024, which will provide and updated and final version of the SEDIMARK architecture.

10 Bibliography

[1] SEDIMARK, Deliverable 2.1: Use cases definition and initial requirement analysis, SEDIMARK, June 2023.

[2] N. Rozanski and E. Wods, Software systems architecture: working with stakeholders using viewpoints and perspectives, Addison Wesley, 2012.

[3] A. Bassi and e. al., Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model., Springer Nature, 2013.

[4] INTERNATIONAL DATA SPACES ASSOCIATION, "IDS RAM 4.0," December 2022. [Online]. Available: https://docs.internationaldataspaces.org/knowledge-base/ids-ram-4.0.

[5] Gaia-X, "GXDCH Gaia-X Digital Clearing House," 2023. [Online]. Available: https://gaia-x.eu/gxdch/.

[6] I3-Market, [Online]. Available: https://cordis.europa.eu/project/id/871754.

[7] SMART4ALL, [Online]. Available: https://smart4all-project.eu .

[8] OMEGA-X consortium, "OMEGA-X EU project website," 2022. [Online]. Available: https://omega-x.eu/.

[9] DSBA, "Data Spaces Business Alliance," [Online]. Available: https://data-spaces-business-alliance.eu/] .

[10] IDS-G, "IDS-G," [Online]. Available: https://github.com/International-Data-Spaces-Association/IDS-G.

[11] IDS, "IDS Information Model," [Online]. Available: https://w3id.org/idsa/core.

[12] IDS , "IDS protocol," [Online]. Available: https://docs.internationaldataspaces.org/dataspace-protocol/overview/readme.

[13] W3, [Online]. Available: https://www.w3.org/TR/vc-data-model/.

[14] GAIA-X, [Online]. Available: https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/.

[15] DSSC, [Online]. Available: https://dssc.eu/.

[16] DSSC, "Data Spaces 101," [Online]. Available: https://dssc.eu/space/SK/32407574/1+Data+Spaces+101.

[17] DSSC, "DSSC Starter Kit," [Online]. Available: https://dssc.eu/space/SK/29523973/Starter+Kit+for+Data+Space+Designers+%7C+Version+1.0+%7C+March+2023?attachment=/rest/api/content/29523973/child/attachmen

t/att110592007/download&type=application/pdf&filename=DSSC-Starterkit-Version-1.0.pdf.

[18] DSSC, “DSSC Glossary,” [Online]. Available: https://dssc.eu/space/Glossary/55443460/DSSC+Glossary+%7C+Version+1.0+%7C+March+2023?attachment=/rest/api/content/55443460/child/attachment/att110362680/download&type=application/pdf&filename=DSSC-Data-Spaces-Glossary-v1.0.pdf.

[19] IEC SyC Smart Energy, “IEC SRD 63200 - SGAM basics,” 2023. [Online]. Available: https://syc-se.iec.ch/deliveries/sgam-basics/.

[20] ISO, “ISO/IEC/IEEE 42010:2011 Systems and software engineering - Architecture description,” [Online]. Available: https://www.iso.org/standard/50508.html.

[21] GitHub Inc., “Eclipse EDC Data Space Connector,” Eclipse, 2023. [Online]. Available: https://github.com/eclipse-edc/Connector.

[22] GAIA-X, “GAIA-X SSI Self Description,” [Online]. Available: https://docs.gaia-x.eu/technical-committee/architecture-document/22.10.

[23] IDS, “IDS RAM 4.0,” [Online]. Available: https://docs.internationaldataspaces.org/ids-ram-4/.

[24] European Commission, “COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS A European strategy for data,” 19 February 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52020DC0066.

[25] S. Popov, “The Tangle,” April 30, 2018. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf .

[26] E. Drasutis, “"IOTA Smart Contracts", IOTA Foundation white paper,” 15 November 2021. [Online]. Available: https://files.iota.org/papers/ISC_WP_Nov_10_2021.pdf.

[27] W. Entriken, D. Shirley, J. Evans and N. Sachs, “"ERC-721: Non-Fungible Token Standard,"Ethereum Improvement Proposals, no. 721,” January 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721.

[28] F. Vogelsteller and V. Buterin, “"ERC-20: Token Standard,"Ethereum Improvement Proposals, no. 20,” November 2015. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20.

[29] Preukschat and Reed, “"Self-Sovereign Identity – Decentralized digital identity and verifiable credentials". Manning, Shelter Island, NY,” 2021. [Online]. Available: https://www.manning.com/books/self-sovereign-identity.

[30] W3C, “"Decentralized Identifiers (DIDs) v1.0. Core architecture, data model, and representations," W3C Recommendation,” 2022. [Online]. Available: https://www.w3.org/TR/did-core/.

[31] W3C, “W3C, "Verifiable Credentials Data Model v1.1," W3C Recommendation,” 2022. [Online]. Available: https://www.w3.org/TR/vc-data-model/.

[32] W3C, “W3C, "DID Specification Registries. The interoperability registry for Decentralized Identifiers," W3C Group Note,” 2023. [Online]. Available: https://www.w3.org/TR/did-spec-registries/.

[33] GAIA-X, [Online]. Available: https://gaia-x.gitlab.io/technical-committee/federation-services/federation-service-specifications/L13_IP_POR/ip_por/#references.

[34] GAIA-X, “GAIA-X 22.04,” [Online]. Available: https://docs.gaia-x.eu/technical-committee/architecture-document/22.04/ecosystem/.

[35] Big Data Value Association, European big data value strategic research and innovation agenda, Brussels: BDVA, 2017.

[36] OpenDEI, [Online]. Available: https://www.opendei.eu.

[37] OpenDEI, “OPENDEI ref architecture,” [Online]. Available: https://www.opendei.eu/wp-content/uploads/2022/10/REFERENCE-ARCHITECTURES-AND-INTEROPERABILITY-IN-DIGITAL-PLATFORMS.pdf.

[38] kaggle, [Online]. Available: https://www.kaggle.com/docs/api.

[39] ckan, [Online]. Available: https://docs.ckan.org/en/2.10/api/index.html.

[40] GAIA-X, “GAIA-X Fed Catalog,” [Online]. Available: https://gitlab.com/gaia-x/data-infrastructure-federation-services/cat/architecture-document/-/jobs/artifacts/main/raw/build/pdf/architecture/catalogue-architecture.pdf?job=generate_pdf .

[41] Nextis, [Online]. Available: https://nextis.org.

[42] Antonio Kung, Sergio Gusmeroli, Gabriella Monteleone, Alberto Dognini , Luc Nicolas and Carmen Polcaro, Reference Architectures and Interoperability in Digital Platforms, OpenDEI, 2022.