

Project acronym: CEDAR

**Project full title:** Common European Data Spaces and Robust AI for Transparent Public Governance

**Call identifier:** HORIZON-CL4-2023-DATA-01

Type of action: HORIZON-RIA

Start date: 01/01/2024

End date: 31/12/2026

Grant agreement no: 101135577

## D3.2 DataOps, MLOps, and Secure CEDS Connectors

**Document description:** Results from WP3 tasks, initially focusing on the SotA analysis and refining research gaps, and then presenting results addressing them.

Work package: WP3

**Author(s):** Anastasios Nikolakopoulos (NCI), Charalampos Ipeksidis (NCI), Filodamos Papanatsios (NCI), Davide Profeta (ENG), Nicola Leonardi (ENG), Silvio Sorace (ENG), Jose Miguel Blanco (UPM), Amaia Gil Lerchundi (VICOM), Thomas Marchioro (CEA), Jolanda Modic (ICS)

**Editor(s):** Charalampos Ipeksidis (NCI)

**Leading partner:** NCI (Netcompany-Intrasoft S.A.)

Participating partner: ART, CEA, CERTH, ENG, ICS, INS, SNEP, TRE, UBI, UPM, VICOM

Version: 1.0

Status: Final

Deliverable type: DATA

Dissemination level: PU

Official submission date: 30/06/2025

Actual submission date: 01/07/2025



The CEDAR project has received funding from the European Union's Horizon Europe project call HORIZON-CL4-2023-DATA-01 funded project Grant Agreement no. 101135577

## Disclaimer

This document contains material, which is the copyright of certain CEDAR contractors, and may not be reproduced or copied without permission. All CEDAR consortium partners have agreed to the full publication of this document if not declared “Confidential”. The commercial use of any information contained in this document may require a license from the proprietor of that information. The reproduction of this document or of parts of it requires an agreement with the proprietor of that information.

The CEDAR consortium consists of the following partners:

No.	Partner Organization Name	Partner Organization Short Name	Country
1	Centre for Research and Technology Hellas	CERTH	Greece
2	Commissariat al Energie Atomique et aux Energies Alternatives	CEA	France
3	CENTAI Institute S.p.A.	CNT	Italy
4	Fundacion Centro de Tecnologias de Interaccion Visual y Comunicaciones VICOMTECH	VICOM	Spain
5	TREBE Language Technologies S.L.	TRE	Spain
6	Brandenburgisches Institut für Gesellschaft und Sicherheit GmbH	BIGS	Germany
7	Christian-Albrechts University Kiel	KIEL	Germany
8	INSIEL Informatica per il Sistema degli Enti Locali S.p.A.	INS	Italy
9	SNEP d.o.o	SNEP	Slovenia
10	YouControl LTD	YC	Ukraine
11	Artelligence	ART	Ukraine
12	Institute for Corporative Security Studies, Ljubljana	ICS	Slovenia
13	Engineering – Ingegneria Informatica S.p.A.	ENG	Italy
14	Universidad Politécnica de Madrid	UPM	Spain
15	Ubitech LTD	UBI	Cyprus
16	Netcompany-Intrasoft S.A.	NCI	Luxembourg
17	Regione Autonoma Friuli Venezia Giulia	FVG	Italy
18	ANCEFVG – Associazione Nazionale Costruttori Edili FVG	ANCE	Italy
19	Ministry of Interior of the Republic of Slovenia / Slovenian Police	MNZ	Slovenia
20	Ministry of Health / Office for Control, Quality, and Investments in Healthcare of the Republic of Slovenia	MZ	Slovenia
21	Ministry of Digital Transformation of the Republic of Slovenia	MDP	Slovenia
22	Celje General Hospital	SBC	Slovenia
23	Transparency International Deutschland e.V.	TI-D	Germany
24	Katholieke Universiteit Leuven	KUL	Belgium
25	Arthur's Legal B.V.	ALBV	Netherlands
26	DBC Diadikasia	DBC	Greece
27	The Lisbon Council for Economic Competitiveness and Social Renewal asbl	LC	Belgium
28	SK Security LLC	SKS	Ukraine
29	Fund for Research of War, Conflicts, Support of Society and Security Development – Safe Ukraine 2030	SU	Ukraine
30	ARPA Agenzia Regionale per la Protezione dell’Ambiente del Friuli Venezia Giulia	ARPA	Italy

## Document Revision History

Version	Date	Modifications Introduced
---------	------	--------------------------

Page 2 of 46

CEDAR – 101135577

D3.2 – DataOps, MLOps, and Secure CEDS Connectors V2

		Modification Reason	Modified by
0.1	13/05/2025	ToC released; partners matched to sections.	Anastasios Nikolakopoulos (NCI), Charalampos Ipeksidis (NCI)
0.2	25/05/2025	First round of input received by partners.	VICOM
0.3	30/05/2025	Second round of input received by partners.	ICS, CEA
0.4	05/06/2025	Third round of input received by partners.	ENG NCI
0.5	18/06/2025	Manuscript submitted for internal review.	NCI
0.6	22/06/2025	Internal review comments received.	UBI, CERTH
0.7	26/06/2025	Final Edits and provision of final version	NCI
1.0	01.07.2025	Final review and submission	CERTH

## Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise.

Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

## Table of Contents

Table of Contents	4
List of Terms and Abbreviations	6
Executive Summary	7
1 Introduction	8
1.1 Positioning of the Deliverable within CEDAR	8
1.2 Structure of the Deliverable	8
2 The CRISP-DM model	9
3 The DataOps Pipeline	10
3.1 DataOps Actual Implementation	10
3.1.1 Cedar DataOps pipeline for the synthetic dataset	10
3.1.2 The Knowledge Graph REST Service	12
4 MLOps Methodology	15
4.1 Introduction to MLOps	15
4.2 Core Principles of MLOps	15
4.2.1 Continuous Integration (CI) / Continuous Deployment (CD) Automations	16
4.2.2 Developments and Integrations in the ALIDA Platform	21
4.2.3 MLOps and CEDAR project	25
5 Integration with CEDS and Data Alignment Tools	27
5.1 Overview	27
5.2 Connectors Research & Final Selection	27
5.3 Connector Selection	28
5.4 The CEDAR Minimum Viable Dataspace (MVD)	28
5.4.1 Connector Deployment & Configuration	29
5.4.2 Enabling the Dynamic Attribute Provisioning Service (DAPS)	32
5.4.3 The CEDAR Connector UI	34
5.5 Autonomous Data Alignment Tool	35
6 Cybersecurity	37
6.1 Cybersecurity Risk Assessment	37
6.2 Real-Time Network Intrusion Detection System	39
7 Conclusion	41
8 List of References	42

## List of Figures

Figure 1. Synthetic data DAG implementation .....	11
Figure 2. A detail of the graph generated by the pipeline.....	12
Figure 3. A portion of the Swagger interface for the KG REST Service .....	14
Figure 4. MLOps Principles within Technical Components. Source from [25].....	16
Figure 5. The CI/CD Flow.....	17
Figure 6. The CI/CD Pipeline [33] .....	17
Figure 7. The Pipelines built for all partners in Jenkins .....	18
Figure 8. The main flow in the CI/CD pipeline .....	19
Figure 9. ALIDA Platform – Model upload and registration.....	22
Figure 10. MLflow – Model experiments.....	23
Figure 11. ALIDA Platform – Model details.....	24
Figure 12. API Key Management.....	25
Figure 13. The CEDAR Minimum Viable Dataspace (MVD) Architectural Overview .....	29
Figure 14. Screenshot from the connector's UI in the home page .....	31
Figure 15. Screenshot from the connector's UI in the Catalog Browser, exploring assets from other Dataspace participants.....	32
Figure 16. Screenshot from Keycloak DAPS's UI, listing two connectors as Clients in the 'CEDS' Realm .....	34
Figure 17. Screenshot from the - still under development - User Interface's home page of the CEDAR Connector .....	35
Figure 18. Screenshot from the - under development - User Interface's Catalog Browser of the CEDAR Connector, using mocked data .....	35
Figure 19. Sigmo-IDS' s High-Level Multi-Probe Architecture .....	39

## List of Terms and Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
BoW	Bag of Words
BDA	Big Data Analytics
CD	Continuous Delivery
CI	Continuous Integration
CM	Continuous Monitoring
CT	Continuous Training
DoA	Description of Action
DVC	Data Version Control
CEDS	Common European Data Spaces
CRISP-DM	Cross-Industry Standard Process for Data Mining
DevOps	Software Development and IT Operations
DSML	Data Science and Machine Learning
EDA	Exploratory Data Analysis
ETL	Extract, Transform, Load
ETSI	European Telecommunications Standardization Institute
EU	European Union
GNN	Graph Neural Network
IDSA	International Data Spaces Association
KG	Knowledge Graph
ML	Machine Learning
MVG	Minimum Viable Graph
NGSI-LD	Next Generation Service Interfaces – Linked Data
nIDS	network Intrusion Detection System
NLP	Natural Language Process
PCA	Principal Component Analysis
RFE	Recursive Feature Elimination
SDM	Smart Data Models
SFS	Sequential Feature Selection
SotA	State of the Art
TF-IDF	Term Frequency-Inverse Document Frequency
WP	Work Package

## Executive Summary

This deliverable provides an update on the work carried out in WP3, focusing on the progress made since the submission of D3.1 at M06. The main objective remains to improve how data within the CEDAR ecosystem will be collected, cleaned, and managed, using modern DataOps and MLOps approaches. During this period, the CEDAR platform's software assets and workflows have been refined, aiming to make each technical offer more reliable and scalable.

Furthermore, significant progress has been made in the development, testing, and establishment of a Minimum Viable Dataspace for CEDAR, securely interconnecting the three pilots of the project for seamless data exchange. This is a step closer towards CEDAR Dataspace's ability to integrate with CEDS. These improvements help ensure that EU-wide data-driven services are more effective and inclusive.

In addition, the consortium has advanced the development of secure data management technologies. Security requirements defined in WP1 have been followed, ultimately aiming to verify the robustness of the software solutions of CEDAR. Overall, this deliverable documents the technical updates, outcomes, and challenges addressed during this stage of WP3.

## 1 Introduction

Within the CEDAR project, WP3 is responsible, in alignment with the roadmap defined in WP1 and data modelled in WP2, for the efficient, scalable, secure management of big data and their integration with CEDS. It also ensures cybersecurity of CEDAR technologies through cybersecurity risk assessment, security mechanisms, and penetration testing. To facilitate these goals, DataOps, MLOps, and CEDS connectors have been developed and delivered during the reporting period.

WP3 plays a key role in the CEDAR project. It focuses on the secure, scalable, and efficient management of big data, in close alignment with the roadmap from WP1, as well as the data models produced in WP2. Furthermore, it contributes to the safe adoption of CEDAR technologies by the pilots of the project. In order to do so, WP3 addresses cybersecurity through risk assessments, protective mechanisms, and penetration testing, as outlined in the previous paragraph. Especially, the deliverable D3.2 DataOps, MLOps, and Secure CEDS Connectors V2 reports on the development effort invested to improve the quality, diversity and representation of readily available datasets coming from the CEDAR Pilots, train robust models, and deploy a stable data platform that functions as the first release of the CEDAR Data Space. In addition, the Integration with CEDS has shown significant progress since its original analysis in D3.1, moving from the initial research on current Dataspace connectors, to the development of a Dataspace specifically for CEDAR. All assets present their transition from initial planning and implementation, towards finalized software versions, which will be part of the integrated CEDAR platform.

This deliverable (D3.2) builds on the foundation laid in D3.1, presenting the updates and progress made across all WP3 tasks. It highlights how:

- i) DataOps and MLOps pipelines have evolved.
- ii) Integration with the Common European Data Spaces (CEDS) - through the establishment of a CEDAR-specific Dataspace - has advanced.
- iii) Security requirements have been continuously addressed.

The focus remains on enabling reliable, secure data-driven solutions that can scale across the EU.

### 1.1 Positioning of the Deliverable within CEDAR

D3.2 represents the second of three documents. It shifts its focus from the original SotA analysis and refining research gaps of D3.1, to the implementation aspects of the components listed in WP3. Thus, it serves to describe the progress on each of the developed technologies. The final description and analysis of the technologies will be presented in D3.3.

### 1.2 Structure of the Deliverable

The information in this document is structured as follows:

- Section 1 provides a brief overview of this deliverable's objectives.
- Section 2 presents an analysis of the CRISP-DM model for directing data mining projects.
- Section 3 describes the DataOps pipeline, along with the progress made since D3.1.
- Section 4 presents the MLOps methodology for the efficient management and the work carried out during the previous months of the project.
- Section 5 presents the progress on Integration with CEDS and the steps towards the establishment of a true Minimum Viable Dataspace, specifically for the needs of CEDAR. It also includes progress on the Data Alignment tool.
- Section 6 describes how to conduct a (manual) continuous cybersecurity risk assessment of the CEDAR assets.
- Section 7 briefly presents the conclusions, as well as the outlook on the next development phase of CEDAR.



## 2 The CRISP-DM model

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is a well-established method for directing data mining projects.

Published in 1999 to standardize data mining processes across various industries, it has since become the most prevalent methodology for data mining, analytics, and data science projects. Even today, CRISP-DM remains the most widely used approach for data science projects [1].

There are two main ways of interpreting the method:

- As a *Methodology* and guide, it outlines the typical phases of a project, details the tasks associated with each phase, and explains the relationships and dependencies between these tasks. Essentially, it offers a structured approach to planning a data mining project, addressing the question of "How to do it".
- As a *Process Reference* model, CRISP-DM gives an overview of the data mining life cycle, describing common approaches used by data mining experts. It answers the question of "What to do".

Since no changes have been made to the applied methodology, this section remains unchanged from the previous version. Therefore, the reader is referred to Deliverable D3.1 for a complete overview.

### 3 The DataOps Pipeline

DataOps is a methodology inspired by DevOps that integrates data engineering, integration, and quality practices to enhance collaboration and efficiency among data professionals such as data scientists and engineers. While the specific steps may vary across projects or organizations, a typical DataOps pipeline includes:

- **Data Collection:** Gathering data from various sources like databases, APIs, files, or streaming platforms.
- **Data Cleaning and Selection:** Removing duplicates, cleaning inconsistencies, and selecting relevant features.
- **Data Construction:** Applying processes such as feature engineering, normalization, and harmonization.
- **Data Formatting:** Structuring data to fit its intended use (e.g., transforming tables into graphs).
- **Data Storage:** Saving the processed data in appropriate storage systems like data lakes, warehouses, NoSQL, vector databases, graph databases, or relational databases.

The content of Sections 3.1 through 3.6.6 (of the previous version of this document - namely D3.1) has not been modified, therefore these sections are not reported in this version of the document. Readers are referred to the previous version of the document for the full description.

#### 3.1 DataOps Actual Implementation

Given the project characteristics and updated requirements, Apache Airflow [21] has been selected as the technology for DataOps due to its comprehensive workflow orchestration capabilities that address the complex requirements of CEDAR project data operations. It provides good flexibility in scheduling and triggering data pipelines through both time-based and event-based conditions, including user actions or Kafka message, while enabling sophisticated task dependency management that ensures proper execution order. Its robust monitoring and observability features include built-in task tracking, comprehensive logging, and configurable email alerts that provide real-time visibility into pipeline health and performance. The platform's ability to handle complex workflows is enhanced by its support for fallback plans and error handling, ensuring data pipeline reliability. Additionally, Airflow's user-friendly interface allows for easy variable management and configuration changes without code modifications, making it particularly well-suited for dynamic DataOps environments like the CEDAR one, where adaptability and operational control are essential for maintaining efficient data processing workflows. All these are great features to have also in relation to the actual implementation of pilot use cases.

Airflow has been correctly set up in the Kubernetes environment through the official Helm Charts. Additional efforts have been spent on integrating Apache Airflow with Keycloak [43] as the Identity and Access Management (IAM) system. Keycloak is an open-source solution that offers a comprehensive layer for authentication and authorization purpose.

Thanks to this security layer, only authenticated users with the right roles can use both the Airflow graphical interface and its APIs, excluding any unauthorized use.

##### 3.1.1 Cedar DataOps pipeline for the synthetic dataset

For the purposes of the project, CERTH created a synthetic dataset which, based on the insights extracted from the Slovenian pilot, reflects both standard and potentially fraudulent instances that shows the structural, semantic, and variable characteristics of authentic procurement data (ref. Deliverable D2.2).

At this stage of the project, using this synthetic dataset brings numerous advantages since it is ready to use in the sense that there is no personal data or data attributable to natural persons to be managed inside the pipeline.

Furthermore, the tabular format (.csv files) allowed its immediate use without the need for an OCR (Optical Character Recognition) or NER (Named entity recognition) pre-process steps.

This dataset allowed to focus efforts directly on the graph generation by moving the management of these above-mentioned tasks to future releases of the pipeline.

In this setting Apache Airflow orchestrates the data processing workflows by starting data extraction from a source MinIO [44] bucket, transforming them, and saving results to a destination MinIO bucket following a typical ETL (Extract, Transform, Load) pattern.

The DAG executes tasks using operators like PythonOperator or KubernetesPodOperator. These tasks can perform any custom processing logic like data cleaning, aggregation, and finally the knowledge graph creation.

This approach provides a scalable, reliable data processing workflow that can handle everything from simple file transformations to complex multi-stage analytics pipelines between MinIO storage systems.

The pipeline implemented for the synthetic dataset is described in the following figure.

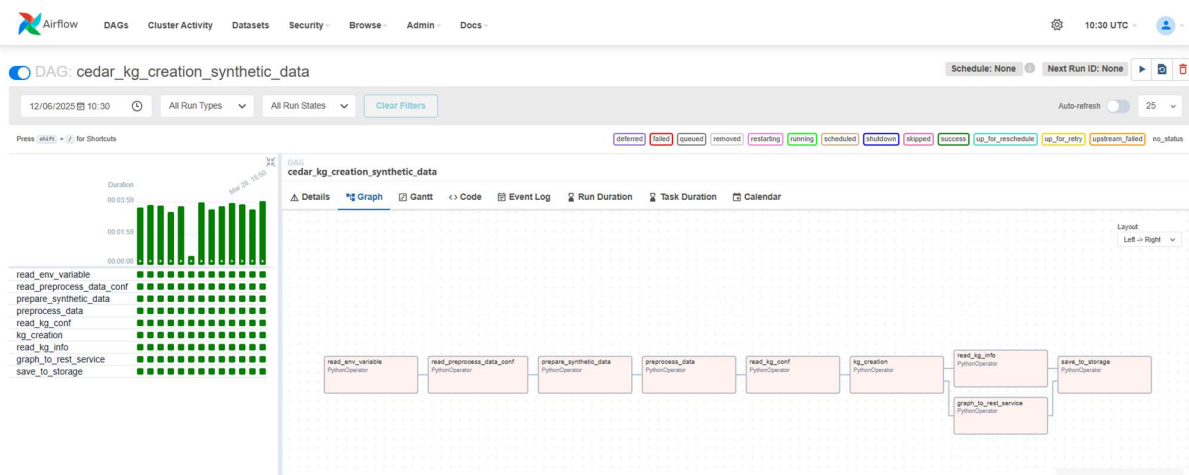


Figure 11. Synthetic data DAG implementation

It consists of the following sequence of tasks, each of which will be briefly described.

- **read\_env\_variable**: reads environment variables like secret and access keys, set during the deployment configuration.
- **read\_preprocess\_data\_conf**: reads the configuration json file that defines how the different .csv files should be aggregated and linked to create all the attributes of the entities described in the project Data Model (ref. Deliverable D2.2).
- **prepare\_synthetic\_data**: modifies and prepares data to conform to a standardized Data Model format. It applies a set of customized functions to the input data and serves as initial preprocessing step in the DAG execution.
- **preprocess\_data**: normalizes data, ensuring it is ready for analysis. This operator handles data cleaning steps like duplicate removal, ensures consistency in handling missing (None) values and standardize data types (such as datetime format).
- **read\_kg\_conf**: reads the json configuration file that defines attributes and features of the generated knowledge graph. Nodes and links are created based on these settings. Additional attributes like the output graph name and the max amount of desired nodes can also be managed.
- **kg\_creation**: the task involved in the knowledge graph generation. The graph is built with the python networkX library [83].
- **read\_kg\_info**: reads information about the created knowledge graph. Reading the logs of this task is extremely important because it provides immediate feedback on the correctness and completeness of the operation just performed.
- **graph\_to\_rest\_service**: if enabled by the config file, the task sends the newly created graph to the Knowledge Graph Rest server making it immediately available for client requests.
- **save\_to\_storage**: the newly created graph is saved on a dedicated MinIO bucket.

This division into consecutive tasks allows for careful analysis of the logs and therefore to track easily any problems that may have arisen during the execution of the pipeline.

The following figure shows a portion of a graph generated from the synthetic dataset.

It is evident how the effort in implementing a correct pipeline is rewarded in a new representation of the information that is much richer and more explicit. Based on that it is now possible to conduct a multitude of sophisticated analyses, activities that were initially impossible given the raw and disorganized nature of the starting data.

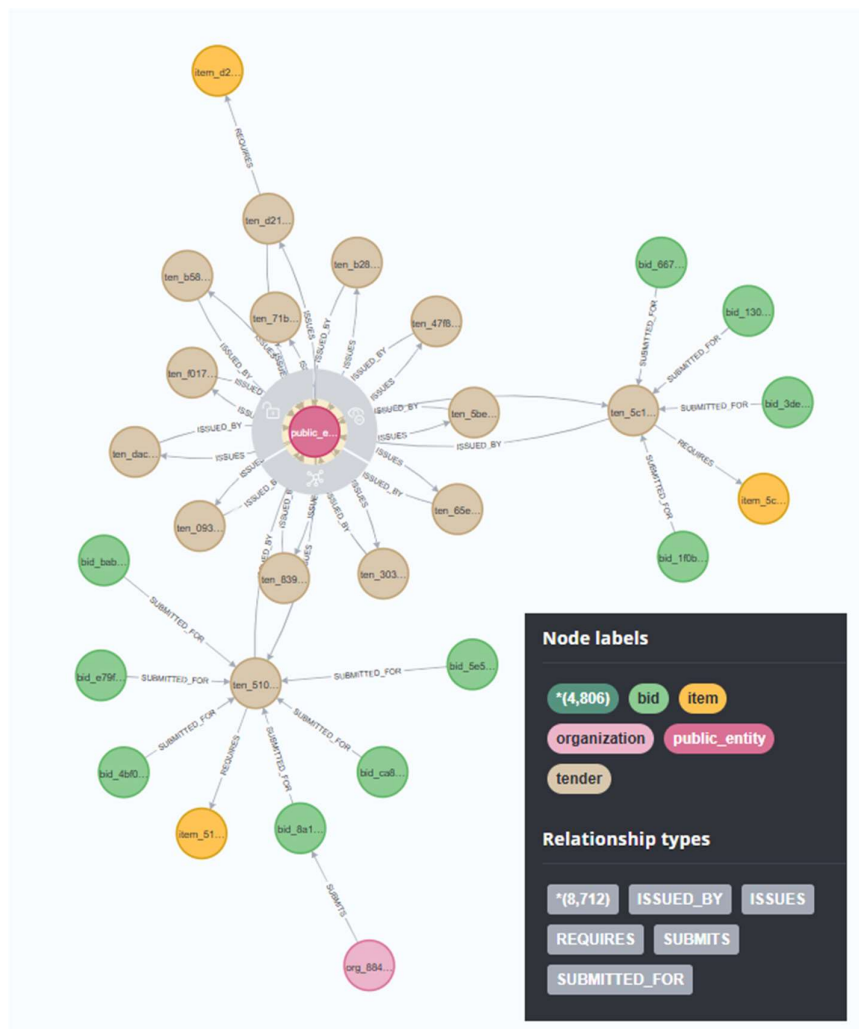


Figure 22. A detail of the graph generated by the pipeline

Regarding real-world pilot pipelines, they will follow a similar approach, adding some preliminary steps in relation to the particular type and format of the pilot data. The integration of the NER and pseudonymization services is currently in progress.

### 3.1.2 The Knowledge Graph REST Service

As seen in the previous paragraph, the pipeline ends with the creation of the graph. It proceeds to send to a specific service, the Knowledge Graph REST service. Since automatic sending from the DataOps pipeline is optional, if disabled the new graph can be loaded at a later time.

This is a standard REST API service that provides the following main features:

- Multiple Property-Graphs management.
- Graph generation and manipulation (CRUD -Create, Read, Update, Delete- operations).
- Graph Analytics & Metrics:
  - Centrality
  - Clustering
  - Connected Component
  - Assortativity
  - Path-related (shortest/longest path)
  - Communities' extraction
  - Other
- Sub Graph extraction (Node Ego Graph).
- Nodes, edges filtering by attributes.
- Dynamic nodes and edges management.
- Neo4j graph generation. This endpoint transforms the selected kg into a Neo4j compatible format and push it into a Neo4j instance.
- Neo4j Cypher query forwarding. This endpoint allows to send queries in a Neo4j-like format using all the benefits of the graph representation effortlessly. This endpoint also allows to build all the custom indicators useful for the analyses carried out by the pilots.

## Nodes

POST	/graphs/{graphname}/nodes	Graph Nodes Details Filtered By Attributes	🔒
POST	/graphs/{graphname}/nodes/create	Create Or Merge A Node (If The Node Already Exists)	🔒
DELETE	/graphs/{graphname}/nodes/{nodeid}	Delete A Node From A Specified Graph	🔒
GET	/graphs/{graphname}/nodes/{nodeid}	Graph Node Information By Node Id	🔒
GET	/graphs/{graphname}/nodes/{nodeid}/ego_graph	Ego Graph From Node Id	🔒
GET	/graphs/{graphname}/nodes/{nodeid}/neighbors	Node Neighbors From Node Id	🔒

## Edges

POST	/graphs/{graphname}/edges	Graph Edge Details Filtered By Attributes	🔒
POST	/graphs/{graphname}/edges/create	Create Or Merge An Edge (If The Edge Already Exists)	🔒
DELETE	/graphs/{graphname}/edges/{source}/{dest}	Delete An Edge By Source And Dest Ids	🔒
GET	/graphs/{graphname}/edges/{source}/{dest}	Graph Edge Information By Edge Source And Destination Ids	🔒
GET	/graphs/{graphname}/edges/{extreme}	Graph Edges Information By An Edge Extreme Id	🔒

## Paths

GET	/graphs/{graphname}/paths/shortestpath/{source}/{dest}	Shortest Path Between Source And Destination Node Ids	🔒
GET	/graphs/{graphname}/paths/shortest_simple_paths/{source}/{dest}	Shortest Simple Paths Between Source And Destination Node Ids	🔒
GET	/graphs/{graphname}/paths/longestpath	Longest Path In A Directed Acyclic Graph (Dag)	🔒

## Neo4j

POST	/neo4j/graphcopy/{graphname}	Copy The Graph To Neo4J	🔒
DELETE	/neo4j	Delete The Graph From Neo4J	🔒
POST	/neo4j/query/	Query To Neo4J Db	🔒

Figure 33. A portion of the Swagger interface for the KG REST Service

This service also is completely integrated with the IAM Keycloak solution, preventing unauthorized users from accessing data or manipulating information.

By providing a working technology for the creation of the knowledge graph through Airflow and a dedicated service to get all the information from the graph itself, the solution covers the entire path of data ingestion, elaboration and the final data consumption through a graph format.

## 4 MLOps Methodology

With the increasing integration of machine learning and artificial intelligence into various aspects of daily life and business operations, the efficient management and deployment of ML models have become essential. MLOps (Machine Learning Operations) addresses this need. This chapter provides an overview of MLOps, beginning with an "Introduction to MLOps". Next, the "Core Principles of MLOps" are examined to highlight the practices that ensure effective implementation. The section "MLOps Key Components" breaks down the essential elements required for a robust MLOps framework. Finally, "MLOps Technologies and Tools" explores the various technologies and tools that facilitate reliable and scalable MLOps practices, ensuring smooth transitions from development to production.

### 4.1 Introduction to MLOps

MLOps [22], a combination of "Machine Learning" and "Operations," involves a set of practices aimed at making the lifecycle of machine learning models more efficient and streamlined. The main goal of MLOps is to bridge the gap between data science and operational teams, enabling better collaboration and faster deployment of ML models. Traditionally, data scientists focus on creating and training models, while operational teams manage deployment and monitoring. MLOps brings these roles together, encouraging continuous integration and continuous deployment (CI/CD) specifically for ML. This approach addresses challenges unique to machine learning, such as handling large datasets, ensuring model reproducibility, and maintaining models in production environments. By implementing MLOps, organizations can become more agile, reduce time-to-market, and ensure their ML models perform consistently and reliably. As AI-driven insights become increasingly critical for maintaining a competitive edge, MLOps provides a framework for scaling and sustaining machine learning projects effectively.

Introducing MLOps into an organization can significantly improve how ML projects are managed. MLOps fosters a DevOps-like culture where data scientists, engineers, and operations teams work closely together, ensuring that models are both technically robust and practically viable in production environments. This collaboration reduces the friction often seen between development and operations, leading to smoother transitions from model training to deployment. Additionally, MLOps encourages the use of standardized processes and tools, which can reduce duplication of effort and improve the efficiency of the ML lifecycle. By focusing on automation, MLOps minimizes manual intervention, which not only speeds up the deployment process but also reduces the risk of human error, ensuring models are deployed consistently and reliably. Overall, MLOps is a transformative approach that enhances the robustness and scalability of ML projects, ensuring they deliver real value to businesses.

### 4.2 Core Principles of MLOps

The core principles of MLOps emphasize automation, collaboration, continuous improvement, and governance. Automation is central to MLOps, reducing manual tasks and minimizing the risk of human error through automated model training, validation, and deployment pipelines. Collaboration between data scientists and operations teams ensures models are technically robust and aligned with business goals and operational needs. CI/CD practices are fundamental, enabling rapid iterations and updates of ML models in response to new data and evolving requirements. These practices ensure models are frequently and reliably updated, decreasing the time from development to production. Additionally, principles of reproducibility and traceability are critical, allowing for consistent model replication and easy tracking of changes over time. Governance, encompassing security, compliance, and ethical considerations, ensures ML models adhere to regulatory requirements and organizational policies. Collectively, these principles form a robust framework supporting the sustainable and scalable deployment of machine learning models. These MLOps principles are listed in Figure 4 along with the typical technical components found in a standard MLOps architecture.

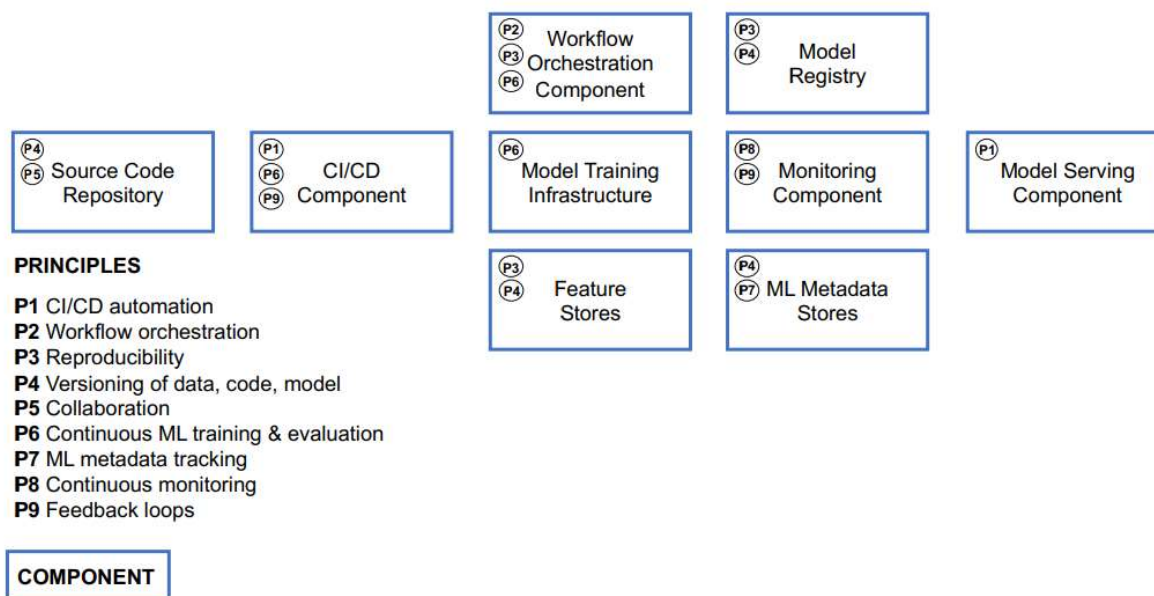


Figure 44. MLOps Principles within Technical Components. Source from [25]

Since no changes have been made to the methodology and state of the art, this section remains unchanged from the previous version for multiple paragraphs. Therefore, we refer the reader to the Deliverable D3.1 for a comprehensive overview of the state of the art of MLOps.

#### 4.2.1 Continuous Integration (CI) / Continuous Deployment (CD) Automations

The CEDAR project adopts Continuous Integration/Continuous Delivery (CI/CD) practices to streamline the project's software development lifecycle. CI/CD automates software testing and deployment, ensuring rapid integration of code while reducing errors. Continuous Integration is a set of practices and tools that allow for frequent code integration by automatically building and testing changes as soon as they are committed to a source code repository. Continuous Integration conventions and tools make this automation possible. Continuous Delivery completes the process by automatically deploying software after successful integration based on well-defined deployment pipelines that are predictable and easy to monitor and maintain. A well-designed CI/CD platform allows developers to deliver features without having to manage the underlying integration and deployment processes.

By using CI/CD, the CEDAR project ensures more efficient, reliable, and secure software delivery. These practices foster collaboration among partners, making the integration and deployment process faster and less prone to human error.

##### 4.2.1.1 The CI/CD Stack

CEDAR's CI/CD pipeline is built using a suite of integrated tools. GitHub serves as the primary source control platform for code versioning and collaboration. Jenkins<sup>1</sup> automates the process of building, testing, and deploying software, leveraging Docker for containerization. Docker enables the packaging of applications into portable containers, which are then managed in the Harbor Container Registry for secure storage and distribution.

<sup>1</sup> [Jenkins](#)





Figure 5. The CI/CD Flow

Other key components include NGINX [30] for managing network requests, Portainer [31] for user-friendly container management and monitoring, and pfSense [32] for secure access control using project specific VPNs. Finally, Keycloak is used for providing single sign-on (SSO) authentication, allowing all partners to securely access CI/CD dashboards, tools and services simply by using their GitHub credentials.

#### 4.2.1.2 Continuous Integration and Continuous Deployment Workflow in CEDAR

The CEDAR CI/CD workflow begins with local software development by partners, who then commit code updates to GitHub. Jenkins automatically detects these updates, by continuously polling the source code repositories for changes. Once a change is detected, Jenkins retrieves the code and executes a predefined pipeline associated with the repository in question. Such pipelines, build the relevant Docker images, run the provided integration and unit tests to ensure stability as well as perform static code analysis to ensure that low quality code is not being deployed. Once all tests pass a docker image is flagged as verified. Verified Docker images are then stored in the Harbor Container Registry and finally deployed in their respective environment (staging or production). This process not only accelerates integration but also provides a consistent and traceable deployment pathway which increases productivity while strengthening the quality of the software.

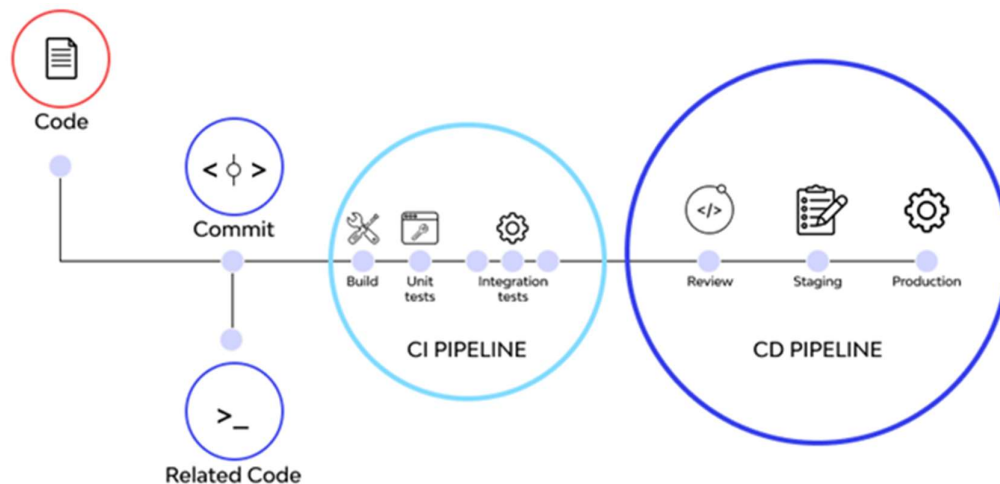


Figure 66. The CI/CD Pipeline [33]

As for the main CI/CD Tools utilized in the pipeline, they are as follows:

- **GitHub and Repository Structure:** Each repository must contain specific files (Jenkinsfile, Dockerfile, etc.) to enable the automated pipeline. Partners are onboarded into the CEDAR GitHub organization, with clear team structures and access controls.
- **Docker-compose Networking:** Partners configure their docker-compose files to connect with project-defined networks on specific servers, allowing inter-service communication during deployments.

- **pfSense and VPN Configuration:** pfSense secures access to the CI/CD environment, providing VPN-based connectivity. Each partner is issued a unique certificate for OpenVPN access, with setup steps detailed to ensure secure remote server connection.
- **Keycloak SSO Login:** Keycloak enables single sign-on for CI/CD services using GitHub as the identity provider. The initial login requires authorization and user group assignment, after which access to all dashboards is streamlined.
- **Jenkins Pipeline Setup and Operation:** Jenkins serves as the core automation tool, executing CI/CD pipelines based on Jenkinsfile definitions within each partner's repository. The Jenkins dashboard provides visibility into build and deployment statuses, while the Jenkinsfile.kill allows for clean removal of test deployments. Partners receive unique Harbor credentials for secure Docker image storage.
- **Harbor Registry Usage:** Harbor manages Docker images, providing project-specific storage for each partner. The Harbor dashboard allows users to review image history, manage tags, and control access, with mappings defined for each technology offering.
- **Portainer for Container Management:** Portainer offers a graphical interface for managing containers, volumes, and images on deployment machines. Due to its powerful control capabilities, access to Portainer is restricted and granted upon request.

























S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">CEA Pipelines</a>	N/A	N/A	N/A
		<a href="#">CERTH Pipelines</a>	N/A	N/A	N/A
		<a href="#">CNT Pipelines</a>	N/A	N/A	N/A
		<a href="#">Demo Pipelines</a>	N/A	N/A	N/A
		<a href="#">ENG Pipelines</a>	N/A	N/A	N/A
		<a href="#">INTRA Pipelines</a>	N/A	N/A	N/A
		<a href="#">MNZ Pipelines</a>	N/A	N/A	N/A
		<a href="#">SNEP Pipelines</a>	N/A	N/A	N/A
		<a href="#">TRE Pipelines</a>	N/A	N/A	N/A
		<a href="#">UBI Pipelines</a>	N/A	N/A	N/A
		<a href="#">UPM Pipelines</a>	N/A	N/A	N/A
		<a href="#">VICOM Pipelines</a>	N/A	N/A	N/A

Figure 7.7 The Pipelines built for all partners in Jenkins

To accommodate different technology offerings, CEDAR implements two main integration pipelines. The primary pipeline is for partners providing their codebase to GitHub, utilizing the full functionality of the CI/CD workflow. The

secondary pipeline is for partners who only provide pre-built Docker images of their software components. This flexible approach ensures that the needs of all partners are met efficiently.

#### 4.2.1.3 Source Code Pipeline

The source code pipeline leverages a dedicated GitHub organization, where each partner maintains their own repository containing the component codebase, Jenkinsfile, Dockerfile, and other necessary files. This structure allows Jenkins to automate the build, test, and deployment steps, while also providing role-based access and collaboration features. Docker-compose networking is configured to enable seamless communication between services across different deployment servers, with each server assigned a unique network. Access to the CI/CD environment is tightly controlled via pfSense VPN, ensuring only authorized partners with certificates can connect.

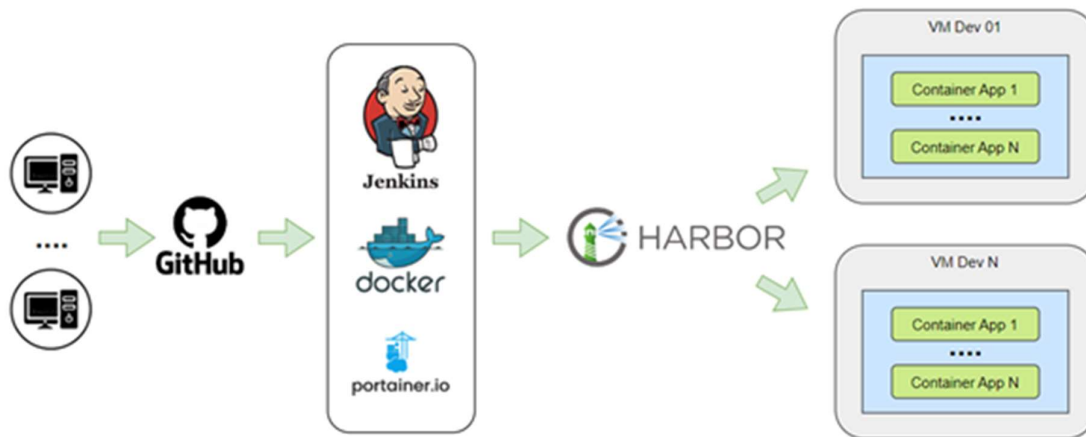


Figure 8.8 The main flow in the CI/CD pipeline

A step-by-step version of this process can be seen below.

1. Repository Setup
  - Each partner creates and maintains a dedicated repository within the CEDAR GitHub organization.
  - The repository must contain the component source code, a Jenkinsfile, Dockerfile, and any additional configuration files (e.g., docker-compose.yaml).
2. Local Development
  - Partners develop and test software components in their local environments.
  - Once ready, code changes are committed and pushed to the respective GitHub repository.
3. Continuous Integration Trigger
  - Jenkins continuously polls the GitHub repository for new commits or pull requests.
  - When changes are detected, Jenkins automatically triggers the pipeline defined in the Jenkinsfile.
4. Automated Build
  - Jenkins retrieves the latest source code.
  - The component is built into a Docker image according to the Dockerfile instructions.
5. Testing and Analysis
  - The pipeline executes automated unit tests and integration tests.
  - Static code analysis is performed to detect potential quality issues or security vulnerabilities.
6. Docker Image Verification
  - If all tests and checks pass, the resulting Docker image is flagged as verified.
7. Image Storage
  - The verified Docker image is pushed to the partner's Harbor Container Registry project.
8. Deployment

- Jenkins deploys the image to the designated environment (staging or production) for further validation and use.
9. Monitoring and Cleanup
- Deployment status and logs can be monitored via the Jenkins dashboard.
  - Once testing is complete, partners can trigger the `Jenkinsfile.kill` to remove test deployments and free resources.

It is important to highlight and remember that each GitHub repository must include:

- ✓ the component **codebase**
- ✓ a **Jenkinsfile** (accompanied by a **Jenkinsfile.kill**)
- ✓ a **Dockerfile**
- ✓ and/or **docker-compose.yaml** file

These files enable the creation of pipelines, docker images and containers while also performing tests. An example of a GitHub repository is provided in the CEDAR GitHub organization [34]

It should be noted that for the Large ML models pipeline, a GitHub repository must also be created which should include:

- ✓ a **Jenkinsfile** (accompanied by a **Jenkinsfile.kill**)

#### 4.2.1.4 Docker Image Pipeline

The Docker Image Pipeline provides an integration pathway for partners who deliver pre-built Docker images instead of source code. In this workflow, partners are responsible for building, testing, and packaging their software into Docker images before uploading them to the Harbor Container Registry assigned to their organization. Once uploaded, these images are subject to automated vulnerability and security checks within the CI/CD environment. Even though source code is not provided, each partner must still maintain a dedicated repository within the CEDAR GitHub organization for their component. This repository should include the specific Jenkinsfile and any necessary configuration files that define the steps Jenkins must perform, such as pulling the Docker image from Harbor, and deploying the image to the designated environment (staging or production). This ensures that all components, regardless of whether they are delivered as source code or as images, follow a unified, transparent, and traceable deployment process.

This process can be seen in a step-by-step manner below:

1. Repository Setup
  - Each partner maintains a dedicated repository in the CEDAR GitHub organization.
  - The repository includes a Jenkinsfile and any required configuration files, even though the source code itself is not provided.
2. Local Image Creation
  - Partners are responsible for building, testing, and packaging their software into Docker images locally, outside the CEDAR CI/CD environment, optionally the Continuous Integration's quality assurance tests can be run locally.
3. Image Upload
  - The completed Docker image is uploaded (pushed) by the partner to their assigned Harbor Container Registry project.
4. Pipeline Trigger
  - Jenkins is configured to monitor the repository (and optionally Harbor) for new versions or tags, triggering the Jenkinsfile-defined pipeline as required.
5. Image Retrieval
  - Jenkins pulls the specified Docker image from Harbor, based on tags or configuration in the Jenkinsfile.
6. Automated Checks

- The pipeline may execute vulnerability scans, or integration checks to ensure the image meets security and quality standards.
- 7. Deployment
  - Upon successful validation, Jenkins deploys the Docker image to the appropriate environment (staging or production).
- 8. Monitoring and Cleanup
  - The Jenkins dashboard provides status and logs for monitoring deployments.
  - Cleanup steps (if needed) can be included in the Jenkinsfile to remove old or test deployments.

#### 4.2.2 Developments and Integrations in the ALIDA Platform

The following frameworks have been integrated within the ALIDA platform to enable and support MLOps features:

- MLflow platform to facilitate the reproducibility, versioning, and management of ML experiments
- Seldon Core V2 [44] to automate the serving of ML models via REST APIs

Additional components have been integrated and developed to facilitate and enable the functionalities provided by MLflow and Seldon Core V2 framework such as Keycloak and MinIO.

These components have been properly installed and configured in K8S (Kubernetes) infrastructure using the official Helm Charts of each chosen component. For the custom components, the K8S manifest have been also defined.

##### 4.2.2.1 Keycloak

As part of the ALIDA platform's architecture, Keycloak [45] has been adopted as the IAM system. Keycloak is an open-source solution that offers comprehensive support for authentication and authorization, based on industry standards such as OAuth 2.0, OpenID Connect, and JWT (JSON Web Tokens). Its integration into ALIDA enables secure Single Sign-On (SSO) capabilities and facilitates federated identity with external providers (e.g., Google, LDAP, GitHub).

The use of OAuth 2.0 and JWT tokens provides significant advantages, such as:

- Stateless authentication mechanisms ideal for microservice-based architectures;
- Token-based fine-grained access control for users and services;
- Scalability and extensibility via integration with third-party identity providers;
- Built-in support for multi-tenancy and role-based access control.

This IDM setup serves as the foundation for securing all access flows within ALIDA, including model management, serving, and API gateway traffic.

The Keycloak instance can be reached at <https://security.cedar.alidalab.it> where an 'alida' realm and appropriate Keycloak clients have been configured to enable SSO through various applications of the ALIDA platform.

##### 4.2.2.2 MinIO

MinIO [46] has been chosen as the object storage backend for the ALIDA platform due to its high compatibility with the S3 API, scalability, and lightweight footprint suitable for Kubernetes-based deployments. It is used to store machine learning artifacts, such as trained model binaries and experiment logs, in a durable and versioned manner. ALIDA automatically provides each user registered on the platform with key/secret pairs to access the MinIO instance of the platform, both to store only the private data of that specific user, and keys to store in a shared space at team level, i.e. of users belonging to the same organisation. This integration is fundamental for enabling secure model management workflows and consistent storage for tracking and serving components across the MLOps lifecycle.

MinIO's graphical user interface can be accessed at <https://minio.cedar.alidalab.it>, which is, however, only used as an interface for administrators; MinIO's API from the outside can be accessed at <https://s3.cedar.alidalab.it>. Each user within the ALIDA platform will have customised keys and access to specific sub-folders within the 'alida' bucket on MinIO

due to the registration of specific data sources within the ALIDA catalogue. Through the ALIDA platform, as shown in the figure below, it is possible to upload and register data and models in the personal spaces provided through a defined data source pointing to the internal ALIDA MinIO.

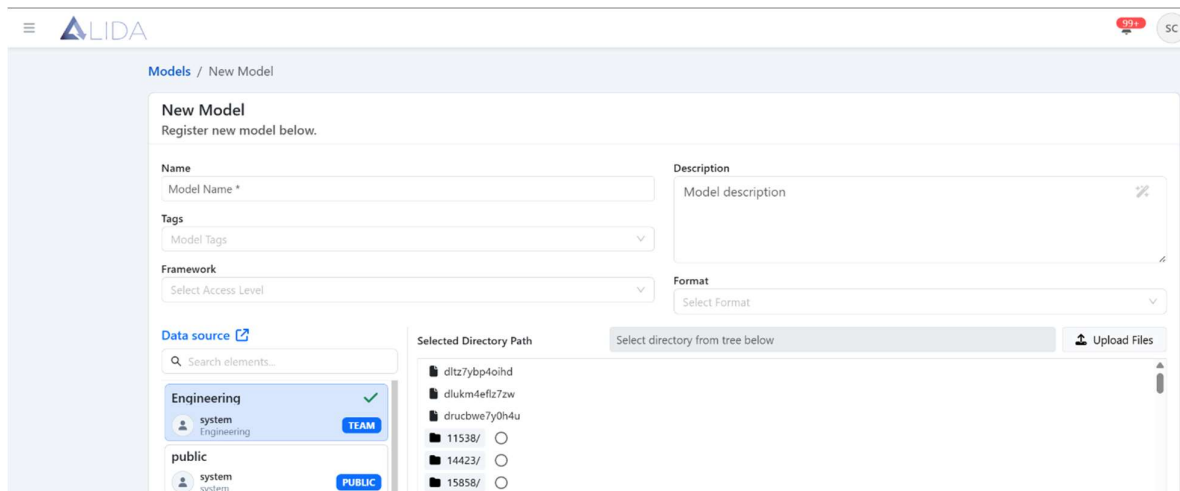


Figure 9.9 ALIDA Platform – Model upload and registration

#### 4.2.2.3 MLflow

MLflow [27] is a core component of a modern MLOps architecture because it provides the tools required to manage the entire machine learning experiment lifecycle. In ALIDA, its integration allows teams to track experiments, log parameters, metrics, and artifacts, and register trained models in a versioned way. This is essential for collaboration between data scientists and engineers, supports auditability and reproducibility, and helps ensure that only validated and approved models move forward to production.

The MLflow Tracking Server has been deployed in the ALIDA Kubernetes-based infrastructure. It is configured with MinIO as its backend artifact store to manage experiment logs, model artifacts, and metrics. This setup enables:

- Data scientists and engineers to log, register, and compare trained models;
- Centralized model versioning and traceability;
- Access control via the ALIDA Gateway, which authenticates requests through Keycloak.

Access to MLflow is secured and routed through the platform gateway, ensuring only authorized users can interact with it. This integration supports reproducibility and traceability of ML experiments across teams and use cases.

The MLflow Tracking Server can be accessed via the ALIDA GUI via a dedicated button link. Only ALIDA users can access MLflow; as Keycloak authentication is not natively supported, controls take place on the ALIDA Gateway side so that only ALIDA users, with the appropriate JWT or API Key, can communicate with the MLflow Tracking Server. The figure below shows an example of a model logged into MLflow, also containing metrics and hyperparameters that data scientists can exploit to compare different models trained using MLflow functionalities.

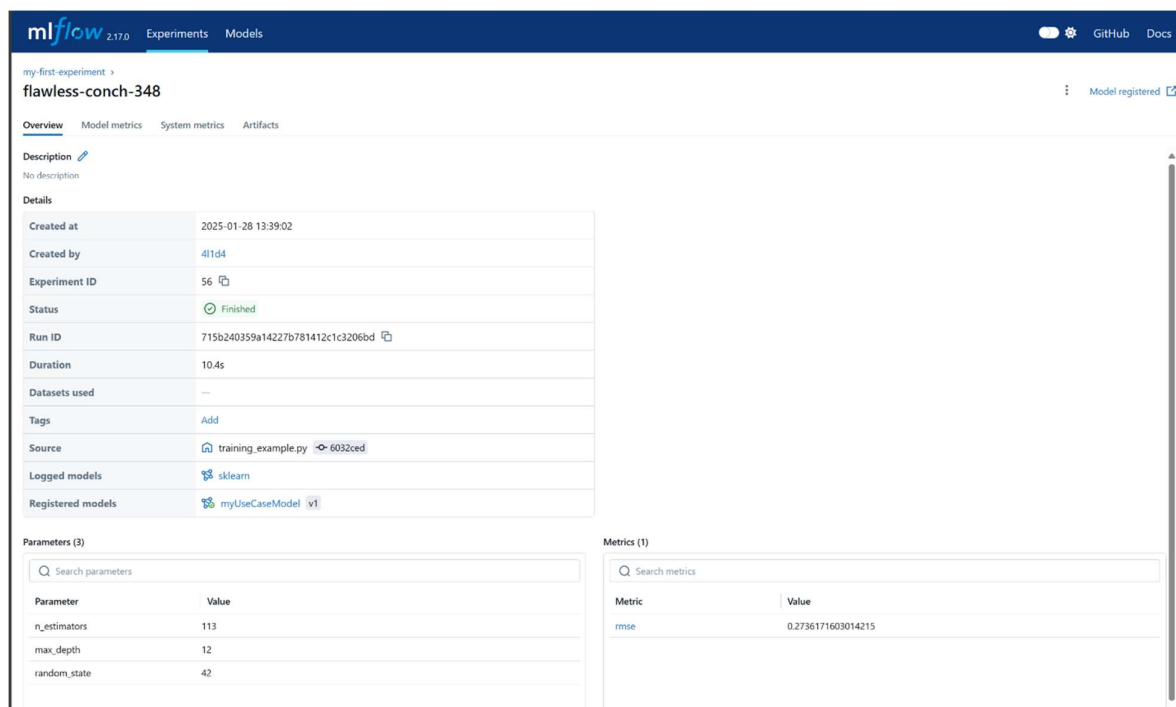


Figure 10.10 MLflow – Model experiments

#### 4.2.2.4 Seldon Core V2

Seldon Core is a key enabler of MLOps as it automates and standardizes the serving of ML models. Its integration into ALIDA facilitates scalable, framework-agnostic model deployment and monitoring. By supporting diverse backends (e.g., MLServer, Triton server), Seldon Core allows the serving infrastructure to adapt to various model types while maintaining high performance.

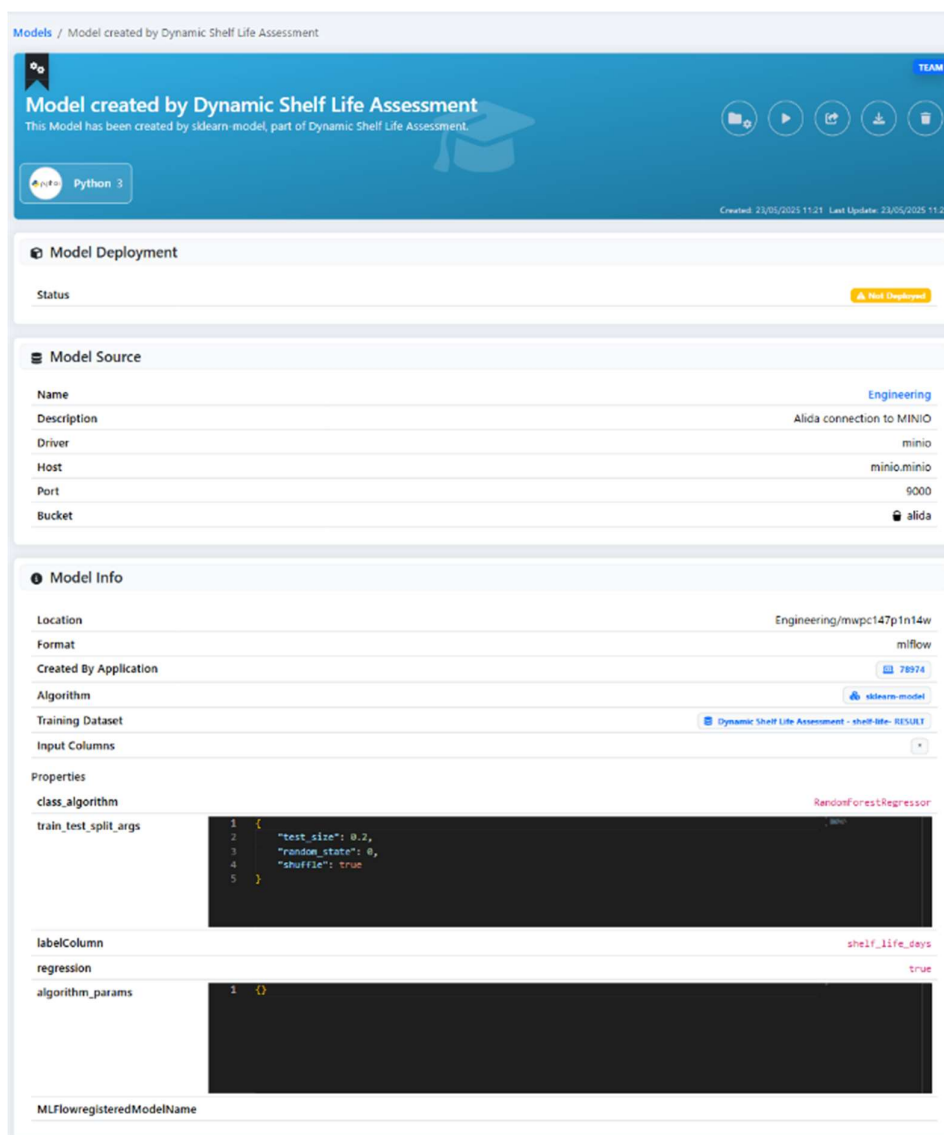
The ALIDA platform incorporates Seldon Core V2, leveraging MLServer and NVIDIA Triton inference servers for dynamic, scalable ML model serving. Key aspects of this integration include:

- Models are uploaded to MinIO and registered through the ALIDA GUI/API with associated metadata (name, description, framework type, etc.);
- Model serving is managed through ALIDA's interface, triggering the appropriate backend serving component based on the model framework;
- Served models are exposed through RESTful APIs protected by the ALIDA Gateway and JWT-based authentication via Keycloak;
- Real-time monitoring of the serving ML model status is available (e.g., deployed successfully, serving error, etc.).

Security is enforced at every layer: only users with proper roles (defined in Keycloak) can register or invoke served ML models. Role-based policies ensure that access to a given model is restricted to specific users or groups, using JWT scopes for runtime authorization.

When a model is registered or produced via training pipelines run on the ALIDA platform, it is possible to enter the ML model detail page as shown in the following figure. From this page, it is possible to get an overview of the registered ML model, such as its description, algorithm used for its training, and other useful metadata, as well as the option of downloading the model or starting it. During the model start-up, APIs are invoked under the hood towards Seldon Core to serve the selected model appropriately. Once served, the user is given a URL to contact that specific model via

appropriate REST APIs that users can invoke passing as header the JWT or API Key generated for that specific ML model. In the example in the figure, the ML model has not yet been served, so its deployment status is 'not deployed.'



The screenshot shows the 'Model created by Dynamic Shelf Life Assessment' page in the ALIDA platform. The page is divided into several sections:

- Model Deployment:** Shows the status as 'Not Deployed'.
- Model Source:** A table with the following details:
 

Name	Engineering
Description	Alida connection to MINIO
Driver	minio
Host	minio.minio
Port	9000
Bucket	alida
- Model Info:**
  - Location: Engineering/mwpc147p1n14w
  - Format: milflow
  - Created By Application: 78974
  - Algorithm: sklearn-model
  - Training Dataset: Dynamic Shelf Life Assessment - shelf-life-RESULT
  - Input Columns: (empty)
- Properties:**
  - class\_algorithm: RandomForestRegressor
  - train\_test\_split\_args: 

```
1 {
2   "test_size": 0.2,
3   "random_state": 0,
4   "shuffle": true
5 }
```
  - labelColumn: shelf\_life\_days
  - regression: true
  - algorithm\_params: 

```
1 {}
```
  - MLFlowRegisteredModelName: (empty)

Figure 11.11 ALIDA Platform – Model details

#### 4.2.2.5 ALIDA API Key Manager

In addition to JWT-based authentication provided by Keycloak, a custom component for API Key management has been developed and integrated into the ALIDA platform. This component allows users to generate and use dedicated API Keys as an alternative to the standard JWT tokens.

An API Key is a unique alphanumeric string used to authenticate a client or an application attempting to access a specific application programming interface (API). It acts as a form of identification, allowing the API provider to recognize the client making the request.

This flexibility is particularly beneficial for enabling machine-to-machine communication and for users who prefer long-lived tokens over session-based JWTs. The API Key system works in parallel with Keycloak's authentication flows and is



fully compatible with the ALIDA Gateway, ensuring secure, controlled access to served models and services. This approach provides an additional access mechanism that enhances usability without compromising security.

Through the ALIDA graphical user interface, it is possible to create API Keys associated with specific applications or deployed models. These keys can be revoked at any time or regenerated as needed, providing flexibility in access management. Although the API Key mechanism provides a lightweight form of authentication, the primary authorization system remains Keycloak. As such, the API Key is generated in a format resembling basic authentication, embedding both the unique Keycloak username and the identifier of the application or served model for which the key is being issued.

This structure allows the API Key Manager, which includes an embedded Apache server, to validate the authenticity of the received API Key. Once validated, the request proceeds to the ALIDA Gateway, which performs additional authorization checks by retrieving and evaluating the roles associated with the user linked to the provided API Key. This layered approach ensures that API access is tightly controlled, traceable to specific users and applications, and fully aligned with the role-based access control policies defined in Keycloak.

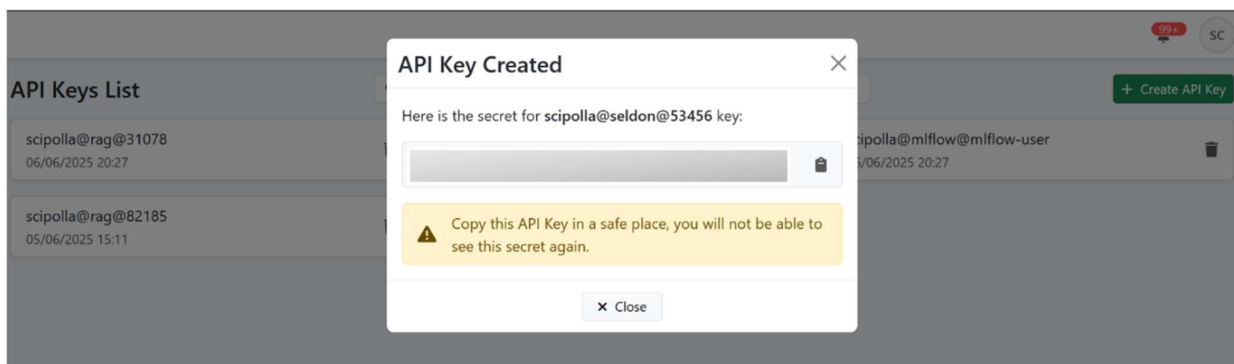


Figure 12.12 API Key Management

### 4.2.3 MLOps and CEDAR project

An instance of the ALIDA platform has been successfully deployed and released as part of the CEDAR project at the following web address: <https://alida.cedar.alidalab.it>. The platform has been extended, as previously described, to support core MLOps functionalities such as experiment tracking (via MLflow), secure model serving (via Seldon Core), and centralized identity management (via Keycloak).

Guidelines and demonstration materials have been provided to facilitate platform adoption by project partners. Today, ALIDA is actively used to deploy and expose containerised servers as secure REST APIs through a graphical service designer interface. This enables users to publish backend services rapidly and safely in a consistent and controlled manner. Initially envisioned as a flexible foundation for supporting ML pipelines, ALIDA has now fully integrated core components for MLOps, including MLflow for model tracking and Seldon Core V2 for serving.

As part of its application in CEDAR:

- ML models can be trained, tracked, and versioned using MLflow with MinIO as the backend store;
- Deployed models are served securely through Seldon Core and exposed via REST APIs protected by Keycloak-based JWT authentication;
- Access to APIs and models is strictly enforced via role-based policies managed through Keycloak;
- Retraining and monitoring capabilities are currently being extended with support for pipeline automation and KPI-driven workflows.

ALIDA has also been used to deploy:

- The Correlax API service application, developed from WP4's T4.4;
- The Knowledge Graph (KG) Client service application developed by ENG to interact with the Knowledge Graph provided in T3.1.

The KG Client has been also extended to expose custom endpoints, for supporting technical partners, such as:

- GET /graphs/{graphName}/tenders/{tenderId}/ego\_graph: to return a tender and its substructure;
- GET /graphs/{graphName}: to return the full knowledge graph;
- GET /tenders/bidder-contacts: to retrieve contacts associated with bidders of given tenders.

These services help other WPs retrieve structured knowledge graph data in a secure and controlled fashion.

Future extensions of the ALIDA platform in the context of MLOps will focus on enabling automatic retraining of ML pipelines based on monitored KPIs. This could include:

- Integration with drift detection tools to monitor data and model performance;
- Triggering retraining workflows automatically using orchestration frameworks;
- Leveraging notifications and alerting systems to inform users of the retraining process.

Additional future development/integrations work will focus on the intelligent orchestrator for resource optimization, and energy-aware policy enforcement in the Kubernetes infrastructure where application will be executed.

Additional future development and integration efforts will focus on the intelligent orchestrator to support resource optimization within the Kubernetes infrastructure where applications are deployed. The goal will be focus on ensuring that application workloads are managed dynamically and efficiently.

By leveraging all the integrated tools and the MLOps guidelines provided, ALIDA will deliver a wide range of benefits. It will operate as a MLOps platform capable of supporting continuous training (CT), continuous integration (CI), and continuous deployment (CD) of machine learning models. This approach ensures end-to-end lifecycle support—from secure model development and versioning to automated deployment and retraining—offering increased reliability, adaptability, and scalability in dynamic and data-driven environments.

## 5 Integration with CEDS and Data Alignment Tools

### 5.1 Overview

Integration with CEDS is the main goal of Task 3.3. More specifically, this task aims to develop dataspace connectors that align with key European initiatives like IDSA, GAIA-X, EOSC, etc., facilitating continuous data exchange and interoperability among these initiatives, to create a secure, reliable, and integrated European data network. Through the progress of Task 3.3, the goal is to establish a fully operational and tested Dataspace within the CEDAR platform. This will bring CEDAR a step closer to the vision of being compliant and integrable with the Common European Data Spaces (CEDS). The concept of CEDS is part of the European strategy to enhance data sharing across different sectors and borders within the European Union (EU). These Dataspaces are intended to create a single market for data, ensuring that data can flow freely across the EU, while remaining subject to high standards of data protection, privacy, and security. Therefore, the establishment of CEDS is important for Europe to operate in a unified manner that reflects core European values such as self-determination, privacy, transparency, security, and fair competition.

The first objective of this Task was to identify the most mature Connectors by the second quarter of 2024, focusing on those that are well-developed and suitable for further distinction. The final selected connector would serve as the foundational element for analyzing the technical specifications of compatible initiatives. The second objective is to proceed with the development of a Dataspace tailored to the needs of the CEDAR project. The ideal outcome is the establishment of a true Dataspace, capable of being operational across the three pilots of the project. This objective has also been achieved. However, the deployment leaves room for further refinement and improvements, in order to finally establish a proper Minimum Viable Dataspace (MVD) for CEDAR, with the ability to be easily integrated with CEDS across the EU.

### 5.2 Connectors Research & Final Selection

The Data Connector Report [47], published regularly by IDSA, is designed to explain data connectors. In summary, the report covers:

- **Explanation of Data Connectors:** it details what data connectors are and why they are crucial in data spaces.
- **Types of Connectors:** the report categorizes data connectors into four types: data connector frameworks, open-source generic solutions, proprietary generic solutions, and off-the-shelf data connectors or those integrated in data-related products.
- **Interoperability Requirements:** it outlines the requirements for making data connectors work together smoothly, like adhering to standards, clear specifications, and promoting semantic interoperability using specific vocabularies such as the Data Catalog Vocabulary (DCAT).
- **Visibility of Implementations:** the report showcases existing implementations, providing information on their license type, maturity, and usage cases, and tracks their evolution.
- **Learning and Enabling Interoperability:** it aims to be a learning hub for data sharing ecosystems, discussing other approaches that help in data-driven business ecosystems and promoting future alignment with IDS.
- **Additional Technologies:** additional technologies like the Gaia-X trust framework, iShare, and SOLID are listed, which aid in trustworthy data sharing.

The connectors included in the IDS Data Connector Report were filtered based on three factors. The first factor was the TRL level. Connectors below TRL 7 would not be considered, since the final dataspace implemented within CEDAR will have to be TRL 7. The High TRL connectors are the ones listed and advertised with Technology Readiness Levels between 7 and 9. The only exception was the examination of the TRUE Connector, due to its relatively high adoption rate. The second factor was the open-source nature of each connector. Although both closed-source and open-source connectors were analyzed, the final selection would be made from the open-source connectors' list. The third factor will be whether each connector has a proof-of-concept Minimum Viable Dataspace (MVD) implemented or not. An MVD implementation is crucial as it not only demonstrates the ease of deployment for a specific connector but also validates (or refutes) the Technology Readiness Level (TRL) claimed by the connector's vendor. This process would ensure that the connector's capabilities are accurately represented and reliable in practical applications. One last filter that was

examined was whether each connector is IDS Certified or not. However, this filter ultimately did not affect the final connector selection.

### 5.3 Connector Selection

Based on the evaluation and analysis of all the aforementioned dataspace connectors, two choices prevailed. The first selection was the Eclipse Dataspace Components (EDC) Connector, and connector implementations based on it. The EDC Connector aligns well with the criteria / factors defined, covering most of the required aspects, and is continuously being enhanced. It conforms to key Dataspace Initiatives such as IDSA and GAIA-X, and is actively utilized in projects like EONA-X and Catena-X. The second selection was Engineering's TRUE Connector. Although the TRUE Connector is involved in several EU research and dataspace projects, its comparatively lower TRL makes it a more challenging option relative to the EDC. However, both Connectors can be tested in order to achieve compliance with at least five (5) CEDS initiatives.

Ultimately, the choice-to-go for the needs of CEDAR was the soviety Open-Source EDC Connector [48] [49], managed by soviety GmbH. It is designed as a versatile and easy-to-use solution for data sharing among participants in data spaces. It is based on the Eclipse Dataspace Components (EDC) framework (it is an extension of the EDC Connector) and is characterized by its ready-to-use, open-source nature, which is aimed at enhancing usability with features such as usage control. The connector is available under an Apache 2.0 license and has reached a maturity level of TRL 9, indicating it is currently used in production environments.

This connector is platform agnostic, allowing for deployment on-premises, in the cloud, or other environments, and is classified as a self-hosted service. It supports access control through mechanisms like Basic Auth and API key, with usage control policies including Connector Restriction and Time Interval implemented to manage data interactions securely and efficiently. The soviety Open-Source EDC Connector utilizes the Dataspace Protocol (HTTPS) for communication and supports out-of-band data transfer protocols. It features a graphical user interface that facilitates ease of use for users, management, and administration. The connector also incorporates centralized DAPS (X.509) and mock IAM for identity management.

### 5.4 The CEDAR Minimum Viable Dataspace (MVD)

The thorough research conducted led to the development of CEDAR's working Minimum Viable Dataspace (MVD) deployment. It is designed to provide a simple and functional core infrastructure for secure and sovereign data exchange between the pilots of the project. However, since it is based on the EDC connector (which complies with several Dataspace initiatives), it is also expandable for integration with other CEDS.

The CEDAR MVD includes three main components:

- A Dynamic Attribute Provisioning Service (DAPS), based on Keycloak, which enables authentication and authorization.
- A NGINX reverse proxy server with SSL certification to secure DAPS (given the fact that a DNS hostname is available).
- The Dataspace Connector, which is an extended version of soviety's EDC Connector.

The deployment process involves setting up one instance of the DAPS Keycloak module, the Dataspace Connector, and the NGINX reverse proxy with SSL certification via certbot [50]. However, to establish a fully functional Minimum Viable Dataspace, at least two Dataspace Connectors must be deployed on separate machines (or premises). Each connector must also be registered as a new client in DAPS, in order to enable the authentication & authorization mechanism for all participants of the Dataspace. This setup leads to a minimal but operational environment, where data providers and consumers can securely exchange data under controlled (and supervised) conditions. This approach finally enables a straightforward initial setup that can be expanded and iterated upon as needed.

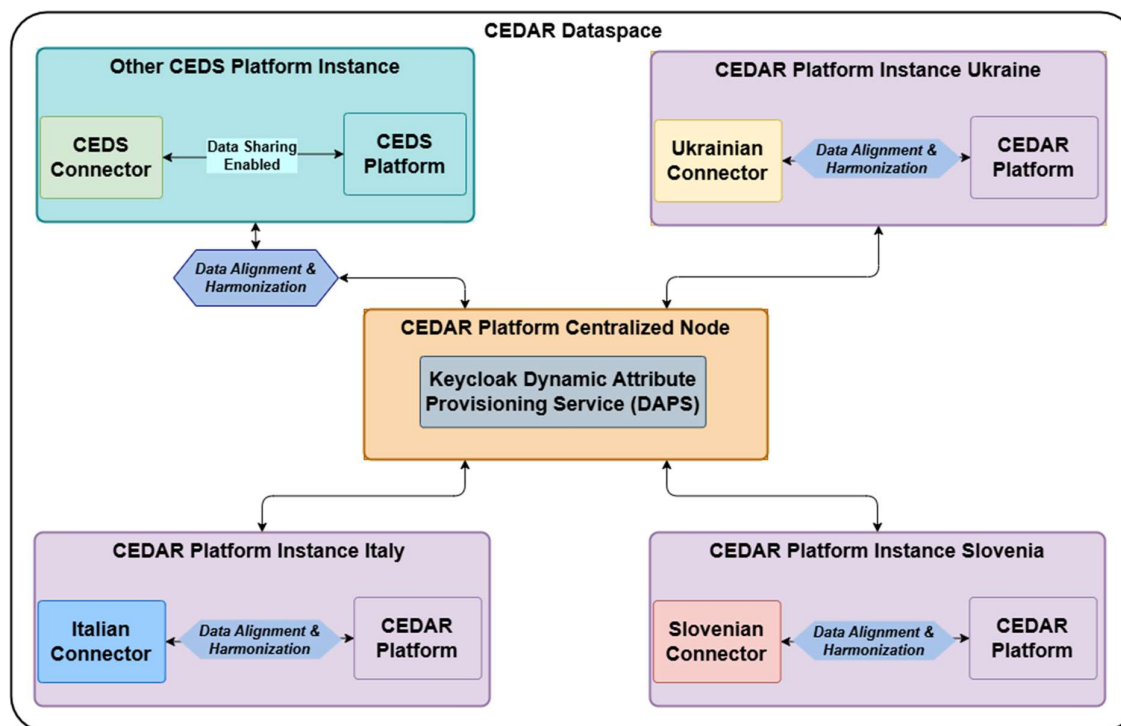


Figure 13.13 The CEDAR Minimum Viable Dataspace (MVD) Architectural Overview

#### 5.4.1 Connector Deployment & Configuration

To ensure secure communication within the Dataspace, especially with key components such as the DAPS, it is first needed to set up digital certificates for every new Connector, acting as a new Dataspace participant. This involves generating a trusted root certificate and then creating a unique certificate for the Connector itself. These certificates act as digital IDs, allowing the system to verify who is who, and thus exchange data securely. The approach followed is a standard certificate generation process, using OpenSSL tools, which are widely adopted for security-related operations. A set of careful adjustments are required to the configuration, in order to meet the specific requirements for identity verification. These adjustments involve the proper inclusion of certain fields needed for OAuth-based authentication.

Once the certificates are in place, they are being bundled into a secure keystore format that the sovity EDC Connector's software can recognize and trust during operation. This step is crucial, since it allows the Connector to prove its identity in the Dataspace (passing through Keycloak DAPS). Therefore, it ensures secure and trusted interactions with other participants in the CEDAR dataspace.

With security taken care of, the next step is to deploy the Dataspace Connector, which – as previously mentioned and explained - is based on an extended open-source version of the Eclipse Dataspace Connector (EDC), maintained and adapted by sovity. The Connector software acts as a bridge between the pilot's infrastructure and the broader dataspace, enabling controlled and standards-compliant data exchange. Instead of starting from scratch, the pre-configured sovity repository has been leveraged, taking into account all the necessary files and configurations. From there, the deployment has been customized to match the specific needs of the CEDAR platform's environment. This includes dynamically defining each pilot's machine's public IP address, pre-registering other connectors (if existing in the Dataspace), and pointing to the correct DAPS service that handles identity verification, also providing the credentials generated. It is also ensured that each Connector instance has a unique name in the dataspace — an important requirement to avoid conflicts.

Once these values are properly set, the Connector software can be safely deployed using Docker. This approach is on par with other common container-based deployment methods. The deployment is straightforward: After running the

Connector's containers ( i) a personal PostgreSQL for metadata storage, ii) a middleware software, iii) the Connector's UI ), the system becomes accessible via its User Interface, allowing the pilot's end-user to interact with it, monitor it, and begin managing data transfers within the CEDAR Dataspace.

The Docker Compose file for the deployment of the CEDAR Dataspace Connector can be seen below:

```

1. version: "3.8"
2. services:
3.   edc-ui:
4.     image: ghcr.io/sovity/edc-ui:latest
5.     ports:
6.       - '11000:8080'
7.       - '11015:5005'
8.     environment:
9.       EDC_UI_ACTIVE_PROFILE: sovity-open-source
10.      EDC_UI_CONFIG_URL: edc-ui-config
11.      EDC_UI_MANAGEMENT_API_URL: http://<INSERT_PUBLIC_IP>:11002/api/management
12.      EDC_UI_MANAGEMENT_API_KEY: ApiKeyDefaultValue
13.      EDC_UI_CATALOG_URLS: http://<INSERT_PUBLIC_IP>:11003/api/dsp,
http://<INSERT_OTHER_CONNECTORS_PUBLIC_IP_IF_EXISTS>:11003/api/dsp,
http://<INSERT_OTHER_CONNECTORS_PUBLIC_IP_IF_EXISTS>:11003/api/dsp
14.      EDC_UI_MANAGEMENT_API_URL_SHOWN_IN_DASHBOARD:
http://<INSERT_PUBLIC_IP>:11002/control/api/management
15.      NGINX_ACCESS_LOG: off
16.   edc:
17.     image: ghcr.io/sovity/edc-ce:10.4.2
18.     #image: ghcr.io/sovity/edc-dev:10.4.2
19.     volumes:
20.       - ./keystore.jks:/app/keystore.jks
21.     depends_on:
22.       postgresql:
23.         condition: service_healthy
24.     environment:
25.       MY_EDC_PARTICIPANT_ID: "<UNIQUE-PARTICIPANT-ID>"
26.       MY_EDC_TITLE: "CEDAR Testing Connector"
27.       MY_EDC_DESCRIPTION: "CEDAR Connector based on the Sovity EDC Connector Community Edition"
28.       MY_EDC_CURATOR_URL: "https://cedar-heu-project.eu/"
29.       MY_EDC_CURATOR_NAME: "CEDAR EU PROJECT"
30.       MY_EDC_MAINTAINER_URL: "https://netcompany.com"
31.       MY_EDC_MAINTAINER_NAME: "Netcompany-Intrasoft"
32.
33.       MY_EDC_FQDN: "edc-IRE"
34.       EDC_API_AUTH_KEY: ApiKeyDefaultValue
35.
36.       MY_EDC_JDBC_URL: jdbc:postgresql://postgresql:5432/edc
37.       MY_EDC_JDBC_USER: edc
38.       MY_EDC_JDBC_PASSWORD: edc
39.
40.       MY_EDC_PROTOCOL: "http://"
41.       EDC_DSP_CALLBACK_ADDRESS: http://<INSERT_PUBLIC_IP>:11003/api/dsp
42.       EDC_WEB_REST_CORS_ENABLED: 'true'
43.       EDC_WEB_REST_CORS_HEADERS: 'origin,content-type,accept,authorization,X-Api-Key'
44.       EDC_WEB_REST_CORS_ORIGINS: '*'
45.       # DAPS config
46.       EDC_LOG_LEVEL: DEBUG
47.       EDC_OAUTH_TOKEN_URL: 'https://<DAPS_PUBLIC_IP_OR_DNS_HOSTNAME>/realms/DAPS/protocol/openid-
connect/token'
48.       EDC_OAUTH_PROVIDER_AUDIENCE:
'https://<DAPS_PUBLIC_IP_OR_DNS_HOSTNAME>/realms/DAPS/protocol/openid-connect/token'
49.       EDC_OAUTH_PROVIDER_JWKS_URL:
'https://<DAPS_PUBLIC_IP_OR_DNS_HOSTNAME>/realms/DAPS/protocol/openid-connect/certs'
50.       # DAPS Credentials
51.       EDC_OAUTH_CLIENT_ID: '<INSERT_AKI/SKI_VALUE>'
52.       # Keystore Config
53.       EDC_KEYSTORE: '/app/keystore.jks'

```

```

54.     EDC_KEYSTORE_PASSWORD: 'password'
55.     EDC_OAUTH_CERTIFICATE_ALIAS: "1"
56.     EDC_OAUTH_PRIVATE_KEY_ALIAS: "1"
57.     ports:
58.       - '11001:11001'
59.       - '11002:11002'
60.       - '11003:11003'
61.       - '11004:11004'
62.       - '11005:5005'
63.     postgresql:
64.       image: postgres:15.3
65.       restart: always
66.       environment:
67.         POSTGRES_USER: edc
68.         POSTGRES_PASSWORD: edc
69.         POSTGRES_DATABASE: edc
70.         #POSTGRESQL_USERNAME: edc
71.         #POSTGRESQL_PASSWORD: edc
72.         #POSTGRESQL_DATABASE: edc
73.       ports:
74.       - '5432:5432'
75.       volumes:
76.       - 'postgresql:/bitnami/postgresql'
77.     healthcheck:
78.       test: [ "CMD-SHELL", "pg_isready -U edc" ]
79.       interval: 1s
80.       timeout: 5s
81.       retries: 10
82.     volumes:
83.     postgresql:
84.       driver: local
85.

```

The default live User Interface of the connector provided by soivity can be seen below:

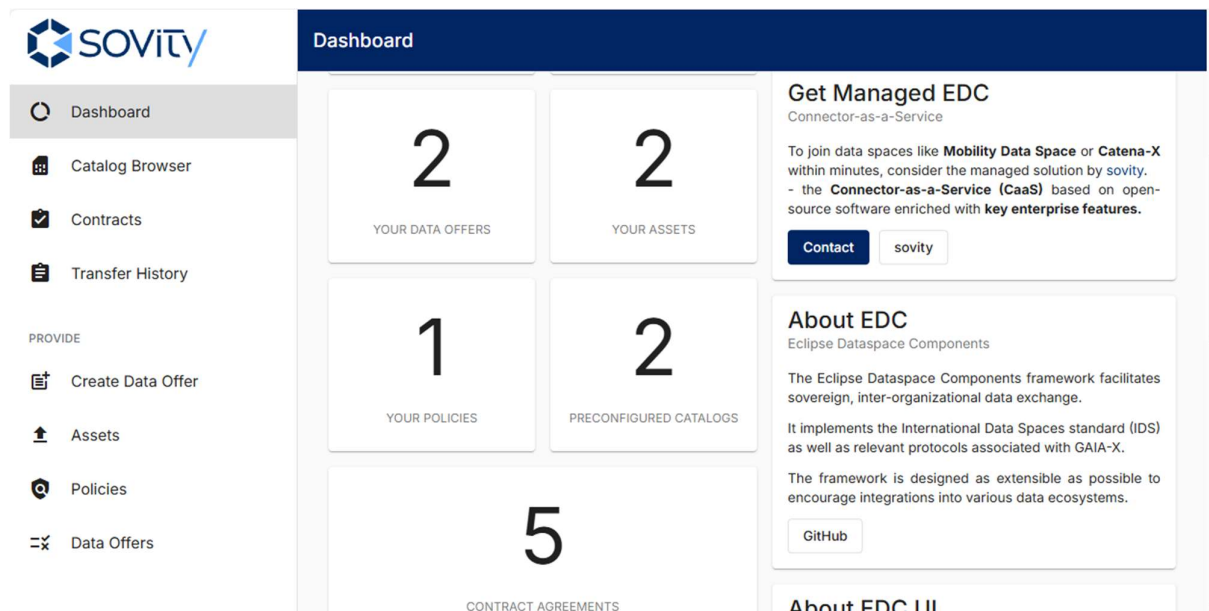


Figure 14. 14Screenshot from the connector's UI in the home page



## Catalog Browser

Search catalog

Connector Endpoints

1 - 3 of 3

Already using (2): http://88.198.203.27:11003/...



27newdata

CEDAR EU PROJECT

No Description



167datas

CEDAR EU PROJECT

No Description



167newdata

CEDAR EU PROJECT

No Description

Figure 15.15 Screenshot from the connector's UI in the Catalog Browser, exploring assets from other Dataspace participants

Figure 14. and Figure 15. show the interface of the soivity Dataspace Connector, on which the CEDAR MVD is being based on. Figure 14. depicts the UI's homepage, while Figure 15. presents the *Catalog Browser*. The latter view demonstrates how the CEDAR pilots — Italy, Slovenia, and Ukraine — can make their datasets part of a real, operational data space. Each of these pilots acts as both a provider and a consumer. They can publish their own data, and at the same time discover and use data from others. Through this browser, users can explore available data offers, using the search bar. They can also browse by endpoint. The pilots used this interface to locate datasets like the dummy “27newdata” and “167datas” ones (as shown in the Figure), and begin testing real data sharing scenarios. During this phase, user feedback played a key role. They highlighted the clarity of separating shared and received assets, and how helpful the search functionality was. This interaction and experience have shaped the steps for development of a customized CEDAR Connector UI (as presented in subsection 5.4.3 later on), tailored to better support real-world needs. The user journey leading up to this point, by M18, includes exposing the pilots to the UI, receiving their feedback, and preparing selected assets of theirs for discovery across the dataspace.

### 5.4.2 Enabling the Dynamic Attribute Provisioning Service (DAPS)

As the main tool that enables secure 2identity and access management within data spaces, a Dynamic Attribute Provisioning Service (DAPS) is being deployed, based on the open-source Keycloak platform. This DAPS implementation was initially adapted from soivity's open-source distribution. This approach aims to offer flexibility and integration potential within the broader IDS (International Data Spaces) architecture.

The first step is to set up the DAPS environment using Docker. The deployment is carefully configured to ensure it runs under a trusted hostname. This hostname is important, since it acts as the digital identity of the DAPS itself, allowing secure communication with other services (mainly connectors / participants in the Dataspace). Passwords and other basic settings are also adjusted to match the internal security policies. The system is then launched using Docker, allowing for reproducible and manageable deployment across different environments, in case an engineer wishes to replicate the MVD architecture.

As already outlined, security is a critical factor in this deployment. For DAPS to function properly, especially in scenarios involving trust-sensitive operations like token issuance, communications need to be encrypted. This means using HTTPS rather than HTTP. To achieve this, a secure proxy layer has been configured around DAPS, using NGINX, a widely adopted web server. Digital certificates are also being set up, using Let's Encrypt through Certbot. This way, it is ensured that the connection to DAPS is encrypted and recognized as trustworthy by external systems. This guarantees data privacy



between Dataspace participants' communication, confirming the authenticity of the deployment to connected connectors.

Once DAPS is secured and operational, it is then configured to interact with the connectors deployed in the pilots' premises. As mentioned before, these are the components that enable pilots to exchange data securely and under specific usage policies. A new identity domain (a "realm") within DAPS is established. Then, each connector that will participate in the Dataspace has to be carefully registered. Each connector is assigned a unique identifier derived from its cryptographic keys. These identifiers allow DAPS to issue authorization tokens that are trusted and verified by others in the data space.

It should be noted that special care is taken to embed the right attributes and claims into these tokens. This ensures each participant can prove who they are, what they are allowed to access, and under which terms.

To validate the integration, two Dataspace connectors are deployed, one authorized through DAPS and one not. The expected scenario is for the unauthorized connector not to be able to see or access the data shared by the authorised one. However, once proper configuration (for registration with DAPS) is performed to the second connector as well, secure communication and data exchange between the two becomes possible. This confirms that the DAPS is correctly enforcing access control and identity verification. With this, a **Minimum Viable Dataspace** is successfully established. It is a functioning, secure environment where data exchange only happens between authenticated and trusted entities.

The Docker Compose file for the deployment of the Keycloak DAPS can be examined below:

```
1. services:
2.   keycloak:
3.     build: .
4.     restart: always
5.     environment:
6.       KC_DB: postgres
7.       KC_DB_URL_HOST: postgres
8.       KC_DB_URL_DATABASE: keycloak
9.       KC_DB_SCHEMA: public
10.      KC_DB_USERNAME: keycloak
11.      KC_DB_PASSWORD: ${POSTGRES_PASSWORD}
12.      KC_HOSTNAME: <DAPS_IP_OR_HOSTNAME>
13.      KC_PROXY: "edge"
14.      KC_HTTP_ENABLED: "true"
15.      KEYCLOAK_ADMIN: admin
16.      KEYCLOAK_ADMIN_PASSWORD: ${KEYCLOAK_ADMIN_PASSWORD}
17.     ports:
18.       - 8080:8080
19.   postgres:
20.     image: docker.io/library/postgres:16
21.     restart: always
22.     environment:
23.       POSTGRES_USER: keycloak
24.       POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
25.     volumes:
26.       - postgres_data:/var/lib/postgresql/data
27.
28. volumes:
29.   postgres_data:
30.
```

The live User Interface of Keycloak DAPS can be viewed below:

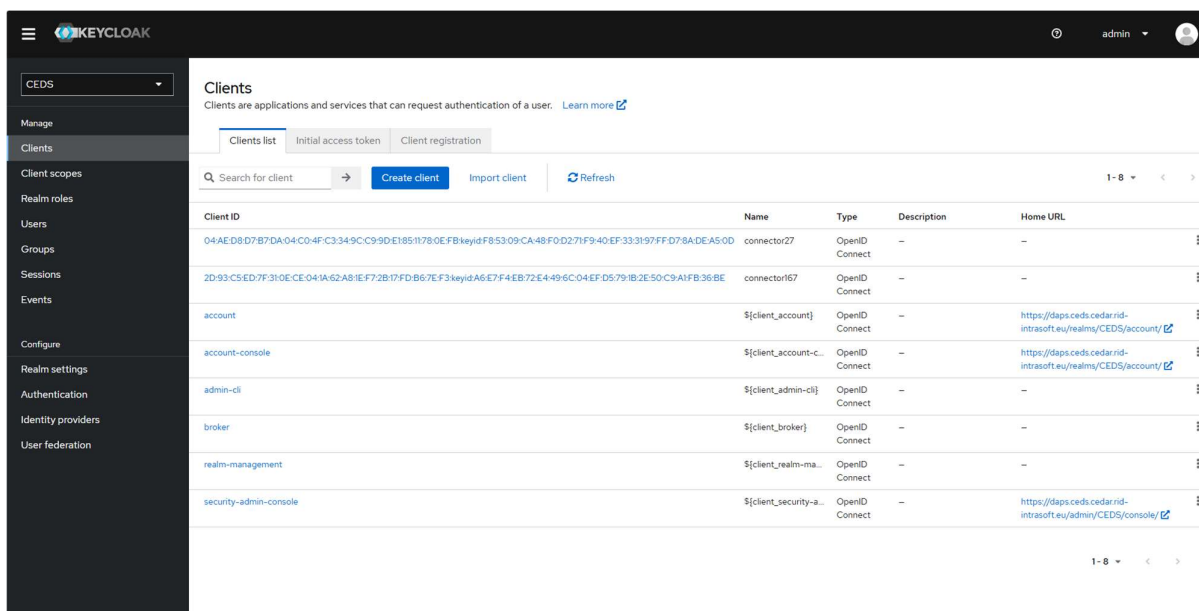


Figure 16.16 Screenshot from Keycloak DAPS's UI, listing two connectors as Clients in the 'CEDS' Realm

### 5.4.3 The CEDAR Connector UI

With the successful establishment of the CEDAR MVD, the efforts shifted towards testing the secure connection and data exchange between the participants of the Dataspace. Refinements to the MVD will continue. The main focus moved to the development of a new User Interface, distinct from the default Connector UI provided by soivity. This is an important decision, since it will make the UI more user-friendly for the pilots of CEDAR and will act as another action point that will separate this implementation from the core Connector infrastructure of soivity. Modifications to the proposed setup have already been made. As time progresses, changes similar to the new UI will mark the completion of a steady transition from the default soivity Connector to a proper CEDAR-oriented Connector deployment.

As previously outlined, the CEDAR Connector UI is under development. Screenshots of the current phase of the interface can be seen in the figures below:

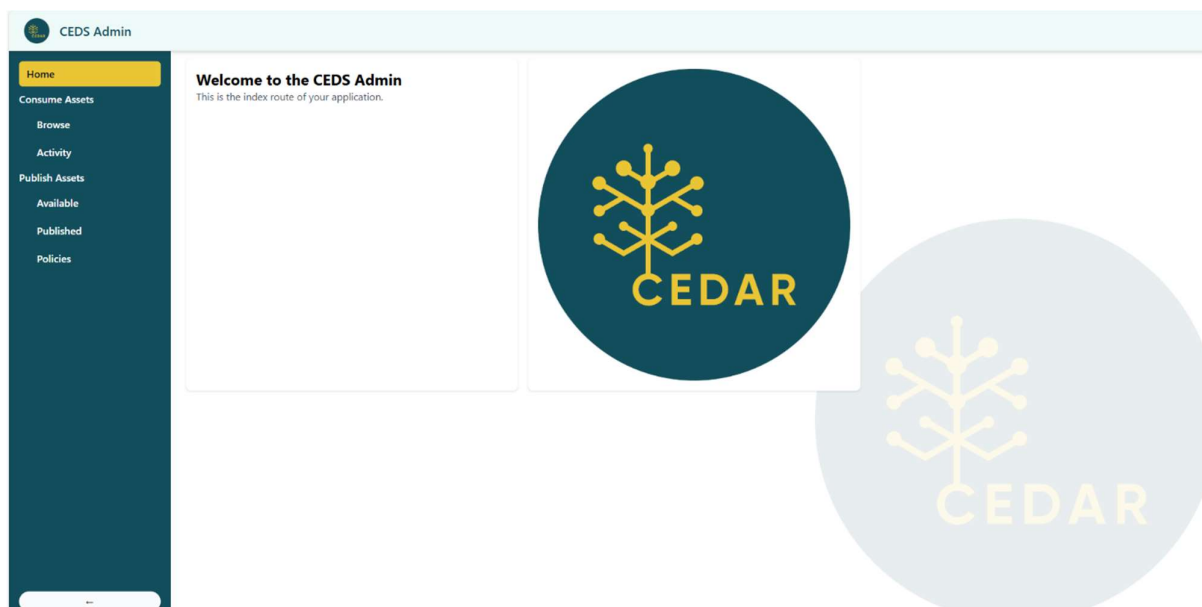


Figure 17.17 Screenshot from the - still under development - User Interface's home page of the CEDAR Connector

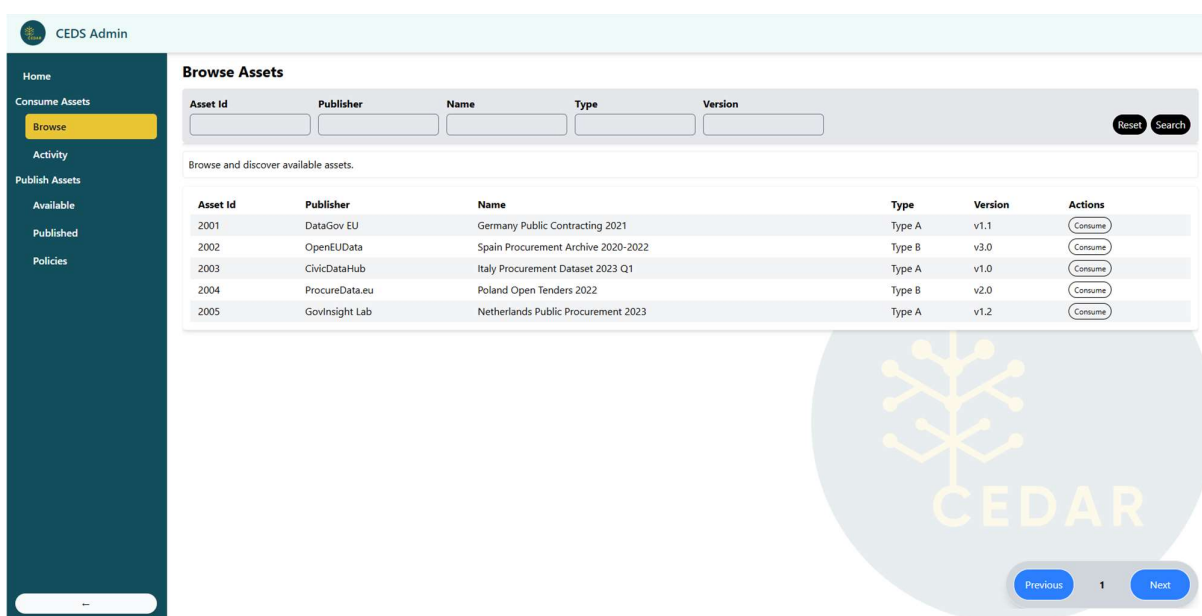


Figure 18. 18Screenshot from the - under development - User Interface's Catalog Browser of the CEDAR Connector, using mocked data

## 5.5 Autonomous Data Alignment Tool

Another challenge that CEDAR will face with relation to the different data stored in the CEDS goes back specifically to the fact that multiple data sources will be used. This means that at any given moment there will be a need to align the different data stored in the CEDS so the tools of CEDAR can perform different analysis and extract different indicators. This puts CEDAR in a position in which a new need arises, going beyond the connection of different data spaces and stepping into the realm of data alignment. This means that CEDAR has the need to establish a means to ensure that the data stored in the CEDS is actually digestible and processable by its different components. On a small scale, this is doable by the use of a detailed data model or ontology. Standardization bodies usually create data models that pertain to their

domain. These data models can be used to solve the data alignment issues but, more often than not, these are the result of extensive labor by experts and are handcrafted and tailored. . The work of the experts is to provide a solid common ground upon which data interoperability is established by providing a robust solution that will ensure the correct operation of the components that will be using the tools. Nonetheless, this is almost impossible to do when the amount of data reaches a certain threshold. Even more, it is not possible to do it in an acceptable time frame, as the effort to develop a consistent and comprehensive one usually requires a considerable amount of time as the experts need to get to understand both the requirements and the domain.

With all the above, producing a consistent data model that is able to cover the myriads of data is a challenge that needs to be tackled for the success of CEDAR, as the data from the CEDS needs to be aligned. Nonetheless, the main issue is that this alignment needs to happen almost in real time. By contraposition, the time required to generate a data model via the traditional methods will not be enough. For that matter, an Autonomous Data Alignment (ADA) tool is being developed. The main goal of this tool is to be able to process the different data that is being shared within the CEDS and provide a data model that will encompass, collect, and align all the data that is being shared. For that matter, the tool is driven by the use of a Large Language Model (LLM). This LLM is able to capture the different metadata that characterized the data stored in the different CEDS. Once the LLM has captured the metadata, it is compared with the data model that has been developed as the ground truth for the pilots and use-cases of CEDAR. Afterwards, the LLM is able to integrate these metadata into the existing data model, expanding or reducing it as needed. This leads to having a tailored data model for every situation in which data is being shared within the CEDS. The current state of the development of the tool is situated in a stage in which it has been tested in a laboratory environment while fed with real data. This situates the tool at a TRL 4, meaning that the current development is on the route to providing full support for CEDAR in the medium term.

## 6 Cybersecurity

As outlined in the deliverable D3.1, the cybersecurity task in CEDAR focuses on ensuring the reliability and robustness of all developed software artefacts, datasets, and deployment environments. This is achieved through a **continuous cybersecurity risk assessment** (discussed in Section 6.1) and the adoption of appropriate countermeasures. These include the development of a state-of-the-art **network intrusion detection system** (whose progress is discussed in Section 6.2) and **penetration testing** activities, which have been presented in D3.1, that will take place during the initial piloting cycle and will be reported in subsequent WP3 (and/or WP5) deliverables.

### 6.1 Cybersecurity Risk Assessment

The CEDAR cybersecurity risk assessment follows an asset-based approach, in which we systematically identify key assets, threats, vulnerabilities, and associated risks. This is done across three dimensions: (1) inter-spection (data flows and system-wide interactions), (2) intra-spection (component-level risks), and (3) en-spection (deployment environments and use-case contexts).

The initial **inter-spection results**, including architectural risks and countermeasures, were documented in deliverables D3.1 and D1.2. Since then, we have extended the risk assessment to include intra-spection, which focused on the security posture of individual CEDAR components and their internal configurations, dependencies, and functionalities. This second layer of analysis did not reveal any significant new risks or lead to the identification of additional countermeasures beyond those already defined and prioritised in deliverable D1.2. However, it did confirm the relevance and necessity of the initially defined **non-functional and data-related requirements** (NFRs and NDRs, respectively) for the implementation of the necessary countermeasures.

The next step involved defining an implementation and review plan for the defined countermeasures / requirements. The **implementation plan** covers relevant technical or procedural steps that the component developers follow to incorporate each countermeasure / requirement. This includes code-level changes (e.g., secure coding practices), configuration requirements (e.g., encryption settings), and process updates (e.g., patch management cycles). The **review plan** is essentially a validation strategy that covers, for example, security testing (e.g., penetration testing), compliance checks, documentation audits, or usability testing (for user-facing artefacts). Reviews will be aligned with the overall project implementation and integration plan.

To simplify these next steps for implementation and review, the countermeasures / requirements have been effectively **grouped into functional domains** based on their focus areas and technical relevance. This helps to streamline responsibility assignment, alignment with development workflows, and validation procedures. The groupings and their main **implementation approach and review methods** are presented below.

Grouping	Data Protection
<b>Focus</b>	Protecting data across its lifecycle
<b>Linked countermeasures / requirements</b>	NFR01 – Data Encryption (S) NFR02 – Data Anonymisation (M) NFR03 – Data Minimisation (M) NFR10 – Access Control & User Authentication (M) NDR14 – Personal Data Rectification & Erasure (M) NDR15 – Accuracy (M) NDR16 – Storage Limitation (M) NDR17 – Storage Location (S)
<b>Implementation leads</b>	Data engineers, software developers.
<b>Implementation approach</b>	<ul style="list-style-type: none"> <li>Apply reliable encryption for data at rest (e.g., AES-256).</li> <li>Apply reliable encryption for data in transit (e.g., TLS1.3).</li> <li><b>MUST:</b> Collect and store only data (attributes) required for the analysis defined in the use cases (as per the defined indicators).</li> <li><b>MUST:</b> Continuously review and ensure data quality.</li> <li><b>MUST:</b> Enforce role-based access control.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>MUST:</b> Enforce multi-factor authentication.</li> <li>• <b>MUST:</b> Regularly review permissions settings and assigned roles.</li> </ul> <p>For cases where <b>personal data</b> is processed:</p> <ul style="list-style-type: none"> <li>• <b>MUST:</b> Integrate data anonymisation and data masking components developed in CEDAR that rely on state-of-the-art techniques.</li> <li>• <b>MUST:</b> Enable rectification or erasure of personal data based on data subject requests.</li> <li>• <b>MUST:</b> Establish a data retention policy.</li> <li>• Store data in ISO/IEC 27001-certified environments within the EEA or in a country with an adequate level of protection of personal data.</li> </ul>
<b>Review methods</b>	Penetration tests, data model reviews, data checks, sample user request tests, log audits, IP geolocation checks.

Grouping	Software Security
<b>Focus</b>	Ensuring secure, trustworthy, robust, maintainable, and compliant software from development to deployment.
<b>Linked countermeasures / requirements</b>	NFR04 – Secure Coding Practices (S) NFR05 – Automated Testing (M) NFR07 – Configuration Management (M) NFR08 – Patch Management (S) NDR18 – Robustness of AI models (M) NDR21 – Maintainability & Modularity (M)
<b>Implementation leads</b>	Software developers, data scientists, integration leads.
<b>Implementation approach</b>	<ul style="list-style-type: none"> <li>• Follow the latest version of the OWASP secure coding practices [51].</li> <li>• Regularly check for vulnerabilities and apply patches and updates.</li> <li>• <b>MUST:</b> Design a modular architecture and clear interfaces.</li> <li>• <b>MUST:</b> Use a CI/CD solution that enables continuous unit and integration testing.</li> <li>• <b>MUST:</b> Use a CI/CD solution that enables management of consistent and secure configurations across various environments.</li> <li>• <b>MUST:</b> Ensure adequate data quality with cleaning, augmentation, and elimination of bias.</li> <li>• <b>MUST:</b> Ensure robustness of AI models with adequate data quality, using adversarial training examples, cross validation, and explainability.</li> </ul>
<b>Review methods</b>	Code reviews, vulnerability scans, design checks, CI/CD logs audit, ML model validation.

Grouping	System Monitoring
<b>Focus</b>	Real-time awareness, traceability, and response to incidents.
<b>Linked countermeasures / requirements</b>	NFR06 – Network Monitoring (M) NFR09 – Logging & Auditing (S) NDR20 – Reliability & Availability (S)
<b>Implementation leads</b>	Software developers, integration leads.
<b>Implementation approach</b>	<ul style="list-style-type: none"> <li>• <b>MUST:</b> Integrate solutions for continuous monitoring of the network traffic (e.g., a network IDS).</li> <li>• Implement centralised logging.</li> <li>• Regularly review system logs.</li> <li>• Define alert rules for suspicious patterns and downtime events.</li> <li>• Set up automatic health checks and service restarts.</li> </ul>
<b>Review methods</b>	Log reviews, alert policy reviews, uptime reviews.

Grouping	Governance
<b>Focus</b>	Human factors, institutional controls, and training.
<b>Linked countermeasures / requirements</b>	NFR11 – Policies (M) NFR12 – Compliance (M) NFR13 – Human Competence (S)
<b>Implementation leads</b>	Pilot leads with end users, ethics and legal partners.
<b>Implementation approach</b>	<ul style="list-style-type: none"> <li><b>MUST:</b> Establish information security and data protection policies based on relevant / applicable regulations.</li> <li>Establish training modules and run training sessions with end users.</li> </ul>
<b>Review methods</b>	Policy checks, training records reviews.

Grouping	Usability
<b>Focus</b>	End user experience, cross-system operation, and efficiency.
<b>Linked countermeasures / requirements</b>	NDR19 – Ease of Use & Usability (S) NDR22 – Performance Efficiency & Response Time (S) NDR23 – Compatibility and Interoperability (M)
<b>Implementation leads</b>	UI designers, software developers, integration leads.
<b>Implementation approach</b>	<ul style="list-style-type: none"> <li>Design UIs based on user interviews.</li> <li>Ensure that the user actions and requests are responded to within predefined acceptable time limits (define thresholds).</li> <li><b>MUST:</b> Define clear interfaces.</li> </ul>
<b>Review methods</b>	User acceptance tests, performance tests, interoperability tests.

## 6.2 Real-Time Network Intrusion Detection System

To detect malicious attacks against CEDAR's communication network infrastructure, a Network Intrusion Detection System (NIDS) is being developed by extending CEA's SIGMO-IDS tool [52] Figure19.

SIGMO-IDS utilizes online unsupervised machine learning (ML) to detect anomalies in the network traffic that could be attributed to network attacks [53] [54].

In its original design, SIGMO-IDS utilizes a multi-probe architecture as seen in Figure 19. Each probe runs as a separate ML agent, performing an independent network intrusion detection operation.

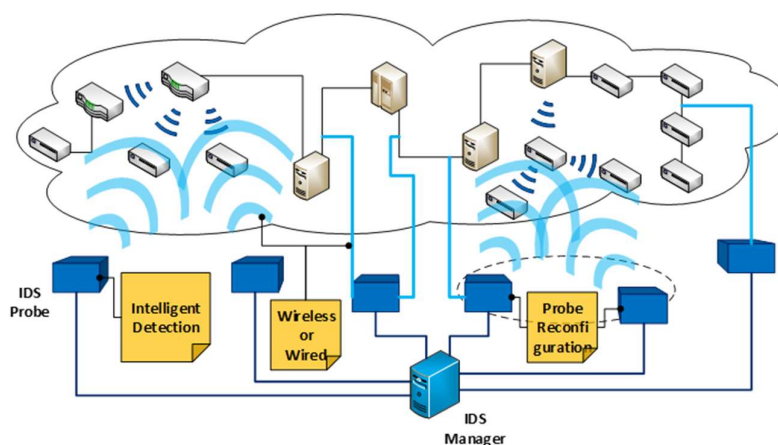


Figure1919. Sigmo-IDS' s High-Level Multi-Probe Architecture

In CEDAR's NIDS implementation, each probe is decoupled in two components: a scanner responsible for the collection of traffic metadata, and an AI engine responsible for intrusion detection and alert forwarding. This decoupling facilitates the integration of SIGMO-IDS within CEDAR's CI/CD platform by enabling the deployment of a scanner alongside each critical component.

A scanner is lightweight and can be easily integrated with other components. It has minimal requirements and can be easily containerized to run on virtually any host.

In its current development state, the scanner can be deployed with network administration capabilities and collects network traffic from a specified network interface. The monitored traffic is aggregated into traffic dumps and forwarded to the CEDAR messaging layer at a constant rate.

The traffic information is retrieved from the messaging layer and processed by the AI engine, which runs as a separate component. The AI engine works in three phases: training, calibration, and detection. During the training phase, the unsupervised ML model learns patterns from regular traffic. The AI engine developed within CEDAR uses an autoencoder-based architecture that compresses and reconstructs network traffic.

The calibration phase consists in the automated generation of a detection rule for classifying regular and anomalous traffic. This is accomplished by selecting a threshold based on the statistical distribution of the reconstruction error.

Once training and calibration are completed, the AI engine switches to "detection mode", in which traffic is iteratively reconstructed and compared against the threshold to identify anomalies. Anomalies are then turned into alerts characterized by a timestamp, source and destination ip, protocol, and anomaly score.

Whenever an alert is raised, it is automatically forwarded to the messaging layer to be ingested by other components.

The NIDS has also been integrated with a simple web interface that allows authenticated users to monitor the traffic analysed by the AI engine and visualize the anomaly scores in real time.



## 7 Conclusion

Since the initial work documented in D3.1, WP3 has undergone solid progress across all key areas. Key activities of the reporting period are highlighted as follows:

- Finetuning DataOps and MLOps for CEDAR datasets;
- Deploying and instantiating the first release of CEDAR Minimum Viable Data Space; and
- Preparing the complete set of technology offerings for pilot execution and validation.

Building on the dataset identification work of WP2, the approaches to data collection have been refined, moving from initial analysis to more concrete implementations. This has had an impact on the majority of components of WP3.

The DataOps infrastructure has evolved further, continuing to follow the CRISP-DM model. Workflows that support large-scale, reliable data processing have been integrated with DataOps. In parallel, MLOps development has advanced to support smoother data preparation, model deployment, and monitoring, making the software more robust and adaptable.

Integration with the CEDS has also moved forward. A CEDAR Minimum Viable Dataspace has been established and deployed, enabling interconnection between the three pilots of the project, ensuring secure data exchange.

Finally, cybersecurity remains a core focus. The processes included have been further analyzed, so that the CEDAR components meet the security requirements set out in WP1 and are validated through targeted testing.

The work presented in this deliverable confirms that WP3 is on track, with important groundwork now maturing into tested, operational elements.

## 8 List of References

- [1] "CRISP-DM," [Online]. Available: <https://www.datascience-pm.com/crisp-dm-2/>.
- [2] J. C. (. R. K. (. T. K. (. T. R. (. C. S. (. a. R. W. (. Pete Chapman (NCR), CRISP-DM 1.0 Step-by-step data mining guide, SPSS.
- [3] C. C. Aggarwal, Data Mining: The Textbook, Springer, 2016.
- [4] "IDSA Data sovereignty," [Online]. Available: <https://internationaldataspaces.org/why/data-sovereignty/>.
- [5] "International Data Spaces," [Online]. Available: [https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/introduction/1\\_1\\_goals\\_of\\_the\\_international\\_data\\_spaces](https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/introduction/1_1_goals_of_the_international_data_spaces).
- [6] C. B. F. H. F. K. Michael R. Berthold, Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data, Springer, 2010.
- [7] "Ydata Profiling," [Online]. Available: <https://docs.profiling.ydata.ai/>.
- [8] R. C. G. a. R. E. Woods, Digital Image Processing, Gatesmark: 3rd ed. Knoxville, 2007.
- [9] D. A. F. a. J. Ponce, Computer Vision a Modern Approach, Pearson.
- [10] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2022.
- [11] "Imputation of missing values," [Online]. Available: <https://scikit-learn.org/stable/modules/impute.htm>.
- [12] "Three Approaches to Encoding Time Information as Features for ML Models," [Online]. Available: <https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/>.
- [13] "Feature Selection," [Online]. Available: [https://scikit-learn.org/stable/modules/feature\\_selection.html](https://scikit-learn.org/stable/modules/feature_selection.html).
- [14] "Smart Data Models," [Online]. Available: <https://github.com/smart-data-models>.
- [15] W. F. a. all, "Graph Machine Learning in the Era of Large Language Models," 2024. [Online]. Available: <https://arxiv.org/abs/2404.14928>.
- [16] N. G. Yaron Haviv, Implementing MLOps in the Enterprise, O'Reilly.
- [17] "Apache Airflow," [Online]. Available: <https://airflow.apache.org/docs/apache-airflow/stable/>.
- [18] "Apache Beam," [Online]. Available: <https://beam.apache.org/documentation/>.
- [19] "Apache Flink," [Online]. Available: <https://nightlies.apache.org/flink/flink-docs-stable/>.
- [20] "Apache NiFi," [Online]. Available: <https://nifi.apache.org/documentation/v2/>.
- [21] "Airflow," [Online]. Available: <https://airflow.apache.org/>.
- [22] D. K. N. & H. S. Kreuzberger, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," 2022. [Online]. Available: <https://arxiv.org/abs/2205.02302>.

- [23] "Docker," [Online]. Available: <https://docker.com>.
- [24] "Kubernetes," [Online]. Available: <https://kubernetes.io/>.
- [25] "Argo Workflows," [Online]. Available: <https://argoproj.github.io/workflows/>.
- [26] "Kubeflow," [Online]. Available: <https://kubeflow.org/>.
- [27] "MLFlow," [Online]. Available: <https://mlflow.org/>.
- [28] "Prometheus," [Online]. Available: <https://prometheus.io/>.
- [29] "Grafana," [Online]. Available: <https://grafana.com/>.
- [30] NGINX, "Nginx Server," 2025. [Online]. Available: <https://nginx.org/en/>.
- [31] Portainer, "Kubernetes and Docker Container Management Software," 2025. [Online]. Available: <https://www.portainer.io/>.
- [32] pfSense, "pfSense - World's Most Trusted Open Source Firewall," 2025. [Online]. Available: <https://www.pfsense.org/>.
- [33] M. Beschokov, "What is CI/CD: Meaning, Definition & Pipeline Concepts (wallarm.com)," [Online]. Available: <https://www.wallarm.com/what/what-is-ci-cd-concept-how-can-it-work>.
- [34] Netcompany, "Cedar EU Demo Pipelines Repo: Demo Pipelines to experiment with Jenkins," 2025. [Online]. Available: <https://github.com/cedar-eu/demo-pipelines-repo>.
- [35] "Data Version Control - DVC," [Online]. Available: <https://dvc.org/>.
- [36] "Pachyderm," [Online]. Available: <https://pachyderm.com/>.
- [37] "Amazon SageMaker," [Online]. Available: <https://docs.aws.amazon.com/sagemaker/>.
- [38] "Jupyter," [Online]. Available: <https://jupyter.org/>.
- [39] "Weights and Biases," [Online]. Available: <https://wandb.ai/>.
- [40] "TensorFlow," [Online]. Available: <https://tensorflow.org/>.
- [41] "Apache Kafka," [Online]. Available: <https://kafka.apache.org/>.
- [42] "React Flow," [Online]. Available: <https://reactflow.dev/>.
- [43] "React," [Online]. Available: <https://react.dev/>.
- [44] "Seldon Core," [Online]. Available: <https://docs.seldon.io/projects/seldon-core>.
- [45] "Keycloak," [Online]. Available: <https://www.keycloak.org/>.
- [46] "MinIO," [Online]. Available: <https://min.io/>.
- [47] *IDSA Data Connector Report*, [https://internationaldataspaces.org/wp-content/uploads/dlm\\_uploads/IDSA-Data-Connector-Report-89-No-13-March-2024-5.pdf](https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Data-Connector-Report-89-No-13-March-2024-5.pdf).

- [48] Sovity EDC Framework, <https://edc.docs.sovity.de/>.
- [49] Sovity EDC Extensions, <https://github.com/sovity/edc-extensions>.
- [50] E. F. Foundation, "Certbot," 2025. [Online]. Available: <https://certbot.eff.org/>.
- [51] OWASP, "OWASP Secure Coding Practices, Quick Reference Guide, v2.1," 2010. [Online]. Available: [https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/assets/docs/OWASP\\_SCP\\_Quick\\_Reference\\_Guide\\_v21.pdf](https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/assets/docs/OWASP_SCP_Quick_Reference_Guide_v21.pdf).
- [52] "CEA Sigmo-IDS platform," [Online]. Available: <https://list.cea.fr/en/page/sigmo-ids-a-software-solution-for-secure-communications/>.
- [53] M. H. e. al., "Network intrusion detection system: A survey on AI-based techniques," *Experts Systems*, vol. 39, no. 9, 2022.
- [54] A. T. e. al., "FEDGAN-IDS: Privacy-preserving IDS using GAN and Federated Learning," *IEEE Comp. Comm.*, vol. 192, no. C, 2022.
- [55] K. R. e. al., "ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model," *Nature Sci. Rep.*, vol. 12, 2022.
- [56] D. P.-P. e. al., "Unveiling the potential of GNNs for robust Intrusion Detection," *ACM SIGMETRICS PER*, vol. 49, no. 4, 2022.
- [57] I. Dm-Crisp, "Dm-Crisp," [Online]. Available: <https://www.ibm.com/docs/en/spss-modeler/18.4.0?topic=dm-crisp-help-overview>.
- [58] J. H. Rüdiger Wirth, "CRISP-DM: Towards a Standard Process Model for Data Mining".
- [59] P. C. Clifton, "Introduction to Data Mining," [Online]. Available: <http://www.cs.purdue.edu/homes/clifton/cs490d/Process.ppt>.
- [60] "Jenkins," [Online]. Available: <https://www.jenkins.io/>.
- [61] "GitHub," [Online]. Available: <https://github.com/>.
- [62] "GitLab," [Online]. Available: <https://about.gitlab.com/>.
- [63] "Argo CD," [Online]. Available: <https://argo-cd.readthedocs.io/>.
- [64] "ISO/IEC 27001:2022," [Online]. Available: <https://www.iso.org/standard/27001>.
- [65] "ISO/IEC 27005:2022," [Online]. Available: <https://www.iso.org/standard/80585.html>.
- [66] "NIST SP 800-37 Rev.2," [Online]. Available: <https://csrc.nist.gov/pubs/sp/800/37/r2/final>.
- [67] ENISA, "Best practices on risk management," [Online]. Available: <https://www.enisa.europa.eu/topics/risk-management>.
- [68] OWASP, "Threat modelling," [Online]. Available: <https://owasp.org/www-project-threat-model/>.
- [69] "MITRE ATTACK," [Online]. Available: <https://attack.mitre.org/>.

- [70] "NIST AI 100-1," [Online]. Available: <https://doi.org/10.6028/NIST.AI.100-1> .
- [71] "NIST AI 100-2 E2023," [Online]. Available: <https://csrc.nist.gov/pubs/ai/100/2/e2023/final> .
- [72] "MITRE ATLAS," [Online]. Available: <https://atlas.mitre.org/> .
- [73] "STRIDE threat model," [Online]. Available: [https://cheatsheetseries.owasp.org/cheatsheets/Threat\\_Modeling\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html) .
- [74] "LINDDUN threat model," [Online]. Available: <https://linddun.org/threat-types/> .
- [75] *Eclipse Dataspace Components (EDC) Connector*, <https://github.com/eclipse-edc/Connector>.
- [76] *ECI Gateway Solutions*, <https://gatewaywise.ecisolutions.com/>.
- [77] *Huawei Boot-X Connector*, <https://www.boot-x.eu/>.
- [78] *OneNet Connector*, <https://www.onenet-project.eu/onenet-connector-included-in-the-idsa-data-connector-report/>.
- [79] *TNO Security Gateway*, <https://tno-tsg.gitlab.io/>.
- [80] *Smart Connected Supplier Network*, <https://smart-connected-supplier-network.gitbook.io/processmanual/>.
- [81] *Trusted Connector*, <https://github.com/Fraunhofer-AISEC/trusted-connector>.
- [82] *AI.SOV Connector*, <https://ai-sov.eu/>.
- [83] *GDSO Connector*, <https://gdso.org/Home>.
- [84] *Tech2B Connector*, <https://app.tech2b.cc/apps/6/SCSN Connector/SCSN>.
- [85] *Telekom DIH Connector*, <https://internationaldataspaces.org/t-systems-and-idsa-achieve-milestone-for-data-spaces-first-certification-of-a-connector-promotes-standardization-and-interoperability/>.
- [86] *Triton Enterprise Connector*, <https://www.dataspace.fi/en/data-intermediation-service>.
- [87] *Dat4Zero Project Website (where Kharon Connector was developed)*, <https://dat4zero.eu/work-packages/>.
- [88] *Trusted Supplier Connector*, <https://gec.io/solutions/gaia-x-dienste/>.
- [89] *VTT DSIL Connector*, <https://www.ids-finance.fi/vtt-has-officially-kicked-off-the-certification-process-for-their-ids-connector/>.
- [90] *TRUE Connector*, <https://github.com/Engineering-Research-and-Development/true-connector>.
- [91] N. Developers, "NetworkX: Network Analysis in Python (Version 3.5)," 2025. [Online]. Available: <https://github.com/networkx/networkx>.
- [92] N. Developers., "NetworkX: Network Analysis in Python," [Online]. Available: <https://github.com/networkx/networkx>. [Accessed 2025].



Funded by the  
European Union

*This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101135577*