D.3.4 Report on data management for applications in CECC – Final

Document Summary Information

Project Identifier	HORIZON-CL4-2022-DATA-01. Project 101093129		
Project name	Agile and Cognitive Cloud-edge Continuum management - Initial		
Acronym	AC³		
Start Date	January 1, 2023	**End Date**	December 31, 2025
Project URL	www.ac3-project.eu		
Deliverable	D3.4 - Report on data management for applications in CECC – Final		
Work Package	WP3		
Contractual due date	30/06/2025	**Actual submission date**	30/06/2025
Type	R — Document, report	**Dissemination Level**	PU
Lead Beneficiary	SPA		
Responsible Author	D. Amaxilatis (SPA)		
Contributors	D. Amaxilatis (SPA), N. Tsironis (SPA), T. Sarantakos (SPA), S. Sengupta (ION), D. Klonidis (UBI), A. Kadouma (FIN)		
Peer reviewer(s)	Adlen Ksentini (EUR), Christos Verikoukis (ATH)		

Funded by the European Union

AC³ project has received funding from European Union's Horizon Europe research and innovation programme under Grant Agreement No 101093129.

Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete	Changes	Contributor(s)
V0.1	24/02/2025	0%	Initial Deliverable Structure	D. Amaxilatis (SPA)
V0.2	15/04/2025	15%	Revisited structure, initial comments	D. Klonidis (UBU), N. Tsironis (SPA), S. Sengupta (ION)
V0.3	16/05/2025	25%	Editing of Section 2	T. Sarantakos (SPA), D. Klonidis (UBI)
V0.4	17/05/2025	45%	Editing of Section 3	D. Amaxilatis (SPA), A. Kadouma (FIN)
V0.5	20/05/2025	60%	Editing of Section 4	S. Sengupta (ION) N. Tsironis (SPA)
V0.6	25/05/2025	75%	Editing of Section 5	D. Amaxilatis (SPA)
V0.7	07/06/2025	95%	Final editing and corrections	T. Sarantakos (SPA)
V0.8	20/06/2025	80%	Updates and Executive Summary	D. Klonidis (UBI)
V0.9	21/06/2025	90%	Internal Review	A. Ksentini (EUR)
V1.0	30/06/2025	100%	Final Review and Submission	C. Verikoukis (ISI)

Disclaimer

Copyright message

Table of Contents

List of Figures

List of Tables

List of Code Extracts

Glossary of terms and abbreviations used

Abbreviation / Term	Description
AC^3	Agile and Cognitive Cloud edge Continuum management
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
AppD	Application Descriptor
CA	Certificate Authority
CECC	Cloud Edge Computing Continuum
CECCM	Cloud Edge Computing Continuum Manager
CI/CD	Continuous Integration (CI) and Continuous Delivery (CD)
CPU	Central Processing Unit
CRUD	Create, Read, Update, and Delete
DCAT-AP	DCAT Application profile for data portals in Europe
DDoS	Distributed Denial of Service
DQN	Deep Q-Networks
EDC	Eclipse Dataspace Components
ETSI	European Telecommunications Standards Institute
GUI	Graphical User Interface
GXFS	Gaia-X Federation Services
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IDS	International Data Spaces
IDSA	International Data Spaces Association
IoT	Internet of Things
JSON	JavaScript Object Notation
LCM	Life Cycle Management

LSTM	Long short-term memory
ML	Machine Learning
MQTT	MQ Telemetry Transport
PaaS	Platform as a Service
QoS	Quality of Service
RBAC	Role-Based Access Control
SLA	Service Level Agreement
OWL	Web Ontology Language
WAF	Web Application Firewall
YAML	YAML Ain't Markup Language

Executive Summary

The goal of this deliverable (D3.4) is to finalize and consolidate the work in the AC³ data management mechanisms, marking the completion of the development cycle that began with the initial design documented in D3.3 [1]. As one of the key innovations of the AC³ project, the data management framework has matured into a robust, scalable, and interoperable platform that seamlessly integrates data handling functionalities into the AC³ architecture, providing native support for all data-related aspects during the deployment and management of applications over the Cloud-Edge-Client Continuum (CECC).

Designed as a Platform-as-a-Service (PaaS) model, the AC³ data management platform enables application developers to incorporate data lifecycle capabilities directly into their deployments. The finalized framework now fully supports efficient data integration, semantic interoperability, advanced data discovery, secure and compliant data sharing, real-time streaming, cataloguing of both data and services, and dynamic deployment—all underpinned by strong privacy and scalability features. It leverages modern architectural patterns, such as microservices and semantic web technologies, and aligns with open European data space standards (DSSC[2], Gaia-X[3] and IDSA[4]).

This deliverable builds upon the three core development areas initially identified in D3.3:

- Interfacing with End Users: The AC³ Data Catalogue has evolved into a comprehensive AC³ Catalogue encompassing both data and service offerings. Powered by Piveau, it provides intuitive and semantically rich access to datasets and associated services. Recent extensions include custom metadata schemas for datasets and services, expressed via SHACL shapes aligned with DCAT-AP profiles. These allow users to explore and deploy data assets through clearly defined, searchable metadata fields, including licensing, geographic scope, endpoints, and required service connectors. A completely updated web interface was also developed to facilitate seamless interaction with the catalogue, supporting dataset-to-service linkage and visual exploration by developers and stakeholders.
- Interfacing with Data Sources: The implementation of Hot and Cold Data Connectors has been finalized. The Hot Data Connector facilitates real-time integration of streaming data from sensors and cameras (as utilized in Use Cases 1 and 2), supporting low-latency ingestion and forwarding via the AC³ runtime. The Cold Data Connector, integrated with IONOS S3 services, enables the secure retrieval and long-term storage of historical datasets, such as astronomical observations from Use Case 3. These connectors are designed to support contract-based data exchange via the Eclipse Dataspace Components (EDC) Connector, enabling controlled access to external data sources through standardized negotiation and trust mechanisms. All connectors are now aligned with GDPR principles and support fine-grained access control via Keycloak.
- Using Integrated Data Sources through Service Deployment and Management: A major enhancement in this final phase is the tight coupling of datasets to service offerings. Each dataset entry in the AC³ Catalogue can now reference required EDC Connectors and related processing services (e.g., format converters, ETL pipelines, analytics modules) that must be deployed alongside it. This linkage supports automation during application deployment and ensures a high degree of interoperability and reusability. Moreover, the deployment models now incorporate detailed service-level agreements (SLAs) and container resource specifications, ensuring predictable and manageable resource consumption within the CECCM runtime environment.

The entire AC³ data management infrastructure continues to be built upon the Piveau framework, which offers components such as the Hub UI, Hub Repository, Search, Virtuoso RDF Triple Store, and Elasticsearch backend. These components collectively enable semantic indexing, efficient query execution, and rich metadata storage. Keycloak remains integrated for managing user authentication and authorization workflows.

In terms of architecture, the data management elements are now fully embedded within the AC³ platform. All catalogue interactions, data source connectors, and service deployment elements are exposed through well-defined APIs and interfaces. This tight integration supports seamless coordination between the application lifecycle manager (LCM), data connectors, and the AC³ catalogue, allowing end-to-end management of application data requirements.

With these advancements, the AC³ data management platform now meets the objectives set forth in Task T3.4, offering a mature and extensible foundation for data-intensive CECC applications. It enables: (a) Semantic interoperability and dynamic discovery of both datasets and services; (b) Real-time and historical data handling through standardized Hot/Cold connectors; (c) GDPR-compliant data usage via contract negotiation mechanisms; (d) Automated service deployment linked directly to dataset requirements; and (e) Operational scalability through resource-constrained, SLA-aware microservices.

This deliverable represents the culmination of efforts in WP3 to design and implement a full-fledged, standards-aligned data management system for the AC³ platform. With this infrastructure now in place, the platform is prepared for real-world operation, offering value to use case partners, third-party developers, and the broader European data space ecosystem.

1 Introduction

This section provides an overview of the work reported in this deliverable, including the main purpose and the related key objectives. It also describes the link with the other project activities and related deliverables. The targeted outcomes are summarized next and mapped to the GA activities and the deliverable sections. The final subsection provides the structure of the deliverable.

1.1 Overview – Purpose and objectives

This deliverable (D3.4) presents the final status of the AC³ Data Management Platform as a Service (PaaS), concluding the development and integration cycle that began with D3.3. It documents the full realization of the platform's data management capabilities, which are essential for efficient, secure, and interoperable handling of data across the Cloud-Edge-Client Continuum (CECC). The platform has matured from architectural planning to an operational system that supports the dynamic deployment of data-driven applications within a federated computing environment.

Initially conceived as a PaaS framework, the AC³ data management layer was designed to allow application developers to embed complete data lifecycle management within their services. This includes seamless integration of heterogeneous data sources, dynamic discovery and cataloguing, advanced processing capabilities, and strict adherence to data privacy and security standards. With this final deliverable, these goals have been fully implemented and validated.

Since D3.3, the AC³ Data Catalogue has been significantly extended into a unified AC³ Catalogue that also includes service descriptions. Built on the Piveau framework and enhanced with SHACL-based metadata models aligned with DCAT-AP, Gaia-X and IDSA, the catalogue now enables semantic linkage between datasets and the services required to process them. A redesigned web interface further facilitates intuitive discovery and interaction for developers and stakeholders.

The implementation of Hot and Cold Data Connectors is now complete. Real-time data ingestion is enabled through the Hot Data Connector, while persistent storage and retrieval are supported via the Cold Connector integrated with IONOS S3. Both are based on the Eclipse Dataspace Connector and support contract-based data exchange, ensuring GDPR compliance and secure federation.

These developments are now tightly integrated with the CECC lifecycle manager, enabling automated deployment of workflows based on metadata-driven service requirements. All use case data sources have been connected, and the full platform has been tested in operational conditions.

With the completion of the second implementation cycle, AC³ delivers a fully functional, standards-aligned data management platform. It now stands as a critical enabler of intelligent, secure, and scalable application deployment across diverse domains and infrastructures, meeting the technical objectives of Task T3.4 and the broader goals of the AC³ project.

1.2 Link with other project activities

The work presented in this deliverable is primarily linked to Task 3.4, which covers the design and implementation of the AC³ Data Management Platform and is closely supported by Task 3.1 in relation to the development of the AC³ Data and Service Catalogue.

Early architectural input from WP2, particularly through deliverables D2.1 and D2.3, played a key role in shaping the data management design, informing the selection of components and defining key interfacing mechanisms.

These architectural foundations were fully aligned and validated against the final CECC architecture documented in D2.2, ensuring cohesion between the data management layer and the overall platform design.

This work also contributes directly to WP5, particularly Task 5.1, by enabling runtime deployment and orchestration of services that rely on specific data inputs. The integration between the AC³ Catalogue, data connectors, and lifecycle management tools supports automated, metadata-driven service composition, while enforcing access control and compliance with data-sharing agreements.

1.3 Mapping AC³ Outputs

The purpose of this section is to map AC³ Grant Agreement commitments, both within the formal Deliverable and Task description, against the project's respective outputs and work performed.

Table 1: Adherence to AC³ GA Deliverable & Tasks Descriptions

AC³ GA Component Title	AC³ GA Component Outline	Respective Document Chapter(s)	Justification
DELIVERABLE			
D3.4 Report on data management for applications in CECC – Final			
TASKS			
T3.4 Applications data management as a PaaS	Final Data management mechanism	Section 3 Section 4	Interfacing and flow of data between the continuum and external data sources. Utilization of integrated data sources to drive the deployment and ongoing management of services Main purpose and implemented functionalities, and Mapping of data management in the overall architecture
T3.4 Applications data management as a PaaS	Connectivity with Data & Service catalogue	Section 3 Section 2	Provides The components that form the AC³ Data & Service Catalogue, their interactions, and interfaces Mapping of Data & Service catalogue in the overall architecture
T6.3 Exploitation, Data and IPR Management	Data Management & Protection	Section 3 Section 5	Provides information on how the project takes a data protection-by-design approach in accordance with the GDPR, Gaia-X and IDSA reference architectures and guidelines

1.4 Deliverable Overview and Report Structure

This deliverable presents the final outcomes of the AC³ data management framework, reflecting the transition from initial design to full implementation. It follows a structured layout that captures the technical progress, integration efforts, and alignment with project-wide objectives:

- Section 2 revisits the AC³ architecture and presents the complete set of data management functionalities now embedded within the platform, emphasizing their integration across the Cloud-Edge-Client Continuum.
- Section 3 describes the development and enhancement of the unified AC³ Data and Service Catalogue, including its semantic extensions, metadata schema, and the new user interface supporting intuitive data-service linkage.
- Section 4 outlines the final implementation of Hot and Cold Data Connectors, detailing their role in supporting real-time data ingestion and secure access to stored datasets, as well as their integration with the Eclipse Dataspace Connector.
- Section 5 addresses data governance and usage policies, with a focus on GDPR compliance, identity management, and contract-based data access facilitated through the AC³ framework.
- Section 6 provides concluding insights on the overall progress, summarizing the technical outcomes and readiness of the AC³ data management platform for operational deployment.

2 Data management functionalities in the AC³ platform architecture

This section revisits and updates the core aspects of data management within the AC³ Cloud-Edge-Client Continuum (CECC) architecture, building upon the initial framework detailed in D3.3. It first outlines the main concept and targeted functionalities of the data management scheme, highlighting its integrated functionalities within the CECC framework. Subsequently, the section presents the mapping of the data management modules onto the overall AC³ architecture, detailing their interdependencies and interfaces with other key components.

2.1 Main Concept and Targeted Functionalities

The AC³ data management scheme provides a comprehensive and mature set of functionalities natively integrated within the Cloud-Edge-Client Continuum (CECC) architecture. Building upon the initial design outlined in D3.3, these finalized mechanisms ensure efficient, secure, interoperable, and compliant data handling for applications deployed across the continuum. The framework is designed to support the entire data lifecycle within applications, from discovery and ingestion to processing and compliant sharing. Key functionalities are realized through the AC³ Catalogue, versatile Data Connectors, and mechanisms for data-driven service enablement. These are listed in the following paragraphs and further detailed in Sections 3, 4, and 5, respectively.

2.1.1 Functionalities related to the AC³ Catalogue

The AC³ Catalogue is central to the data management scheme, offering robust capabilities for discovery, access, and semantic understanding of data and service resources:

- Unified Discovery and Access: The AC³ Catalogue serves as a consolidated platform for discovering and accessing both datasets and the services required for their processing. It has evolved from a solely data-focused registry into a unified Data and Service Catalogue.
- Semantic Interoperability and Rich Metadata: By adopting the Piveau framework, the catalogue utilizes custom metadata models expressed as SHACL shapes, aligned with DCAT-AP profiles, to provide rich semantic descriptions for both datasets and services. This approach facilitates a clear semantic linkage between datasets and the specific services or connectors needed to interact with them.
- Enhanced User Interaction: A completely updated web interface offers users an intuitive way to search, explore, and visually understand the available datasets, services, and their interconnections. This includes views for dataset lists, individual dataset details with geographical information and contact points, and similar views for services.
- Automated Application Composition Support: The catalogue is designed to be programmatically accessed, notably by the Ontology and Semantic Reasoner (OSR). The OSR queries the AC³ Catalogue to retrieve detailed metadata about datasets and their associated services (e.g., requiredConnector, requiredServiceOffering). This information is then used to dynamically and automatically incorporate the necessary data management components into deployable application descriptors.

2.1.2 Functionalities related to Data Connectors

The AC³ Data Connectors are responsible for managing the interaction with diverse data sources, ensuring secure and efficient data exchange:

- Versatile Data Ingestion and Access: The finalized implementation includes distinct Hot and Cold Data Connectors to provide standardized mechanisms for accessing a variety of data types.
    - The Hot Data Connector facilitates real-time, low-latency integration of streaming data, such as that from IoT sensors and cameras, as used in project Use Cases 1 and 2. It supports efficient ingestion and forwarding of data streams, and its implementation can be extended, for instance, with RabbitMQ, for asynchronous message handling and improved scalability.

   - The Cold Data Connector is integrated with IONOS S3 services, enabling secure retrieval and management of historical or infrequently accessed datasets, like the astronomical observations from Use Case 3.
- Secure and Compliant Data Exchange: Both Hot and Cold Data Connectors are based on the EDC framework. This enables controlled and policy-enforced data transactions through standardized contract negotiation processes between data providers and consumers. These contracts, potentially defined using ODRL, govern the terms of data access and usage.
- Access Control and GDPR Alignment: The connectors operate in alignment with GDPR principles. Fine-grained access control for data transactions is supported and can be managed via integrated identity management solutions like Keycloak.

2.1.3 Functionalities related to Data-Driven Service Enablement

The AC³ data management scheme ensures that data is not just accessible but also effectively integrated into the lifecycle of services, supporting automated and compliant application deployment:

- Integrated Data and Service Deployment: A key advancement is the tight coupling of dataset entries in the AC³ Catalogue with the specific EDC Connectors and processing services (e.g., format converters, ETL pipelines, analytics modules) required for their utilization.
- Automated and Interoperable Workflows: This linkage between datasets and their necessary service components directly supports automation during application deployment by the CECCM. The OSR leverages this to compose application descriptors that include all necessary data handling microservices, thereby ensuring a high degree of interoperability and reusability.
- Resource-Aware and SLA-Driven Management: Deployment models for services, as described in the AC³ Catalogue, now incorporate detailed specifications for service-level agreements (SLAs) (e.g., availability, response time) and container resource requirements (e.g., CPU, memory, storage). This ensures predictable performance and manageable resource consumption within the CECCM runtime environment.
- Governance and Compliance in Data Usage: The framework places a strong emphasis on responsible data usage and adherence to GDPR. This is achieved by integrating Gaia-X principles for data sovereignty, utilizing EDC for policy-enforced data sharing contracts, and ensuring transparency through comprehensive metadata in the Piveau-based catalogue. These mechanisms support auditable and accountable data sharing within the AC³ ecosystem.

2.2 Mapping to the AC³ Architecture

The data management functionalities described in the previous subsection are embedded within the overall AC³ architecture, as described in the final platform architecture presented in D2.2 and depicted in Figure 1. This integration ensures a cohesive approach to handling data across the CECC.

The AC³ Catalogue is positioned within the application gateway layer of the AC³ architecture. It directly interfaces with application developers and end-users through its Web Interface, allowing for the exploration and discovery of datasets and services. It also interconnects with the Ontology and Semantic Reasoner (OSR), which queries the AC³ Catalogue to retrieve metadata about datasets and their associated services (including required connectors and processing services) (see D3.2 for further details). This information is fundamental for the OSR to compose semantically sound and complete application descriptors.

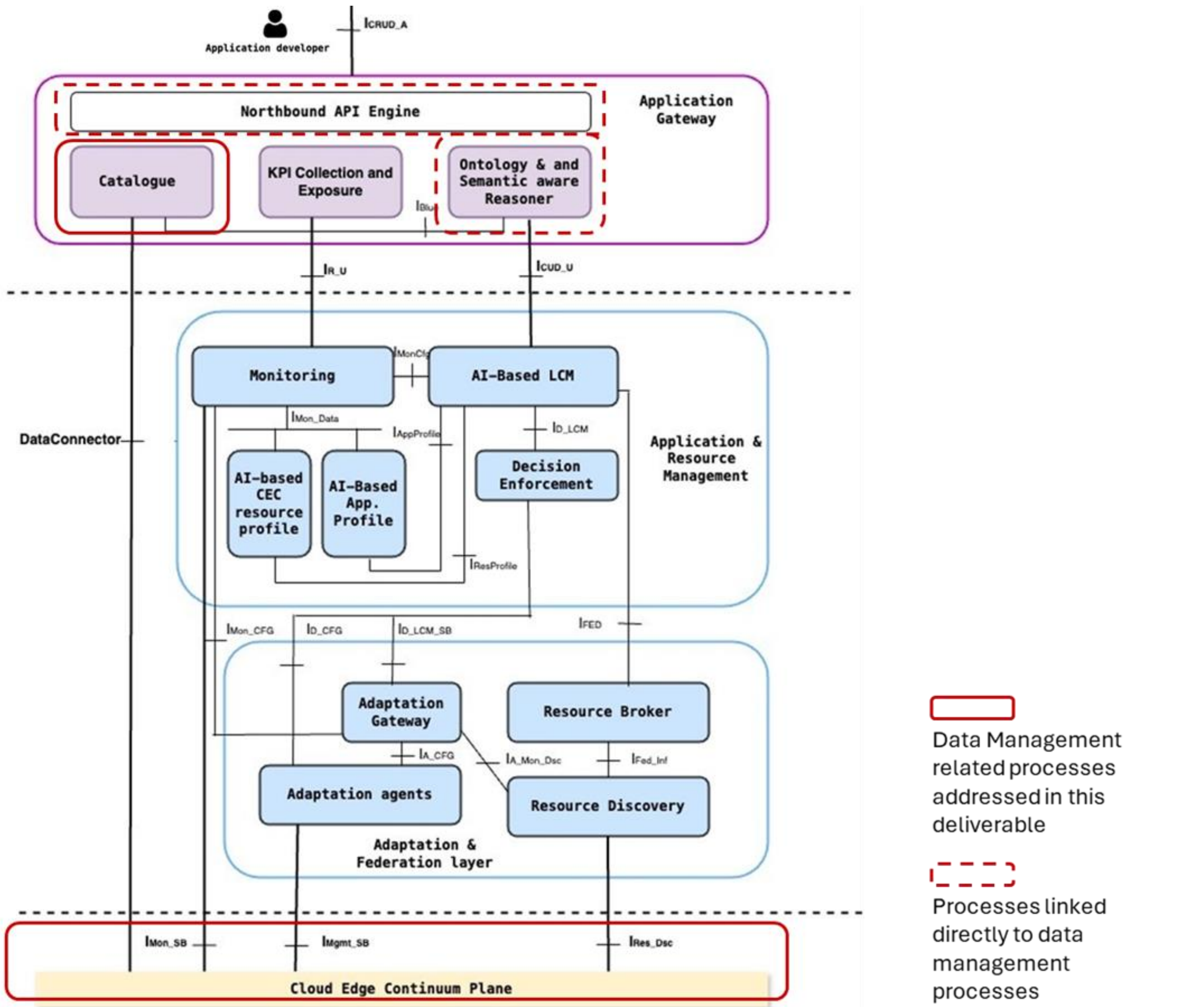

Figure 1 – Mapping of data management processes to the overall AC³ architecture including also the directly linked processes

Data Connectors function as the primary interface for data exchange between data sources and AC³ applications across the Cloud Edge Continuum. On the data provider side, external EDC Connectors publish dataset metadata (like dcat:endpointURL and dcat:assetName) to the AC³ Catalogue.

Within an AC³ application context, the consumer-side Data Connector (deployed as part of the application) utilizes metadata from the AC³ Catalogue (often via the OSR-generated descriptor) to initiate contract negotiations with the provider's EDC Connector and manage data transfer. Their deployment and configuration are managed through the application and resource management layer processes, based on the specifications derived by the OSR from the Catalogue.

Similarly, the data-related services are deployed and managed through the synergistic operation of the Catalogue and Connectors with the OSR and finally the LCM processes, ensuring that data is seamlessly integrated into service deployment and management. The OSR interprets the relationships defined in the AC³ Catalogue between datasets and their required services (connectors, processing logic). It then incorporates these data management services and their configurations into the application descriptor provided to the AI-based LCM. The LCM orchestrates the deployment of the complete application, including these data-specific microservices (e.g., Data Connector instances, data mappers, manipulators) and ensures they operate according to defined SLAs and resource specifications, also catalogued.

3 Data and Service Catalogue

As described in D3.3, sharing data in the context of the AC³ Data Management Platform as a Service is achieved using the AC³ Data Catalogue service. This service is instrumental in providing all the appropriate descriptions, metadata, information, and guidance to software developers regarding the use of the offered datasets and data sources. These data sources can range over a broad spectrum of domains and need to be properly described and annotated to be easily searched and discovered. The annotation of the data sources is done using rich semantic information based on well-established ontologies provided by the owners of the data during the initial phase of the data source registration.

For AC³, we have opted to use a well-established data catalogue implementation, Piveau, rather than re implementing the wheel and providing thus another solution in the field. Piveau is comprised of multiple internal components that have been described in D3.3 and are still used to provide us with the necessary functionalities.

Additionally, the functionalities offered by Piveau have allowed us to extend the system's capabilities to not only host information about the data sources and datasets of AC³ but also host information about the services that are needed in order to interact with these data sources, forming up also a Service Catalogue that lives in the same deployment with the Data Catalogue and can be accessed using a uniform way. To achieve this, we had to extend the schema of information that can be stored and retrieved from Piveau and provide an updated web interface for accessing and viewing the data hosted in it.

As a result, we now have an implementation of a combined AC³ Data & Service Catalogue that we now call the AC³ Catalogue for simplicity, offering both functionalities.

3.1 Extending Piveau with custom metadata models

The first step to migrate our AC³ Catalogue was to deploy two custom metadata models to describe our Datasets and Services. These models are described as shapes in Shapes Constraint Language (SHACL)[5] and should abide by the DCAT Application profile for data portals in Europe (DCAT-AP)[6]. The process followed to implement this transformation is described (PIVEAU[7]) and contains the following:

- Enabling the use of profiles in Piveau and defining a new provider based on a local directory.
- Creating the SHACL files that contain the descriptions of the metadata assigned to each entity to be stored under these new data type definitions.
- Defining the names under which each new shape will appear in the catalogue
- Redeploying the Piveau components with the new configuration

After this change, the internal models and metadata that Piveau processes and stores have been updated and are ready to be used. Information on the exact shapes used for our two main objects, Dataset and Service are provided in the next two sub-sections.

3.2 AC³ Data Catalogue Schema

The Data Catalogue is comprised of multiple Dataset entities. Each Dataset entity has metadata that defines its characteristics, the providing organization, as well as the Services needed for it to be effectively used in the context of AC³. The Dataset shape used in AC³ is presented in Code 1.

Specifically, at the time of writing this document it contains:

- *dct:title, schema:name and schema:description* as text-based information that describes the dataset.
- *gx:license*: as a description of the associated License under which the data provider offers the dataset.

- *dcat:assetName, dcat:endpointURL, dcat:endpointDescription*: as information on how the dataset can be accessed, though an EDC Connector (endpointURL) and the assetName under which the dataset is available, as one EDC Connector can offer more than one datasets at the same endpoint.
- *geo:lat and geo:long* define the numerical geographical coordinates of where the dataset is hosted.
- *dcat:contactPoint* specifies the contact information, including *vcard:fn* for the organization or individual name, *vcard:hasName* for the contact person's name, and *vcard:hasEmail* for the contact email address.
- *ac3:requiredConnector*: is an AC³ metadata element that provides a link to a Service Offering in the AC³ (or an external) Service Catalogue that contains information on EDC Connector service that can connect to and access the data from the providing connector.
- *ac3:requiredServiceOffering*: is an AC³ metadata element that links this dataset to a number of Service Offerings that need to be deployed also by AC³ to effectively receive and handle the incoming datasets once the EDC Connector has successfully requested them from the providing Connector. These services can be of multiple types, providing functionalities that range from simple data transformations (e.g., converting a csv to a json file, converting data units of measurements) or more complex processing tasks like generating aggregated data from raw measurements, building simple ETL pipelines.

```
PREFIX piveau: <https://piveau.eu/ns/voc#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX shacl:  <http://www.w3.org/ns/shacl#>
PREFIX skos:   <http://www.w3.org/2004/02/skos/core#>
PREFIX dct:    <http://purl.org/dc/terms/>
PREFIX dcat:   <http://www.w3.org/ns/dcat#>
PREFIX schema: <https://schema.org/>
PREFIX gx:     <https://w3id.org/gaia-x/development#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ac3:    <https://ac3-project.eu/#>

gx:DatasetShape
    rdf:type                  shacl:NodeShape ;
    shacl:ignoredProperties   ( rdf:type ) ;
    shacl:property            [ shacl:maxCount       1 ;
                                piveau:mappingClass "StandardText" ;
                                piveau:mappingName  "id" ] ;
    shacl:property            [ shacl:datatype       xsd:string ;
                                shacl:description    "Human readable title"@en ;
                                shacl:maxCount       1 ;
                                shacl:name           "Title" ;
                                shacl:path           dct:title ;
                                piveau:mappingClass "StandardText" ;
                                piveau:mappingName  "title" ] ;
    shacl:property            [ shacl:datatype       xsd:string ;
                                shacl:description    "Human readable name"@en ;
                                shacl:maxCount       2 ;
                                shacl:name           "Name" ;
                                shacl:path           schema:name ;
                                piveau:mappingClass "StandardText" ;
                                piveau:mappingName  "name" ] ;
    shacl:property            [ shacl:datatype       xsd:string ;
                                shacl:description    "Free text description"@en ;
                                shacl:maxCount       1 ;
                                shacl:name           "Description" ;
                                shacl:path           schema:description ;
                                piveau:mappingClass "StandardText" ;
```

```
                          piveau:mappingName  "description" ] ;
    shacl:property      [ shacl:datatype      xsd:string ;
                          shacl:description   "SPDX identifiers or URL"@en ;
                          shacl:in            ( "0BSD" "BSD-3-Clause" … ) ;
                          shacl:minCount      1 ;
                          shacl:name          "Licenses" ;
                          shacl:path          gx:license ;
                          piveau:mappingClass "StandardKeyword" ;
                          piveau:mappingName  "license" ] ;
    shacl:property      [ shacl:datatype      xsd:string ;
                          shacl:description   "The asset ID." ;
                          shacl:maxCount      1 ;
                          shacl:name          "Asset ID" ;
                          shacl:path          dcat:assetName ;
                          piveau:mappingClass "StandardText" ;
                          piveau:mappingName  "assetId" ] ;
    shacl:property      [ shacl:datatype      xsd:string ;
                          shacl:description   "Free text description"@en ;
                          shacl:maxCount      1 ;
                          shacl:name          "Endpoint Description" ;
                          shacl:path          dcat:endpointDescription ;
                          piveau:mappingClass "StandardText" ;
                          piveau:mappingName  "endpointDescription" ] ;
    shacl:property      [ shacl:datatype      xsd:string ;
                          shacl:description   "Distribution endpointURL"@en ;
                          shacl:maxCount      1 ;
                          shacl:name          "Endpoint URL" ;
                          shacl:path          dcat:endpointURL ;
                          piveau:mappingClass "StandardText" ;
                          piveau:mappingName  "endpointURL" ] ;
    shacl:property      [ shacl:datatype      xsd:double ;
                          shacl:minCount      1 ;
                          shacl:maxCount      1 ;
                          shacl:path          geo:lat ;
                          piveau:mappingClass "StandardText" ;
                          piveau:mappingName  "lat" ] ;
    shacl:property      [ shacl:datatype      xsd:double ;
                          shacl:minCount      1 ;
                          shacl:maxCount      1 ;
                          shacl:path          geo:long ;
                          piveau:mappingClass "StandardText" ;
                          piveau:mappingName  "lon" ] ;
    shacl:property      [ shacl:datatype      xsd:string ;
                          shacl:description   "Contact Point"@en ;
                          shacl:minCount      1 ;
                          shacl:name          "Contact Point" ;
                          shacl:node          ac3:ContactPointShape ;
                          shacl:path          dcat:contactPoint ;
                          piveau:mappingClass "StandardText" ;
                          piveau:mappingName  "contactPoint" ] ;
    shacl:property      [ shacl:datatype      xsd:string ;
                          shacl:description   "EDC connector service
offering"@en ;
                          shacl:minCount      1 ;
                          shacl:name          "Required Connector" ;
                          shacl:node          gx:ServiceOfferingShape ;
```

```
                              shacl:path          ac3:requiredConnector ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "requiredConnector" ] ;
    shacl:property          [ shacl:datatype      xsd:string ;
                              shacl:description   "Service offering needed"@en ;
                              shacl:minCount      0 ;
                              shacl:name          "Required Service Offering" ;
                              shacl:node          gx:ServiceOfferingShape ;
                              shacl:path          ac3:requiredServiceOffering ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "requiredServiceOffering" ] ;
    shacl:targetClass       gx:Dataset .

ac3:ContactPointShape
    rdf:type         shacl:NodeShape ;
    shacl:property [ shacl:path     rdf:type ;
                     shacl:hasValue vcard:Kind ;  # Ensures it is of type
vcard:Kind
                   ] ;
    shacl:property [ shacl:path     vcard:hasEmail ;
                     shacl:datatype xsd:anyURI ;  # Email must be a valid URI
                     shacl:minCount 1 ;
                     shacl:maxCount 1 ; ] ;
    shacl:property [ shacl:path     vcard:hasName ;
                     shacl:datatype xsd:string ;  # Name must be a string
                     shacl:minCount 1 ;
                     shacl:maxCount 1 ; ] ;
    shacl:property [ shacl:path     vcard:fn ;
                     shacl:datatype xsd:string ;  # Full name or organization
name
                     shacl:minCount 1 ;
                     shacl:maxCount 1 ; ] .
```

Code 1 Dataset shape

3.3 AC³ Service Catalogue Schema

The Service Catalogue is comprised of multiple Service entities. Each Service entity has metadata that defines its characteristics, the providing organization, as well as resource requirements or configuration properties to be used during the deployment of the Service. The Service shape used in AC³ is presented in Code 2.

Specifically, at the time of writing this document, it contains:

- *dct:title, schema:name and schema:description* as text-based information that describes the dataset.
- *ac3:image*: the docker image that can be used to deploy the service
- *ac3:EnvironmentVariable*: parameters that need to be provided to the service when deployed by the LCM
- *ac3:ExposedPort*: tcp/udp ports that need to be exposed for the service to be made available to the internet (e.g., 443 for an HTTPS webserver)
- *ac3:volumes*: data volumes that need to be created for the service to permanently store or share data with other services.

- *dcat:contactPoint* specifies the contact information, including *vcard:fn* for the organization or individual name, *vcard:hasName* for the contact person's name, and *vcard:hasEmail* for the contact email address.
- *gx:ContainerResourceLimits*: computational and storage resources required for service deployment, aligned with Gaia-X ContainerResourceLimits (e.g., CPU, memory, storage).
- *ac3:microservicesSLA*: service-level agreements defining operational commitments, including service availability (0-100%), maximum response time (Low/Medium/High) and data throughput (Low/Medium/High).

```
PREFIX piveau: <https://piveau.eu/ns/voc#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX shacl:  <http://www.w3.org/ns/shacl#>
PREFIX skos:   <http://www.w3.org/2004/02/skos/core#>
PREFIX schema: <https://schema.org/>
PREFIX dct:    <http://purl.org/dc/terms/>
PREFIX gx:     <https://w3id.org/gaia-x/development#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX ac3:    <https://ac3-project.eu/#>

ac3:ServiceShape
    rdf:type                shacl:NodeShape ;
    shacl:ignoredProperties ( rdf:type ) ;
    shacl:property          [ shacl:maxCount       1 ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "id" ] ;
    shacl:property          [ shacl:datatype      xsd:string ;
                              shacl:description   "Human readable title"@en ;
                              shacl:maxCount      1 ;
                              shacl:name          "Title" ;
                              shacl:path          dct:title ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "title" ] ;
    shacl:property          [ shacl:datatype      xsd:string ;
                              shacl:description   "Human readable name"@en ;
                              shacl:maxCount      1 ;
                              shacl:minCount      1 ;
                              shacl:name          "Name"@en ;
                              shacl:path          schema:name ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "name" ] ;
    shacl:property          [ shacl:datatype      xsd:string ;
                              shacl:description   "Free text description"@en ;
                              shacl:maxCount      1 ;
                              shacl:name          "Description"@en ;
                              shacl:path          schema:description ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "description" ] ;
    shacl:property          [ shacl:datatype      xsd:string ;
                              shacl:description   "The service's image."@en ;
                              shacl:maxCount      1 ;
                              shacl:name          "Image"@en ;
                              shacl:path          ac3:image ;
                              piveau:mappingClass "StandardText" ;
                              piveau:mappingName  "image" ] ;
    shacl:property          [ shacl:datatype    xsd:string ;
                              shacl:description "env variables."@en ;
```

```
                              shacl:minCount     0 ;
                              shacl:pattern      "^[A-Za-z_][A-Za-z0-9_]*=.+$" ;
                              shacl:path         ac3:EnvironmentVariable ] ;
    shacl:property          [ shacl:datatype     xsd:string ;
                              shacl:description "exposed ports."@en ;
                              shacl:minCount     0 ;
                              shacl:pattern      "^[0-9]+:[0-9]+$" ;
                              shacl:path         ac3:ExposedPort ] ;
    shacl:property          [ shacl:datatype     xsd:string ;
                              shacl:description "volumes."@en ;
                              shacl:minCount     0 ;
                              shacl:path         ac3:volumes ] ;
# https://docs.gaia-x.eu/ontology/development/classes/ContainerResourceLimits/
    shacl:property          [ shacl:description "Resource requirements."@en ;
                              shacl:minCount     0 ;
                              shacl:maxCount     1 ;
                              shacl:path         gx:ContainerResourceLimits ] ;
    shacl:property          [ shacl:description "The service's SLAs."@en ;
                              shacl:minCount     0 ;
                              shacl:maxCount     1 ;
                              shacl:path         ac3:microservicesSLA ] ;
    shacl:targetClass       gx:ServiceOffering .



# Define constraints for Microservices SLA
ac3:MicroservicesSLAShape
    a                     shacl:NodeShape ;
    shacl:targetClass gx:MicroservicesSLA ;
    shacl:property      [ shacl:path          ac3:serviceAvailability ;
                          shacl:datatype      schema:Number ;
                          shacl:minInclusive 0 ;
                          shacl:maxInclusive 100 ;
                          shacl:message       "Service Availability 0 - 100%." ] ;
    shacl:property      [ shacl:path     ac3:maxResponseTime ;
                          shacl:in       ( "Low" "Medium" "High" ) ;
                          shacl:message "Max Response Time Low, Medium or High" ] ;
    shacl:property      [ shacl:path     ac3:dataThroughput ;
                          shacl:in       ( "Low" "Medium" "High" ) ;
                          shacl:message "Data Throughput Low, Medium or High" ] .
```

Code 2 Service Shape

3.4 Interacting with the AC3 Catalogue

There are two ways users can interact with the AC³ Catalogue.

3.4.1 AC3 Catalogue Web Interface

Updating and extending the schema of the entities the AC³ Catalogue can handle and share means that the existing web interface presented in the previous deliverable could not handle all the new information available. As a result, an updated web interface was built for the AC³ Catalogue with clearer structure and information display elements. This portal has advanced search capabilities and allows Datasets to be linked visually to Services, facilitating Data and Service discovery for application developers, data scientists, or simple visitors of the website.

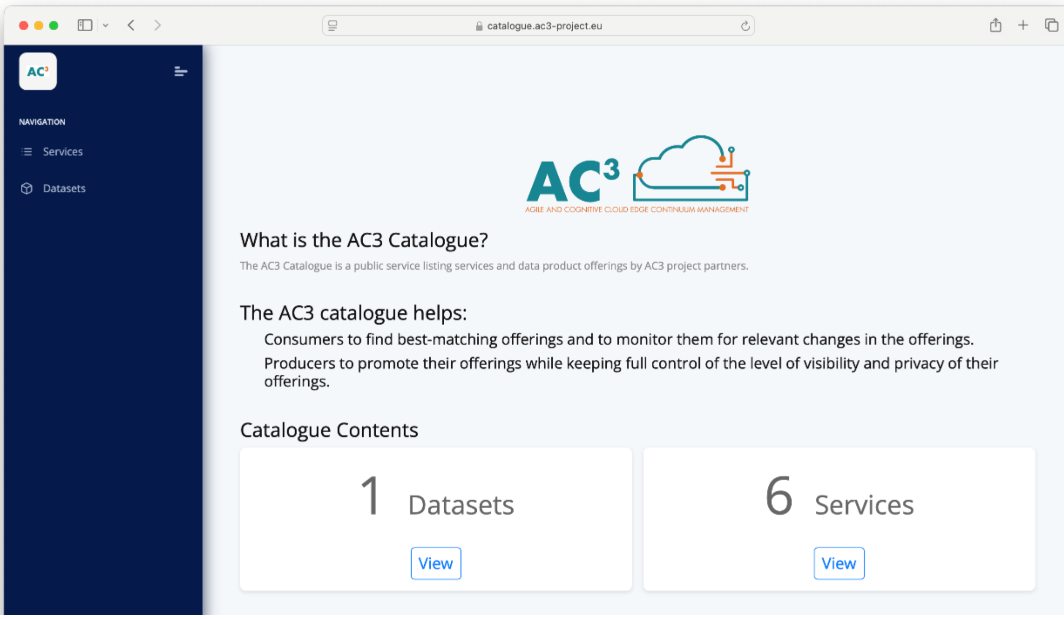

Figure 2 Home View of the AC³ Catalogue (Data & Service)

3.4.1.1 *Home page*

The AC³ Catalogue Home Page (Figure 2) provides an overview of the available resources. It displays the number of datasets and services currently listed in the catalogue, with two datasets and six services available at the time of viewing. Each category is accompanied by a "View" button, allowing users to explore the respective listings in more detail.

3.4.1.2 *Dataset List & View Pages*

The Datasets page (Figure 3) within the AC³ Catalogue provides an interface for users to explore the available datasets. At the top of the page, a search bar allows users to filter datasets by title, with a "Clear" button provided to reset the search field. Below the search functionality, the page displays dataset entries, each presented within a structured card format. Each dataset entry includes a brief description and a "View Details" button, enabling users to access further information. The page design maintains clarity and ease of navigation, ensuring that stakeholders can efficiently locate and review datasets relevant to their needs.

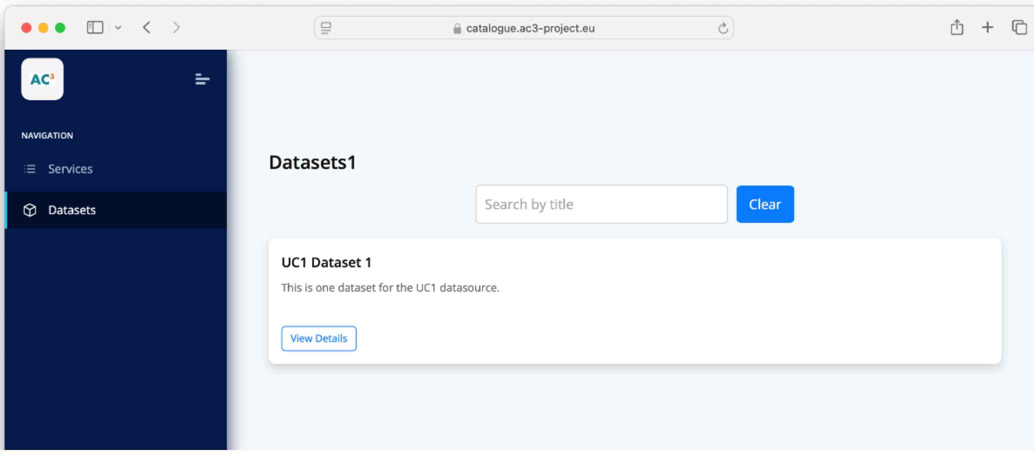

Figure 3 Dataset List in the AC³ Catalogue

The dataset details page (Figure 4) provides comprehensive information regarding its characteristics and accessibility. At the top, the dataset name is displayed, followed by a description explaining its purpose within the AC³ project. Further details include the dataset's license, ensuring clarity on usage rights. A map visualization is embedded within the page, pinpointing relevant geographical locations associated with the dataset. Additionally, contact information for key contributors is provided. Technical details such as the asset ID, endpoint description, and endpoint URL are included to facilitate integration and access. The dataset is identified under the asset ID, with its endpoint description. The endpoint URL is provided for direct interaction with the dataset's access point. At the bottom of the page, references to additional resources are included, listing a required connector and multiple required service offerings. Each of these elements is presented as a clickable link, allowing users to navigate to further relevant components within the AC³ Catalogue. The structured layout ensures that users can efficiently retrieve both high-level and technical details about the dataset.

3.4.1.3 Services List & View Pages

The Services page (Figure 5) within the AC³ Catalogue provides an overview of the available service offerings. At the top of the page, a search bar allows users to filter services by title, with a "Clear" button enabling them to reset their search criteria. Below the search functionality, the page displays a list of services, each presented in a structured card format. Each service entry includes a title, a brief description, and a "View Details" button for further exploration. The descriptions outline the specific functionality of each service, detailing how they contribute to the AC³ project and support data processing, analysis, or other key operations.

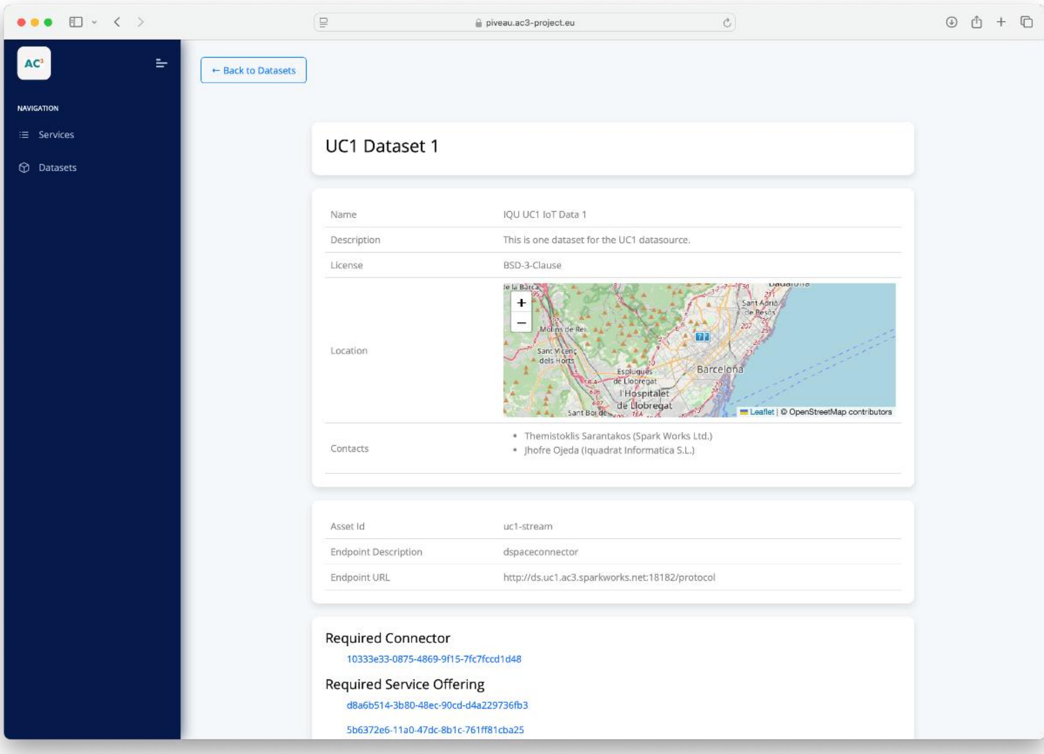

Figure 4 Dataset View for the AC³ Catalogue

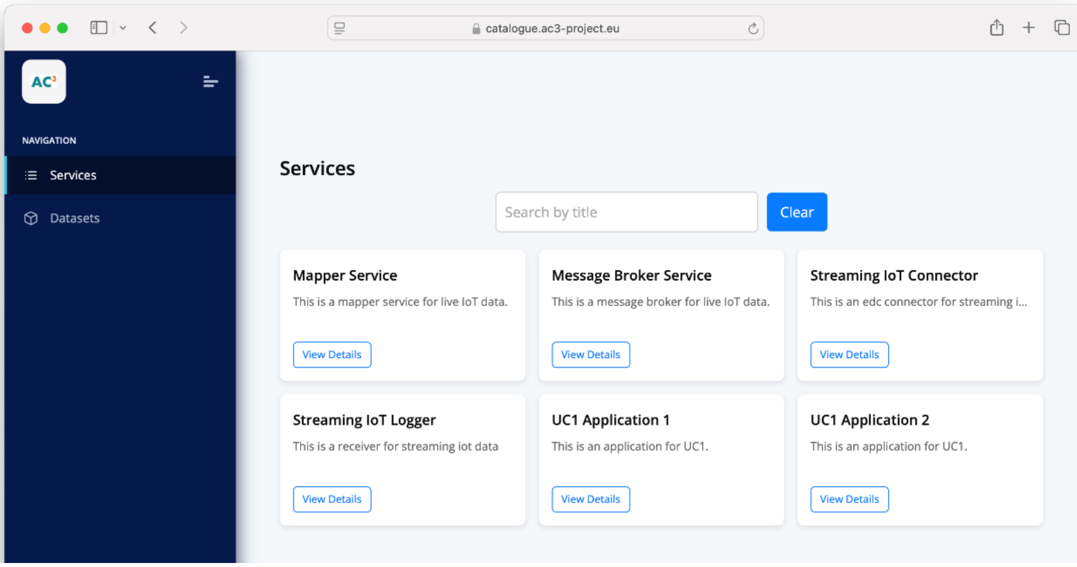

Figure 5 Service List in the AC³ Catalogue

The service details page (Figure 6) provides comprehensive information regarding its deployment and configuration. At the top, the dataset name is displayed, followed by a description explaining its purpose within the AC³ project. Further details include the dataset's license, ensuring clarity on usage rights. Additionally, contact information for key contributors is provided. A link is provided for the image that can be used to deploy the service. Technical details such as the environmental variables, exposed ports, and mappings, as well as data volumes needed, are depicted. Finally, information about the computing requirements and restrictions of the services is provided to better understand their computational needs. The structured layout ensures that users can efficiently retrieve both high-level and technical details about the service.

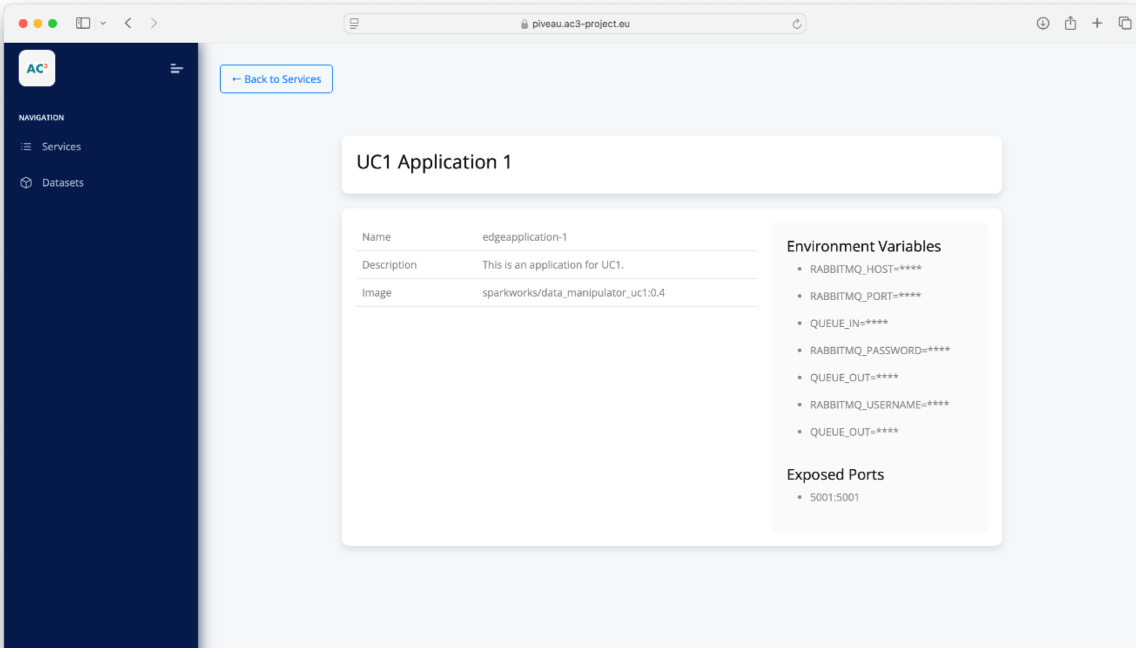

Figure 6 Service View for the AC³ Catalogue

3.4.2 Accessing AC3 data from the OSR

The Ontology and Semantic Reasoner (OSR) component dynamically integrates data management services into deployable application descriptors by interfacing with the AC³ Piveau catalogue. This integration enables the OSR to retrieve metadata about datasets and the associated services required for their access and processing, ensuring seamless interoperability and automation across the Cloud-Edge Continuum.

3.4.2.1 Data Catalogue Client

When a developer defines an application via the Graphical User Interface (GUI), one or more data source requirements may be included as part of the input. These data sources are described at a high level by the user, typically specifying a name or category. The OSR, through its Data Catalogue Client, communicates with the Piveau Data Catalogue to retrieve a complete list of available datasets. This process involves querying the Piveau dataset endpoint, receiving a JSON-encoded list of dataset URLs, and individually fetching detailed metadata for each dataset by its unique identifier. The OSR then compares the schema:name of each dataset entry to the user-provided dataset name. Upon finding a match, the OSR extracts key metadata elements including the dcat:assetId, the external data endpoint (defined by dcat:endpointURL), and most importantly, the references to related services using the requiredConnector and requiredServiceOffering fields. These two fields define the core set of microservices required to access and interface with the data asset, and they are essential for accurate and complete descriptor composition.

3.4.2.2 Service Catalogue Client

Once the OSR identifies the required connector and any additional service offerings associated with the dataset, it utilizes the Service Catalogue Client to query the Piveau Service Catalogue. This involves fetching service metadata for each referenced service using their respective URLs. The metadata returned typically includes the service name, image location, description, exposed ports, and defined environment variables. For connector services specifically, the OSR also enriches its configuration by injecting runtime variables such as ASSET_NAME and PROVIDER_DOMAIN, derived directly from the dataset metadata. These values enable the connector to properly interface with the remote data provider. This automated mechanism ensures that applications are always configured with the most up-to-date service definitions published in the catalogue, reducing the need for manual service management and minimizing configuration errors.

3.4.2.3 Application Descriptor Composer

After collecting and structuring the dataset and service metadata, the OSR semantically validates the retrieved data-related services using internal ontologies and reasoning rules to ensure compatibility and compliance with the application context. It then proceeds to inject these data management services—such as the required data connector and auxiliary service offerings—into the overall application descriptor. The remaining microservices, which correspond to the core application logic, are typically defined by the user via the GUI and are preserved as initially specified. The final output is a standardized YAML-based application descriptor, which merges both user-defined microservices and OSR-composed data management components into a unified structure. This descriptor includes key deployment parameters such as image references, resource requirements, scaling rules, inter-service communication details, and SLAs. While the OSR injects required services derived from Piveau, it does not modify or overwrite any microservice configurations provided by the developer in the original application descriptor. Instead, it appends the relevant data management microservices, ensuring seamless integration.

This process provides a flexible and automated mechanism for integrating external data sources into AC³ applications, enabling developers to request access to datasets without having to manually define service connectors or networking requirements. Instead, the OSR retrieves this information from Piveau, dynamically assembles the appropriate microservices, and merges them into the application descriptor in a consistent, deployment-ready format. The result is a seamless bridge between high-level user intent and low-level deployment reality, achieved through semantic-driven discovery, catalogue integration, and runtime translation. This architecture also makes the OSR extensible. As more metadata properties and integration rules are added to the Piveau catalogue (e.g., for routing, clustering, or security), the OSR can be enhanced to process and reflect these changes without altering the core descriptor structure.

4 Data Connector

The main component that takes the role of providing access to the integrated data sources is called a Data Connector. The Data Connector is not a single piece of software but a concept that facilitates all the interactions needed to get access to the data available in the AC³ Catalogue and serve it to the applications of AC³.

As part of AC³, we have two types of data made available through the AC³ Catalogue: **hot data** from the IoT deployment of UC1 and **cold data** from the space telescopes generated in UC3. For these two types, we need to develop and use two different Data Connectors based on the EDC[5] pattern.

4.1 Hot Data Connector

The diagram in Figure 7 illustrates the data flow and interaction between data management components within the AC³ system, focusing on providing HOT streaming IoT sensor data using the Connector. It depicts a structured process for sharing data from IoT sensors via the provider Connector to a consumer Connector, enabling secure data transactions based on contract negotiations.

On the left side, the Data Source section contains IoT sensors that generate real-time data deployed at an IoT deployment. This data is transferred to a Connector (provider), which facilitates secure data sharing. The AC³ Catalogue serves as a registry for available data assets, where metadata such as dcat:endpointURL and dcat:assetName are published to enable discovery (as presented in Section 3.2). On the right side, the AC³ Application, which can be deployed at the edge or in the cloud based on the AC³ LCM, consumes this data. The Connector (consumer) retrieves asset metadata from the AC³ Catalogue through the OSR's generated application descriptor and initiates contract negotiation with the provider. Once an agreement is reached, the data transfer occurs via a predefined protocol (i.e., HTTP), with an optional receiver application handling the incoming data before passing it to the application for processing. The role of this receiver application is to take some of the load from the Connector itself and provide a more scalable solution based on the deployment of the application.

This architecture ensures controlled and secure data sharing between IoT data providers and consumers, leveraging standardized data exchange protocols and maintaining governance through **contract negotiation**. The contract negotiation process between the Connector (provider) and the Connector (consumer) ensures that data sharing is governed by pre-defined policies before any actual data transfer takes place. This process follows a structured sequence of interactions:

- First, the consumer requests access to a specific asset by querying the provider's Connector. The response to this request includes the policy ID, which outlines the terms under which the data can be shared. The policy is defined using the ODRL (Open Digital Rights Language) framework, ensuring that access control and usage constraints are explicitly stated.
- Once the policy is identified, the consumer initiates contract negotiation by referencing the policy ID and the asset name while specifying the provider's details. This triggers the contract negotiation process within the EDC framework, generating a negotiation ID that uniquely identifies the ongoing transaction.
- The consumer then periodically checks the negotiation status, polling the provider until the negotiation reaches a FINALIZED state. This step ensures that both parties agree to the contract terms before proceeding. Once finalized, a contract agreement ID is issued, confirming the successful establishment of the contract.
- Once the contract is confirmed and finalized, the consumer can access the requested data in compliance with the agreed policies and initiates the data delivery process to the provided receiver endpoint. This negotiation mechanism ensures transparency, security, and regulatory compliance, making data exchange within the AC³ framework both controlled and reliable.

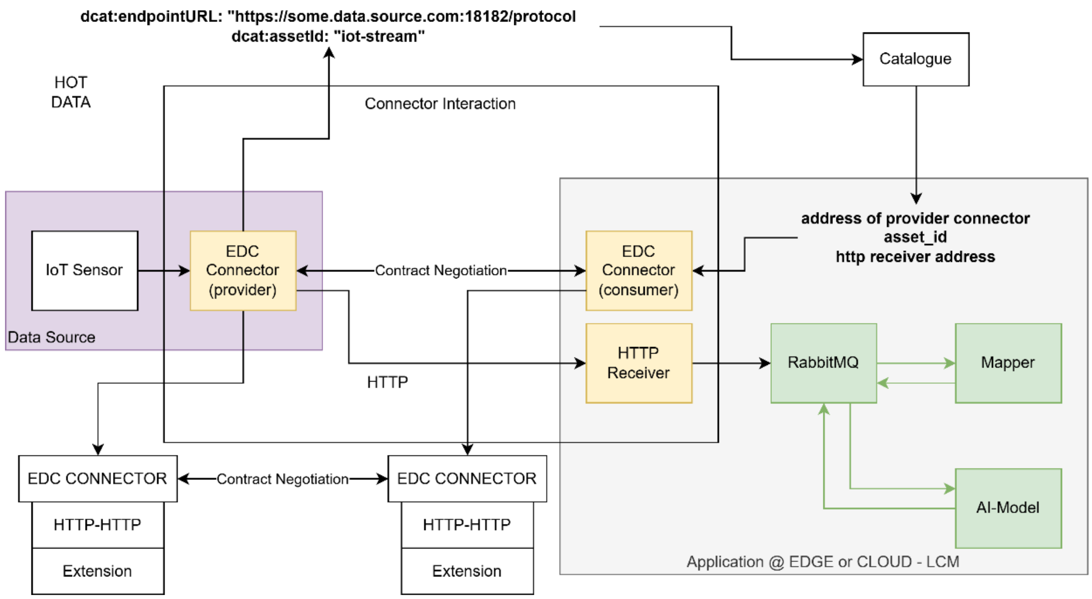

Figure 7 HOT streaming data connector interactions as part of an AC³ application

The implementation of the Connector is based on the operation of the Streaming HTTP to HTTP connector sample [8] and extends the default functionality by integrating a RabbitMQ message broker into the data flow. Instead of simply consuming and processing the data within the receiver, we have extended its functionality by forwarding the received data to a RabbitMQ endpoint. This enhancement enables asynchronous message handling, decoupling the data ingestion process from downstream applications. The HTTP receiver acts as a bridge, pushing the incoming stream to a designated RabbitMQ exchange or queue. This approach enhances scalability and reliability by allowing multiple consumers to process the data independently from the direct HTTP stream. By leveraging RabbitMQ, our implementation facilitates real-time event-driven data processing, ensuring efficient handling of high-throughput data streams. This modification makes it suitable for applications requiring distributed processing, buffering, or message-driven workflows while still maintaining the security and policy enforcement mechanisms provided by Eclipse EDC. Similar receivers can be implemented to forward the data generated to different backends or applications based on the needs of our users.

4.2 Cold Data Connector

The Cold Data Connector plays a crucial role in securely providing access to infrequently accessed data from remote sources. It is designed based on the Eclipse Data Space Connector [9], which follows a secure process for sharing data using contract negotiations to enable controlled data transactions. The primary Cold Data Connector includes the following functional assets:

4.2.1 Data Provider

The Data Provider manages the data, which could be stored in cold storage, such as a scalable cloud-based object storage service (e.g., IONOS S3). It implements a Connector that exposes endpoints for data access and metadata discovery, providing the required information, like endpointURL and assetName, in a registry like the AC³ Catalogue.

4.2.2 Data Consumer

The Data Consumer can be considered as a modular representation that is responsible for discovering, negotiating, and retrieving cold data assets from the Data Provider. It does so by interacting with the AC³ data

Provider Catalogue according to EDC standards and initiating contract negotiation processes for each data asset request. Notably, the data retrieval process in such a case is based on agreed-upon policies.

4.2.3 Contract Negotiation

The AC^3 Cold Data Connector follows EDC standards, including the Open Digital Rights Language (ODRL) for policy management and OAuth 2.0 for authentication and authorization. Contract negotiation is a crucial part of the Cold Data Connector, enabling data consumers to request and receive data access following EDC's contractual processes.

4.2.4 Metadata Management

The Cold Data Connector relies on the AC^3 use-cases' data model, which organizes data assets using the following entities: experiment, observation, dataset, file, and instrument. The Cold Data Connector facilitates metadata registration for data assets in AC^3 registries (similar to the AC^3 Catalogue) and enables data consumers to discover available data using the metadata descriptions.

4.2.5 Authentication

The AC^3 Cold Data Connector can employ other S3 providers' Signature Version 4 Authentication for secure data access to and from the IONOS S3 bucket, which enables the offering of advanced security features within the Other S3-compatible endpoint.

4.2.6 IONOS-S3 extension

The IONOS S3 project simplifies testing and development cycles for the Cold Data Connector. The AC^3 Cold Data Connector is currently being validated by implementing it on the AC^3 UC3 -- Nebula Astronomy Application to generate and store data assets over IONOS S3 object-storage. In that case, metadata descriptions for the UC3 data assets are then registered by the Cold Data Connector in the AC^3 Catalogue, enabling data consumers to discover them. Data consumers search and find relevant data assets by browsing the metadata descriptions from the catalogue. When a data consumer wishes to access certain data assets, they request this via the AC^3 Cold Data Connector, setting in motion a contract negotiation process based on ODRL policies. This process defines the conditions governing data sharing according to pre-established guidelines. If successful, a data transfer is established under agreed terms and conditions, securing controlled data sharing.

Lastly, the data consumer retrieves the requested data assets from cloud-based object storage using the AC^3 Cold Data Connector, following the agreed-upon conditions and secure data access mechanisms. Through these essential components and features, the AC^3 Cold Data Connector provides organizations and developers with a secure, efficient, and interoperable solution for data transactions while embracing other S3-compatible endpoints.

By seamlessly handling data sharing, metadata management, and policy-based control in conformity with EDC standards, the AC^3 Cold Data Connector is planned to optimize multiple use cases within the AC^3 framework, significantly enhancing value for organizations and developers.

5 Data usage and compliance with GDPR, GAIA-X and IDSA

AC^3 places a strong emphasis on the responsible use of data collected from IoT devices and data spaces, particularly in ensuring compliance with the General Data Protection Regulation (GDPR). Recognizing the complexities of modern data governance, security, and interoperability, AC^3 integrates Gaia-X principles, EDC Connectors, and the Piveau-based Data and Service Catalogue to establish a secure, transparent, and federated data-sharing ecosystem.

At the core of AC^3's data management framework is the Gaia-X federated model, which enables a decentralized and trust-based infrastructure where data sovereignty and privacy regulations are natively enforced. By adopting Gaia-X's principles, AC^3 ensures that data ownership remains with the providers, and access is granted based on well-defined, enforceable contracts. Compliance with GDPR is embedded at multiple levels, including data storage, transfer, and access management, ensuring that personal data is processed in accordance with lawful, fair, and transparent principles. To operationalize these compliance mechanisms, AC^3 leverages the EDC framework for secure data exchange. The Connector facilitates controlled and policy-enforced data transactions by enabling negotiable digital contracts between data providers and consumers. These contracts define usage rights, data access conditions, expiration policies, and revocation mechanisms in case of violations. While AC^3 cannot directly enforce compliance beyond the digital contract, the EDC framework provides mechanisms to terminate agreements if any misuse or non-compliance is detected, ensuring accountability within the ecosystem.

In addition to secure transactions, metadata standardization and data discoverability are enhanced through the integration of the Piveau-based Catalogue, developed within the German Gaia-X Lighthouse Project. This catalogue plays a crucial role in ensuring data accessibility, transparency, and interoperability across different stakeholders. It provides semantic metadata descriptions, enabling users to clearly understand the licensing terms, privacy restrictions, and compliance indicators associated with each dataset. By structuring datasets according to Gaia-X metadata models, AC^3 enhances usability, improves regulatory alignment, and ensures seamless integration into broader interoperable data spaces. Furthermore, AC^3 fosters a governance framework based on Gaia-X's trust and compliance mechanisms, promoting structured accountability between data providers and consumers. Governance mechanisms include automated policy enforcement, audit logs, and transparency tools that allow stakeholders to monitor and verify compliance with GDPR, licensing agreements, and contractual obligations. The federated approach ensures that data sharing follows pre-defined and secure processes, reducing the risks associated with unauthorized access, data breaches, or regulatory violations.

Finally, data access and exchange within AC^3 are enabled through the Connector as a core component of the Data Management Application Add-ons. This component establishes a digital contract with the Data Provider Connector, regulating all technical, legal, and operational constraints related to data transfer. The contract negotiation process, described in Section 4, ensures that data transactions are governed by enforceable rules, creating a trustworthy, secure, and GDPR-compliant data-sharing environment.

By combining Gaia-X's federated principles, EDC's secure data-sharing capabilities, and Piveau's structured data cataloguing, AC^3 not only aligns with the latest standards in European data sovereignty but also establishes a scalable, compliant, and future-proof ecosystem for data management.

Similarly, when comparing AC^3 with the International Data Spaces Association (IDSA) Reference Architecture Model [4], the AC^3 Data Management Platform-as-a-Service (PaaS) aligns closely with it, supporting the foundational principles required for secure, sovereign, and interoperable data exchange. At the heart of this compatibility is AC^3's strong focus on data sovereignty, ensuring that data providers retain full control over how their data is shared, accessed, and used. AC^3 enforces this through a combination of fine-grained access control, usage policy enforcement, data lifecycle management, and auditable logging. These capabilities map directly to the IDSA RAM's concept of sovereign data exchange, where the rights of the data owner are protected, and usage control policies are technically enforced throughout the data sharing process.

From an architectural standpoint, AC³ supports integration with IDS Connectors (or EDC Connectors), a key component defined by the IDSA RAM that facilitates secure communication and policy enforcement between participants in a data space. AC³ provides core services such as data ingestion, cataloging, metadata management, policy enforcement, and identity & access management (IAM)—each of which corresponds to functional components within the IDSA architecture. For example, AC³'s metadata and data cataloging capabilities align with the Metadata Broker and Vocabulary Provider services in IDSA, enabling semantic interoperability and data discoverability. The platform's support for policy-based access and usage control mirrors the functionality of the Policy Decision Point (PDP) and Policy Enforcement Point (PEP) in IDSA. Additionally, AC³ supports trusted identity management and federation through integration with identity providers, which is compatible with the Identity Provider and Certification Authority roles in the IDSA ecosystem. This comprehensive alignment enables AC³ to serve as a fully compatible participant within IDSA-compliant data spaces, empowering organizations to securely share data while preserving autonomy and compliance.

6 Conclusions

This deliverable concludes the development and integration of the AC³ Data Management Platform as a Service (PaaS), transitioning from the architectural foundation and initial implementations reported in D3.3 to a mature, fully operational system embedded within the Cloud-Edge-Client Continuum (CECC). The framework now enables seamless end-to-end management of data and services, supporting dynamic ingestion, semantic discovery, secure access, and automated deployment of data-intensive applications.

The data management architecture, as realized in this final stage, integrates all major components initially planned in D3.3. The AC³ Catalogue has evolved from a data-focused registry into a unified data and service catalogue, enabling developers to locate, interpret, and deploy both datasets and their required processing services. This is achieved through the adoption of enhanced metadata models built using SHACL, which provide detailed semantic descriptions aligned with DCAT-AP and Gaia-X specifications. The updated catalogue interface also offers a more intuitive and interactive experience for end users, facilitating more efficient navigation, filtering, and linkage between data assets and services.

The implementation of Hot and Cold Data Connectors has been successfully completed. The Hot Data Connector allows the real-time ingestion of streaming data from IoT devices and video feeds, while the Cold Connector, leveraging the IONOS S3 infrastructure, supports scalable access to historical datasets. Both types of connectors utilize the Eclipse Dataspace Connector framework to enable contract-based, policy-compliant data exchange between trusted entities, with identity and access management enforced via Keycloak.

These data connectors are now fully integrated with the lifecycle management tools of the AC³ platform, allowing services to be dynamically deployed and configured based on metadata-driven requirements. This integration ensures that services are instantiated alongside their associated data sources, improving automation, reducing configuration errors, and enhancing overall system usability.

The microservice-based addons developed for data handling—such as the Data Connector, Message Broker, Data Mapper, and Data Manipulators—have been successfully embedded into the AC³ application framework. These components support the internal processing of hot and cold data within deployed applications, enabling both real-time responsiveness and historical analysis.

In summary, the data management platform delivered in this final stage is modular, extensible, and compliant with emerging European data space principles. It supports real-world use cases in diverse domains and positions AC³ as a strong technological enabler of secure, intelligent, and federated data-driven services. The outcomes of this work fulfill the objectives set out in Task 3.4 and provide a solid foundation for the AC³ platform's continued integration, adoption, and scaling in operational CECC environments.

7 References

[1] "AC3 D3.3." Accessed: Jun. 18, 2025. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/11/AC3_D3.3_20240222_v1.0.pdf

[2] Data Spaces Support Centre, "Data Spaces Support Centre." 2025. [Online]. Available: https://dssc.eu

[3] G.-X. AISBL, "Gaia-X Architecture Document – Version 3.0." 2022. [Online]. Available: https://gaia-x.eu/resources/gaia-x-architecture-document

[4] International Data Spaces Association, "IDS Reference Architecture Model 4.0." 2022. [Online]. Available: https://internationaldataspaces.org/publications/

[5] H. Knublauch and D. Kontokostas, "Shapes Constraint Language (SHACL)." 2017. [Online]. Available: https://www.w3.org/TR/shacl/

[6] European Commission, "DCAT Application Profile for data portals in Europe (DCAT-AP), Version 2.0.1." 2019. [Online]. Available: https://joinup.ec.europa.eu/collection/dcat-ap

[7] "Create Custom Metadata Model - Documentation." Accessed: Jul. 19, 2025. [Online]. Available: https://doc.piveau.eu/guides/create-custom-metadata-model/

[8] "Samples/transfer/streaming/streaming-01-http-to-http at 3018fd09a9e55a710363ef733554bf51f4b1a4df · eclipse-edc/Samples." Accessed: Jul. 19, 2025. [Online]. Available: https://github.com/eclipse-edc/Samples/tree/3018fd09a9e55a710363ef733554bf51f4b1a4df/transfer/streaming/streaming-01-http-to-http

[9] *eclipse-edc/Connector*. (Jul. 19, 2025). Java. Eclipse Dataspace Components. Accessed: Jul. 19, 2025. [Online]. Available: https://github.com/eclipse-edc/Connector