20 DS2 Catalogue Module (CAT)

20.1 DS2 Catalogue Module (CAT)

Owner(s):	VTT
DOA Task:	T4.3
Tier:	3
Nature:	Optional
Results:	Outcome

This task will create a federation mechanism to enable different data spaces to interoperate. This task will orchestrate the lifecycle of data from data collection to data exchange, to data disposal/deletion across a federation of distributed data stores. The data lifecycle will include the establishment of a data contract between the data sources, the establishment of trust between entities in the data flow, and the adherence to data sovereignty and security requirements in the resulting federated data set. Other challenges will be addressed, such as accountability for use of purpose, the propagation of new domain-specific data restrictions (such as policy changes) across the federation, and methods for non-repudiable lineage across the lifecycle. Additionally, topics to reduce the latency involved in the transfer of huge amounts of data, such as caching, and data relocation will be investigated. This task will examine current, emerging technologies in this field, such as work being led by European Data Spaces and from the GAIA-X project as a basis for extension

20.1.1 Introduction

Purpose: The Catalogue module is a module designed to support the exchange of data within and across different data spaces. It ensures robust data governance, secure data exchanges, and compliance with sovereignty requirements. The main goal of DS2 catalogue is to enhance the functionalities of catalogue systems within existing reference architectures, enabling them to support both intra-data space and inter-data space operations. This includes defining data models for data product offers, data product offer searches, and interactions with members of other data spaces, thereby fostering collaboration across different data spaces. It is listed as an optional module since it is technically possible to handle this at participant level but this creates a lot of overhead for data consumers and providers, however in reality it is a core essential module of DS2 for most practical dataspace sharing scenarios.

Description: The core function of the module is to support creating data assets (Data Products), and to provide publication and search interfaces with associated metadata, and data model schemas to support validation. The module ensures that data products are appropriately created, described, and maintained within the catalogue. This includes defining metadata and access policies. It provides intuitive user- and technical interfaces for the publication and discovery of data assets so users can search for and access relevant data products efficiently. It also supports robust data models defined with schemas to ensure data integrity together with description of data service interfaces to access it (mainly based on IDSA reference architecture connectors). Key features of the module include trust building, governance compliance and Interoperability. Catalogues contribute to trust by ensuring that data products and their metadata are accurately described and reliably managed. Catalogues help enforce data governance policies by providing controlled access and visibility to data assets. By adhering to standardized data models like DCAT (Data Catalog Vocabulary, https://www.w3.org/TR/vocab-dcat-3/), catalogues ensure seamless data exchange and integration across various data spaces.

20.1.2 Where this component fits

20.1.2.1 Big Picture

This section outlines how the Catalogue Module integrates into the broader data space infrastructure, detailing its role and functionality within a single data space, across multiple data spaces, and as a potential intermediary.

The main role in DS2 is to implement catalog functionality usable between dataspaces that provides the catalog of data offerings and their associated metadata together with details of data service, especially connector self-descriptions based on DCAT vocabulary. DS2 has decided to base this initially on the IDSA reference architecture and in this instance the Metadata Broker Core

Besides data offerings, and depending on implementation, it may support the definition of catalogs of dataspace participants, various data schemas and data API's, catalogs of catalogs, and additional documentation etc. The catalogue module also provide interfaces for management of catalogs and these interfaces can be used by various other modules to provide additional information associated to offering to Data Providers and Consumers - for example to the Data Marketplace.

Some catalog functionality may also be deployed in the Connector providing DS2 compliant secure access for catalog with UI for Data Provider using catalog API. For example, in the EDC reference implementation, access to the catalog API can be implemented as a connector extension and DS2 provides a catalog UI component reference implementation using it. Optionally, a Data Provider may access a DS2 Catalog directly and relay required information to the connector.

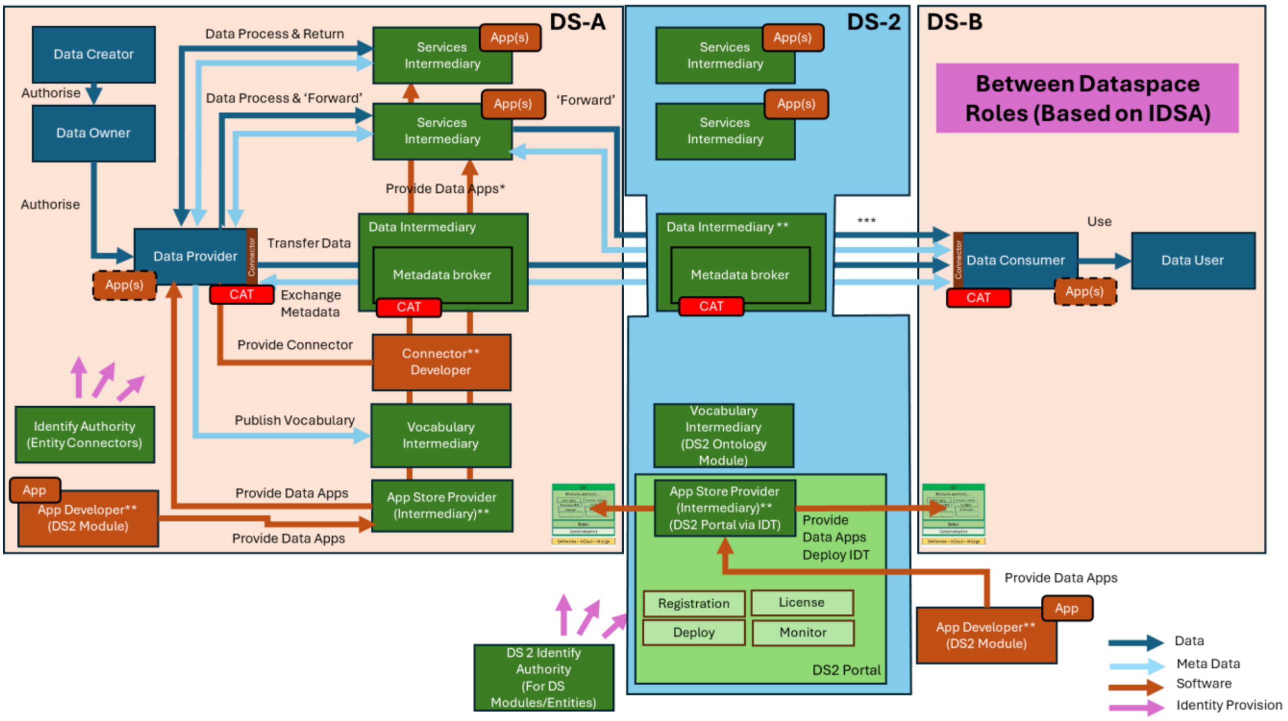


Figure 1: Module in the DS2 Architecture.

Where	Status
Within a single Dataspace for use between participants in that Dataspace only	Yes. Catalogue module can be used as metadata catalog for single data space, but this is not is not main role.
Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario	No
Across Dataspaces without Service Intermediary	Yes. Main role of Catalogue module is to support sharing between dataspaces such as providing catalog of dataspaces, or a catalog of shared offers from different dataspaces
Across Dataspace with Intermediary	Yes. Potentially could be performed by a Service Intermediary.
Other Comments	NA

20.1.2.2 Within a single Dataspace

The Catalogue module supports IDS Metadata Broker style services and catalogue user interfaces with extended metadata support within a single dataspace, but it is not its main objective in the project

20.1.2.3 Deployed and used by a single participant

Catalogue functionality is not usually needed by single participant.

20.1.2.4 Across Dataspaces without Intermediary(where applicable)

To support operations across different data spaces without an intermediary, it is crucial for the data space to provide information on which data spaces have agreed to collaborate with it. This collaboration indicates that the data spaces have made agreements to respect each other's rules, principles, and values.

20.1.2.5 Across Dataspace With Intermediary

A DS2 service intermediary acting across dataspace can be utilized to host the service catalog for both consumers and providers in different data spaces. This intermediary allows for the enactment of services without requiring prior knowledge of the participants involved. The process operates similarly to the use of a IDSA metadata broker.

20.1.3 Component Definition

The diagrams below illustrate the architecture of a Data Space Connector and its sub-modules, showing the internal structure and how the modules interact with each other.

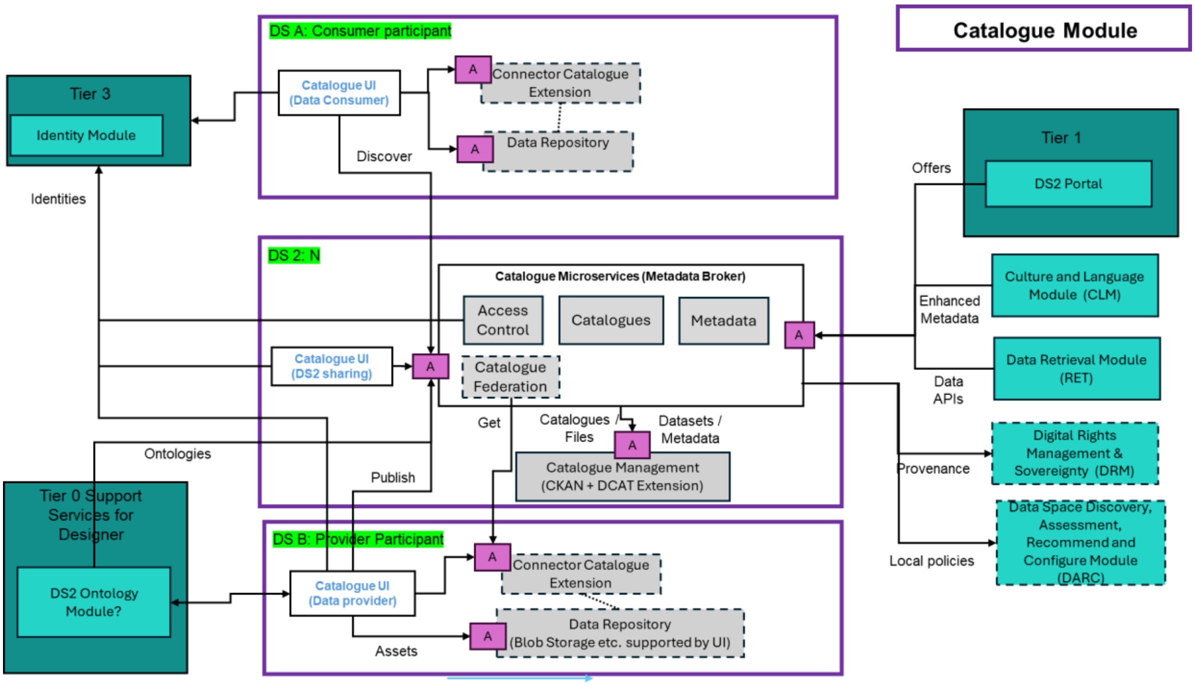

Catalogue Module
DS A: Consumer participant
Catalogue UI (Data Consumer)
A
Connector Catalogue Extension
Data Repository
Discover
Tier 3
Identity Module
Identities
Tier 1
DS2 Portal
Offers
DS 2: N
Catalogue Microservices (Metadata Broker)
Access Control
Catalogues
Metadata
Catalogue UI (DS2 sharing)
Catalogue Federation
Enhanced Metadata
Culture and Language Module (CLM)
Data Retrieval Module (RET)
Data APIs
Get
Catalogues / Files
Datasets / Metadata
Catalogue Management (CKAN + DCAT Extension)
Ontologies
Publish
Provenance
Digital Rights Management & Sovereignty (DRM)
Data Space Discovery, Assessment, Recommend and Configure Module (DARC)
Local policies
Tier 0 Support Services for Designer
DS2 Ontology Module?
DS B: Provider Participant
Catalogue UI (Data provider)
Connector Catalogue Extension
Data Repository (Blob Storage etc. supported by UI)
Assets

Figure 1: Schema for the Module

This module has the following subcomponents and other functions:

- **Catalogue UI**: Support different users of DS2 to access and manage DS2 offer catalogue. While shown as separate components this implemented as single web UI and backend component using the Catalogue Microservices API. Depending on participant and user role, UI provides different type of user experience:
    - **Catalogue UI for data providers**: Extends existing UIs provided by connector implementations with better support for DS2 metadata descriptions. Also supports sharing of offers into multiple dataspaces using Catalogue Microservices. May also provide support for creating metadata based on selected vocabularies.
    - **Catalogue UI for data consumers**: Extends existing UIs provided by connector implementations with better metadata descriptions and enables browsing and consuming offers shared by multiple dataspaces. Supports presenting user extended metadata associated with DS2 catalogue and offerings.
    - **Catalogue UI for data cross-dataspace data sharing**: Provides UI for supporting sharing of offerings between dataspaces by e.g. definition of catalogues of catalogues or interoperable data schemas and vocabularies.

- **Catalogue Microservices:** Contains extensible set of microservices providing Catalogue Module API for querying and managing catalogue offerings. Initially including:
    - **Access Control**: Using DS2 trust system services ensures authorization of what functionalities and data provided by other Catalogue microservices is the client UI allowed to access.
    - **Catalogues**: Using Open-Source catalogue interface provides creation and management DS2 specific catalogue types verifying against the data models defined in DS2 for interoperable data sharing cross data spaces.

- **Metadata**: Provides access to metadata related properties in catalogs and datasets for other DS2 components. Provides API functionality for inferred hierarchical ontologies for the UI component.
- **Catalogue Federation**: Supports IDSA Catalogue Protocol Specification to share catalogues in DCAT format. Service Can access catalogues (typically from connectors) combine them and provide catalogues using the protocol. This is optional because it may require participant access rights to the individual Dataspaces. Federation may also be supported through Catalogue Module UI in the connector by sending the connector catalogue directly to main DS2 catalogue. Provider UI has access rights to dataspace connector APIs and using catalogue extension the p2p federated catalogue of dataspace. Alignment with older version of IDSA catalogue data model and transformation between it and DCAT may also be provided by this component.

Microservices rely on functionality of lower-level APIs that can be already provided using well stabilized APIs of Open-Source components. In order to avoid vendor lock-in the API operations needed by CAT from these APIs are documented and can be seen as a reference specification if the underlying platforms need to be re-implemented with different components.

- **Catalogue Management:** Core operations on creating and managing catalogues and datasets in DCAT format together with resources (files) associated with them. While developed as reference implementation on top of management API provided by CKAN portals, this depends on selected set of API operations and specific configuration of CKAN platform, so that this subcomponent could be implemented with another platform if needed. Existing CKAN portals can also be used, potentially with restrictions on DS2 catalogue functionality. The CKAN API provides operations for managing datasets, resources, tags, organizations, and groups, allowing CAT microservices to create, update, delete, and search these objects within the catalogue. It supports user management, activity tracking, and offers flexible querying options for catalog management. Whilst CKAN provides a single catalog, separate logical catalogs can be managed by assigning them to different organizations or groups. DCAT extension helps to organize and expose datasets in a way that simulates multiple DCAT catalogs within one CKAN instance by leveraging organizations, groups, and tags. Each of these can be presented as distinct catalogs when publishing their metadata in the DCAT format. Implementing some of the functionality of DS2 catalogue may require specific configuration of CKAN or has to be implemented with CKAN extension mechanism so this is considered here as an internal component.
- **Connector Catalogue Extension:** Provides catalogue for IDSA information model-based description of catalogues, offers and agreements implemented as EDC connector extension. CAT optionally extends this existing component to support extended metadata required by DS2. Connector Catalogue Extension should support extended metadata also for IDSA federated catalogue protocol specification data model already implemented by EDC connector as described in next chapter.
- **Data Repository:** To provide metadata UI optionally supports selected set of repository platforms providing storage of resources that document what is offered by connectors. Documentation provided for data may be used to create extended metadata and data schemas. Suitable documents can be selected by user to be added to DS2 catalogue offer to be analysed for extended metadata. At minimum simple document file upload is supported by UI.

20.1.4 Technical Foundations and Background

Catalogue module should use existing Open-Source components as much as possible. There are two reference implementations for dataspace catalogues: IDSA (International Data Spaces Association) Metadata Broker and EDC (Eclipse Data Components). Each provides a framework for managing data exchanges and interactions within and across data spaces. As DCAT is used for new version IDS information model and EDC connector catalogue extension, it needs to be supported by catalogue. This can be implemented using commonly used open data platform CKAN and its CKAN extension API. CKAN also provides an robust API to manage logical catalogues and datasets in multi-user environment.

Data Space Support Centre blueprints provide guidelines related to Data Spaces. It tries to cover at high level both IDSA and Gaia-X approaches. The technical building block guidelines include data models and vocabularies for data exchange, and data service and offering descriptions using DCAT, publication and discovery, and marketplaces for data value creation. (https://dssc.eu/space/BBE/178422228/Technical+Building+Blocks)

IDSA Metadata broker core is implemented on top of Apache Fuseki RDF store, and ElasticSearch (https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol/overview/model). The component provides Web frontend for management. Connectors access the Broker using IDS message protocol that has been deprecated. Latest IDSA information model has decided to support DCAT based catalogue information model instead of proprietary model supported by latest Metadata Broker reference implementation. ODRL language is used for policy description directly instead of tailored IDSA information model version.

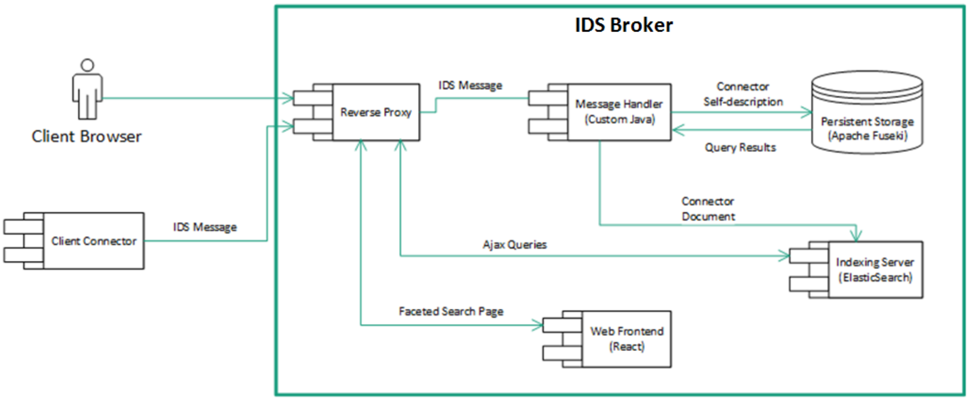


Figure 3: Functional Architecture of IDS (Metadata) Broker Reference implementation.

Eclipse Data Space Connector includes federated catalogue implementation (https://github.com/eclipse-edc/FederatedCatalog) and Catalog Protocol specification supporting it. It employs a set of crawlers, that periodically scrape the dataspace requesting the catalogue from each participant in a list of participants and consolidates them in a local cache. The Catalogue Protocol provided by EDC connector is defined by IDSA (https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol/overview/terminology#catalog-protocol) and defines a set of allowable message types that are used to request a catalogue from a catalogue Service.

CKAN is a tool for making open data websites. It helps you manage and publish collections of data. It is used by national and local governments, research institutions, and other organizations who collect a lot of data. There is an extension that provides plugins that allow CKAN to expose and consume metadata from other catalogs using RDF documents serialized using DCAT (https://extensions.ckan.org/extension/dcat/). It provides endpoints to manage a particular dataset as DCAT using various RDF serializations. Additionally, the extension also offers a catalog-wide endpoint for retrieving multiple datasets at the same time. DCAT can be extended for user defined vocabularies by storing metadata graphs but search for that data cannot be provided without modifying the extension. All CKAN core functionality (everything you can do with the web interface and more) can be used by external code that calls the main CKAN API. It provides means be used for example to store non-RDF documents such as Open API, or data schema files.

Subcomponent/Component	Owner	License
CKAN	Open Knowledge Foundation (OKF)	GNU AGPL version 3
CKAN Extension	OKF	GNU AGPS version 3
Metadata Broker	IDS (Fraunhofer IAIS)	Apache License 2.0
EDC Connector	Eclipse Foundation	Apache License 2.0

20.1.5 Interaction of the Component

The following table specifies the primary input/output controls/data to blocks which are not part of the module.

With Module/Feature	Receives from/Gives To	What
T6.3 Portal	Give To	When published on the portal information (technical, how tos etc) will be provided according to the general model
Tier 3 Identity Module	Receives From	Authenticity of participant information in participant-participant scenario
Tier 3 Identity Module	Give To	Participant identity
RET	Give To	Open API descriptions for data access
DRM	Give To	Store provenance information (optional)
DARC	Receives From	Configuration information such as based on local privacy rules, legislation (optional)
Culture and Language	Give To	Name and descriptive information from services and their bindings
Culture and Language	Receives From	Ontological alternative suggestions of the given information
Ontology Module	Receives From	Ontologies and vocabularies for designer and catalogue use

20.1.6 Technical Risks

Risk	Description	Contingency Plan
Integration Challenges	Difficulty in integrating the Catalogue Module with existing data space connectors and other components due to compatibility issues or differing standards.	Minimize needs to depend on external APIs. Support the new DCAT based IDSA catalog protocol. Develop adapter for alignment of catalog descriptions between old IDSA catalog format and DCAT.
Scalability Issues	The system may face performance bottlenecks as the number of data	Depend on DS2 platform capabilities for scalability. Restrict size of offer related

	products, participants, and transactions increase.	documentation stored in this module as resources.
Interoperability Issues	Difficulty in ensuring seamless catalogue data exchange between different data spaces due to varying standards and protocols.	Support primarily the IDSA catalogue protocol specification. Provide alignment service with older version of protocol.
Vendor Lock-In	Dependence on specific technologies or vendors may limit flexibility and increase costs in the long term.	Specify what API capabilities of COTS components (mainly CKAN) are used so they can be implemented with different underlying platform if needed.

20.1.7 Security

Security Issue	Description	Need
Catalog offering data security	Unauthorized access or data leaks could compromise sensitive data store in the Catalogue Module.	Implement identity and security mechanism defined in DS2.

20.1.8 Data Governance

Data Governance Issue	Description	Need
Data ownership and visibility	Ensure that catalogue can only be accessed with correct access rights	Catalogue descriptions should have clearly stated policies that Catalogue Module can validate.

20.1.9 Requirements and Functionality

This module will be used in the following use cases:

City Scape	✓
Green Deal	✓
Agriculture	✓
Inter-Sector	TBD

Catalogues are a core provision needed by any participant in a data space and for in general a dataspace level is a near-mandatory necessity. A data space cannot exist without a catalogue, as it is essential for managing and facilitating the exchange of data within and across different data spaces. The catalogue provides a centralized repository for data product offers, member information, and common data space offers. It enables the discovery and access to data products and services, ensuring robust data governance, secure data exchanges, and compliance with sovereignty requirements. Because of this fundamental role, the use case for the catalogue module is clear and indispensable across various domains.

Their requirements and functions/extensions to achieve them relative to this module, specifically extracted from the use cases, are as per the table below. . Note that Catalogue Module typically provides pilots the same technical APIs, the pilot requirements are mostly seen in what data catalogue metadata should be able to model.

WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
Section 2.2 UC1.1	Not clearly defined, needs clarification	Determine specific points of integration and data orchestration requirements	R & D	M
Section 2.2 UC1.2	Not clearly defined, needs clarification	Determine specific points of integration and data orchestration requirements	R & D	M
Section 2.2 UC1.3	Sharing and gathering data from multiple sources and sectors	To orchestrate the data from the sources to one location	R & D	M
	Use Case 2: Green Deal			
Section 2.2 UC2.1	Relevant data sources to be obtained from both data spaces within the use case	To orchestrate the data from the sources to one location	R & D	M
Section 2.2 UC2.2	Relevant data sources to be obtained from both data spaces within the use case	To orchestrate the data from the sources to one location	R & D	M
Section 2.2 UC2.3	Not clearly defined, needs clarification	Determine specific points of integration and data orchestration requirements	R & D	M
Section 2.2 UC2.4	Not clearly defined, needs clarification	Determine specific points of integration and data orchestration requirements	R & D	M
Section 2.2 UC2.5	Not clearly defined, needs clarification	Determine specific points of integration and data orchestration requirements	R & D	M
	Use Case 3: Agriculture			
Section 1.1.4	Data Integration and Accessibility”	Facilitating seamless integration	R & D	M
Section 2.2 UC3.1	Movement of data between different parties and processes	To achieve efficiency	R & D	M
Section 2.2 UC3.2	From fruit sourcing to forecasting/crop management	To achieve efficiency	R & D	M
Section 2.2 UC3.3	Based on Crop Productivity	To achieve efficiency	R & D	M

20.1.10 Workflows

The following sub-sections describe the sequence diagrams of the Catalogue Module. It is assumed that single dataset offers catalogue is created into CKAN under DS2 organization topic using CKAN API. All features here assume this catalogue is already created.

The Catalogue module is a passive component. It does not call other DS2 components than those providing identity and security infrastructure services for the DS2 platform.

If active workflows need to be supported, Catalogue Module could offer a subscription API for consumer components to receive events when Catalogue or a dataset is updated.

20.1.10.1 Request a catalogue

This feature provides the capability to get the whole catalogue, optionally filtered by a query. Note this can be used also by Catalogue Module to create a federated catalogue. Figure 11 shows the sequence diagram of this feature.

Main Steps/Functionalities:

- Catalog Request
- Fetch Catalog from CKAN
- Catalog ACK or ERR response

The Catalog Request is message sent by a Consumer to a Catalog Module REST API. The Catalog Module must respond with a Catalog, which is a valid instance of a DCAT Catalog or error message.

Components Involved:

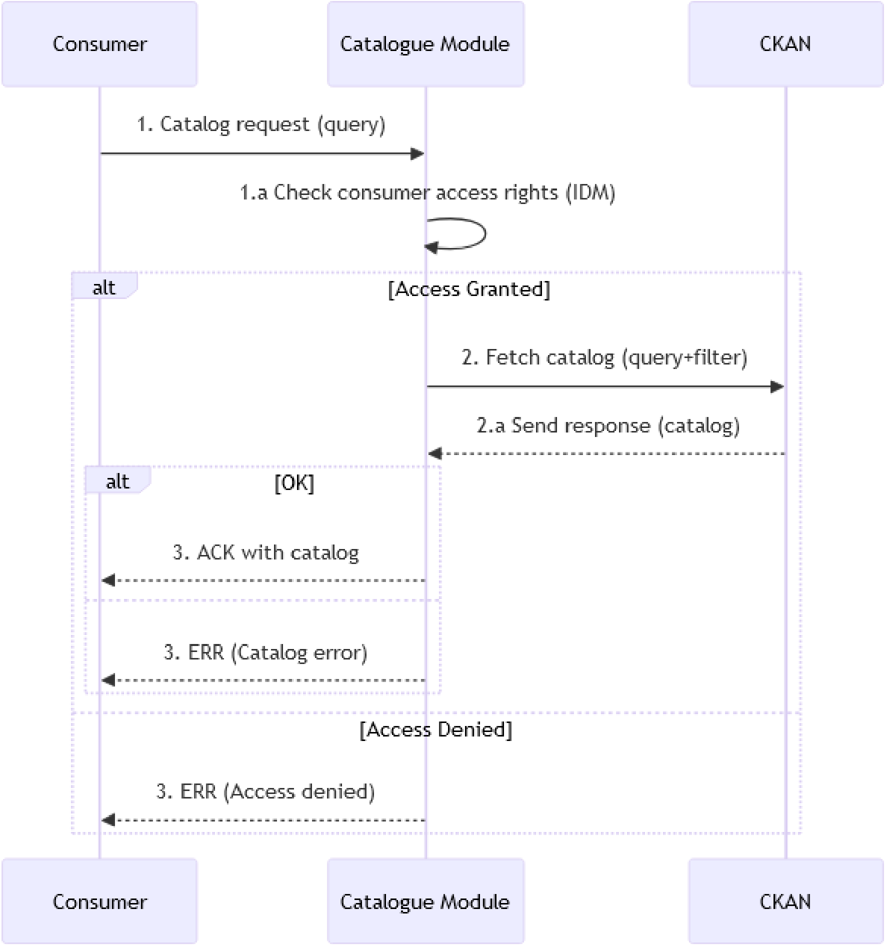
- Consumer component such as any DS2 component or another Catalogue module or connector for catalogue federation
- Catalogue Module

Data:

- The message may have a filter property which contains an implementation-specific query or filter expression type supported by the Catalog Module (types of queries TBD).

Sequence Steps:

1. Consumer component calls request catalogue with query parameters adding filter to only show publicly available offering datasets
    a. Catalogue module checks access right of consumer
2. If no query is defined, Catalogue Module calls DCAT API to provide whole Offer catalogue filtered to show only public offers.
3. Catalogue Module returns either ACK response with Catalogue in DCAT format or ERR error message to consumer



20.1.11 Figure 11: Sequence Diagram for Catalog Request

20.1.11.1 Request a dataset

The Dataset Request is sent by a Consumer to a Catalog Module REST API. The Catalog Module must respond with a Dataset, which is a valid instance of a DCAT Catalog or with an error message.

Main Steps/Functionalities:

- Dataset Request
- Fetch Catalog from Catalogue management (CKAN)
- Dataset ACK or ERR response

Components Involved:

- Consumer component such as any DS2 component or Connector
- Catalogue Module

Data:

- The Request must have a dataset property which contains the id of the Dataset.

Sequence Steps:

1. Consumer component calls request dataset with id of dataset
    a. Catalogue module calls DCAT API using id of dataset to get dataset
2. Catalogue module check that consumer has read access rights for the dataset
3. Catalogue module returns either ACK response with dataset in DCAT format or ERR error to the consumer.

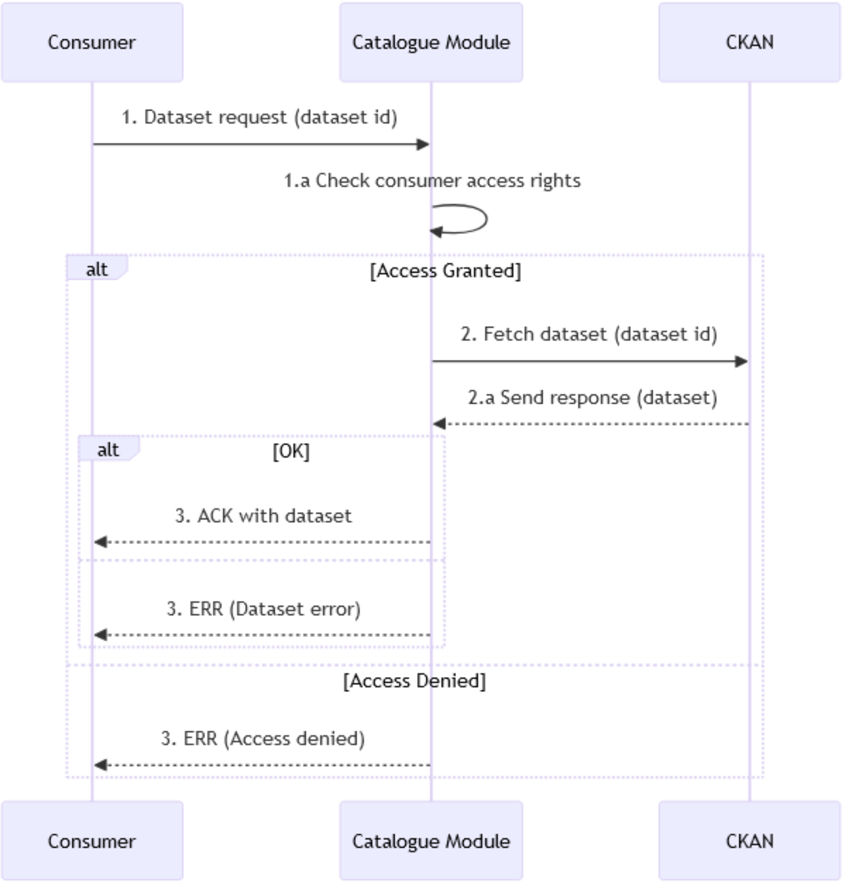


Figure 12: Sequence Diagram for Dataset Request

20.1.11.1 Dataset CRUD (Create, Read, Update, Delete)

Dataset CRUD feature is more fine-grained set of operations for managing datasets. The Dataset CRUD operations are sent by a Consumer to a Catalog Module REST API.

CRUD operations of datasets require using CKAN core API instead of DCAT extension API as the latter only provides read operations on the datasets.

The Catalog Module must check that consumer has the rights for these operations on dataset provide modification access, and also show hidden or inactive offerings visible only to the owner of dataset.

Components Involved:

- Consumer component such as any DS2 component, Connector or Catalogue UI component
- Catalogue Module

Data:

- The requests must contain the id of the Dataset operation is targeted.

Sequence Steps:

1. Consumer component calls CRUD operations for dataset with id of dataset
    a. Catalogue module checks the access rights of consumer for that dataset
2. Catalogue module calls CKAN API to perform the CRUD operation on dataset
3. Catalogue module returns either ACK response with current version dataset in DCAT format or ERR error to the consumer.

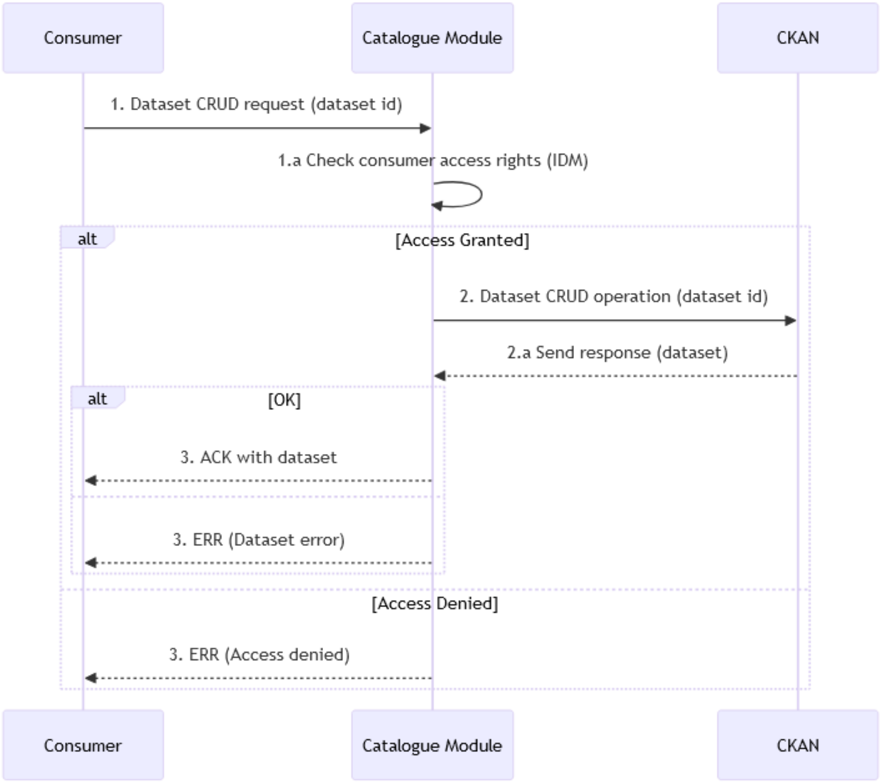


Figure 12: Sequence Diagram for Dataset CRUD

20.1.11.2 Catalogue metadata (Create, Read, Update, Delete)

Catalogue metadata feature provides means to modify Catalogue metadata information. Only the metadata property of catalog description can be accessed and modified with this.

The Catalog Module must check that consumer has the rights for these operations on dataset provide read or modification access.

Main Steps/Functionalities:

- Metadata CRUD Request
- Perform operation using CKAN API
- Dataset ACK or ERR response.

Components Involved:

- Consumer component such as any DS2 Language (has modification rights), Connector or Catalogue UI component (have read rights)
- Catalogue Module

Data:

- Metadata read or updated in JSON-LD graph format.

Sequence Steps:

1. Consumer component calls CRUD operations for Catalogue metadata
    a. Catalogue module checks the access rights of consumer for metadata.
2. Catalogue module calls CKAN API to perform the CRUD operation on catalogue metadata.
3. Catalogue module returns either ACK response with latest version of version metadata in JSON-LD format or ERR error to consumer.

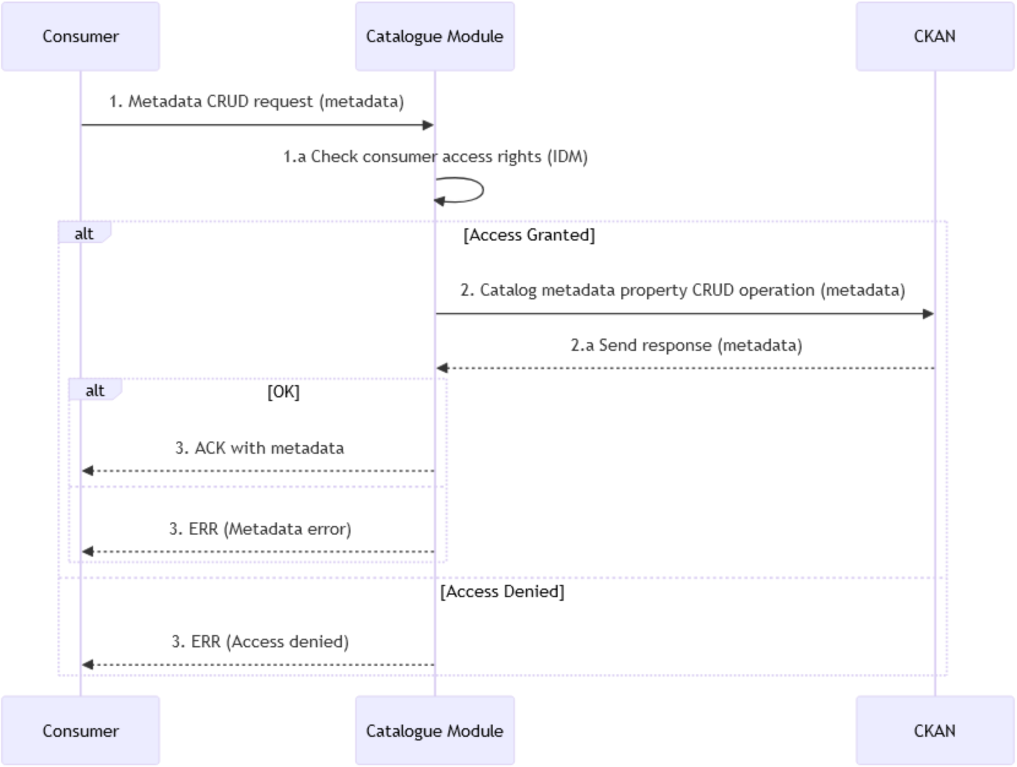


Figure 12: Sequence Diagram for Catalogue metadata

20.1.12 Role, Resourcing, and Milestones (skip for now)

Sub-component	Main Activity	M18	M24	M30	M36
Catalogue Management	Configuration of CKAN Server with DCAT extension. Provides initial catalogue UI.	■			
Catalogue Management	Develop potential extension needed to support DS2 functionality with CKAN		■	■	
Catalogue Microservices (framework)	Common framework for microservices that use CKAN API and provide external API for CAT	■			
Catalogues	Develop microservice providing DS2 catalogue API operations. Develop schemas to verify data inserted catalogue. Develop required data transformations.		■	■	
Metadata	Design and implement data model and API operations to manage RDF based metadata for catalogue and datasets.		■	■	
Access control	Take into use access control functionality supporting DS2 identities, integrate into microservices framework.		■	■	
Catalogue Federation	Microservice to read EDC connector catalogues and add to main catalogue, provide EDC and IDSA compatible catalogue protocol API endpoint			■	
Connector Catalogue extension and Data repository	Select and run a test configuration of EDC connector with repository.	■			
Catalogue UI (DS2 Sharing)	Develop UI for catalogue and access management, before functional CKAN UI can be used		■	■	
Catalogue UI (Data provider)	Select metadata vocabularies and provide UI support form to insert metadata into catalogue offerings.	■	■	■	
Catalogue UI (Data consumer)	Browse catalogues and metadata, data consumer can use CKAN UI before this is available.		■	■	
CAT	Integration with set of simulated users to ensure scalability			■	■
Table Total/DOA Task Total/Resilience	**Comments:**				

20.1.13 Open Issues

The following table summarise open issues/uncertainties that need to be resolved during the next stages or implementation.

Issue	Description	Next Steps	Lead or Related Component
Support module interfaces	Final design and at least some Implemented of block chain/DARC/Ontology modules. These are not key/blockers but if opportunity could be useful for the module	Wait for final design/implementations	WP3: Block Chain & WP5: DARC and Ontology