2 Culture and Language Module (CLM)

2.1 Multi-cloud Module (MCL)

Owner(s):	DIGI
DOA Task:	T6.1
Tier:	0
Nature:	System
Result:	Outcome

DS2 Multicloud

This task is two-fold: a) It will create the underlying infrastructure to allow for the efficient transfer of vast amounts of data between data stores. This includes novel solutions such as intelligent data placement and caching of data (in conjunction with T4.1), the establishment of secure connectivity between distributed data stores and observability of created data pipelines to monitor threats to data quality during transfer (in conjunction with 4.2). b) Alongside this it will provide an experimental data space as a service for companies interested in data economy and data sharing. The experimental data space infrastructure will be used as a baseline for developing customized data space solutions for specific application domains or use cases. This allows the testing of business concepts and building of proof of concepts before investing in real-life solutions. The DSIL implements the key components of International Data Spaces (IDS) reference architecture and is part of the VTT Data Space Innovation Lab (DSIL).

2.1.1 Introduction

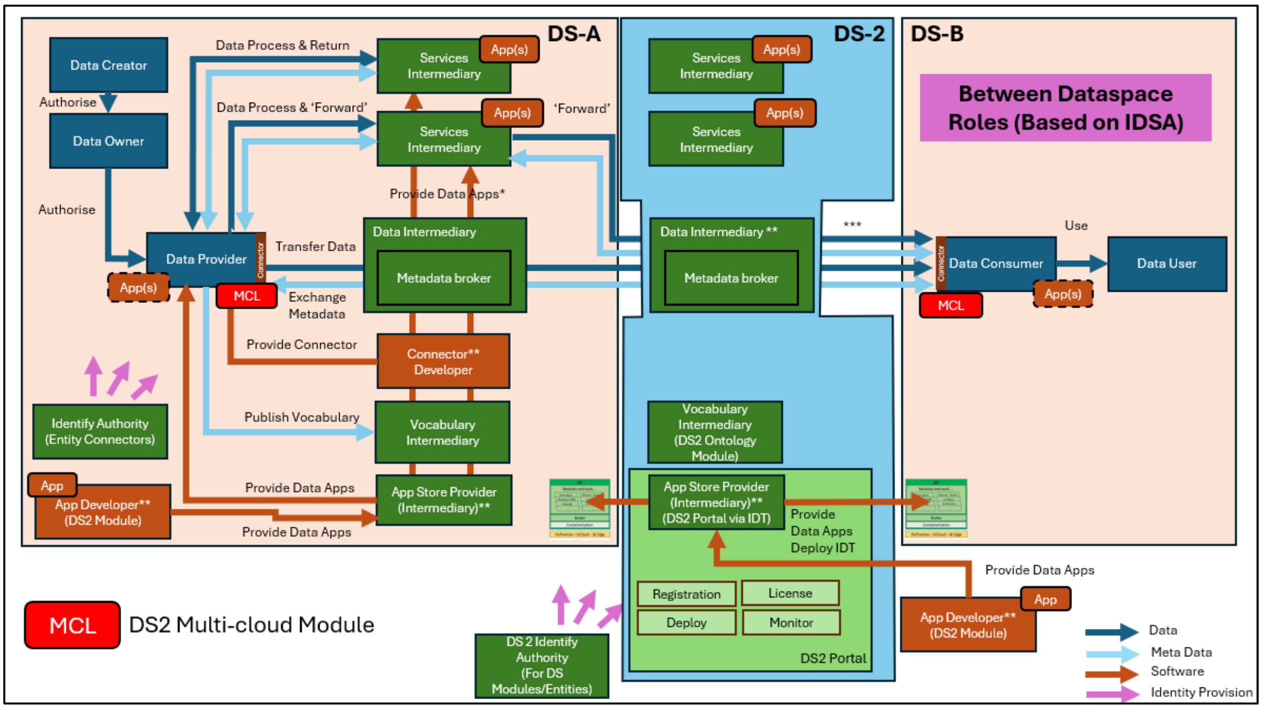
Purpose: The DS2 Multi-cloud module (MCL) enables efficient transfer of discreet data, vast amounts of data, and streaming data between participants of dataspaces from data stores that are distributed across multi-cloud storage infrastructure. MCL includes intelligent data placement and caching at dataspace provider participants with a dataspace consumer participant requesting such data and provide services through use case applications(s). It will also ensure data exchange happens over secure connections using the DS2 Security Module (SEC).

Description: When a use case application initiates a request for data through a dataspace consumer participant, the module ensures that the requested data is swiftly and accurately delivered from discovered and relevant provider participant(s). This process involves intelligent data placement by analysing access patterns and data requests (based on specific parameters) from the consumer participant that allows selecting the optimal data caching locations and employing predictive caching strategies to enhance data availability and retrieval speeds. In addition to push/pull style data sharing, this module introduces two novel extensions of the dataspace connector data plane for vast amount and streaming data sharing. Furthermore, it incorporates secure connectivity ensuring that all data exchanges are protected in transit. The module also aligns with the broader DS2 architecture, ensuring interoperability and synergy with various other modules and their sub-components. This enables the module to support a wide range of application scenarios and data exchange requirements.

2.1.2 Where this component fits

2.1.2.1 Big Picture

The module covers efficient data sharing between dataspaces where within a provider participant of a dataspace, the data is presumed to be stored in multi-cloud infrastructure with is common in industries to day). The sub-components of this module are shown below.



Where	Status
Within a single Dataspace for use between participants in that Dataspace only	MCL could potentially be used within a single dataspace to share data and broadly speaking the approach would be the same as with Across dataspace without Service Intermediary. However, due to the nature of DS2, the implementation and validation is not focused around this.
Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario	Since the module facilitates data sharing, it can be deployed and used by a single dataspace participant. However, from a broad perspective and the context of the DS2 project, such deployment and validation will not be conducted in DS2.
Across Dataspaces without Service Intermediary	Yes – MCL is developed for data sharing through one or more dataspaces to a consumer participant.
Across Dataspace with Intermediary	N/A
Other Comments	N/A

2.1.2.2 Within a single Dataspace (where applicable)

N/A

2.1.2.3 Deployed and used by a single participant (where applicable)

N/A

2.1.2.4 Across Dataspaces without Intermediary (where applicable)

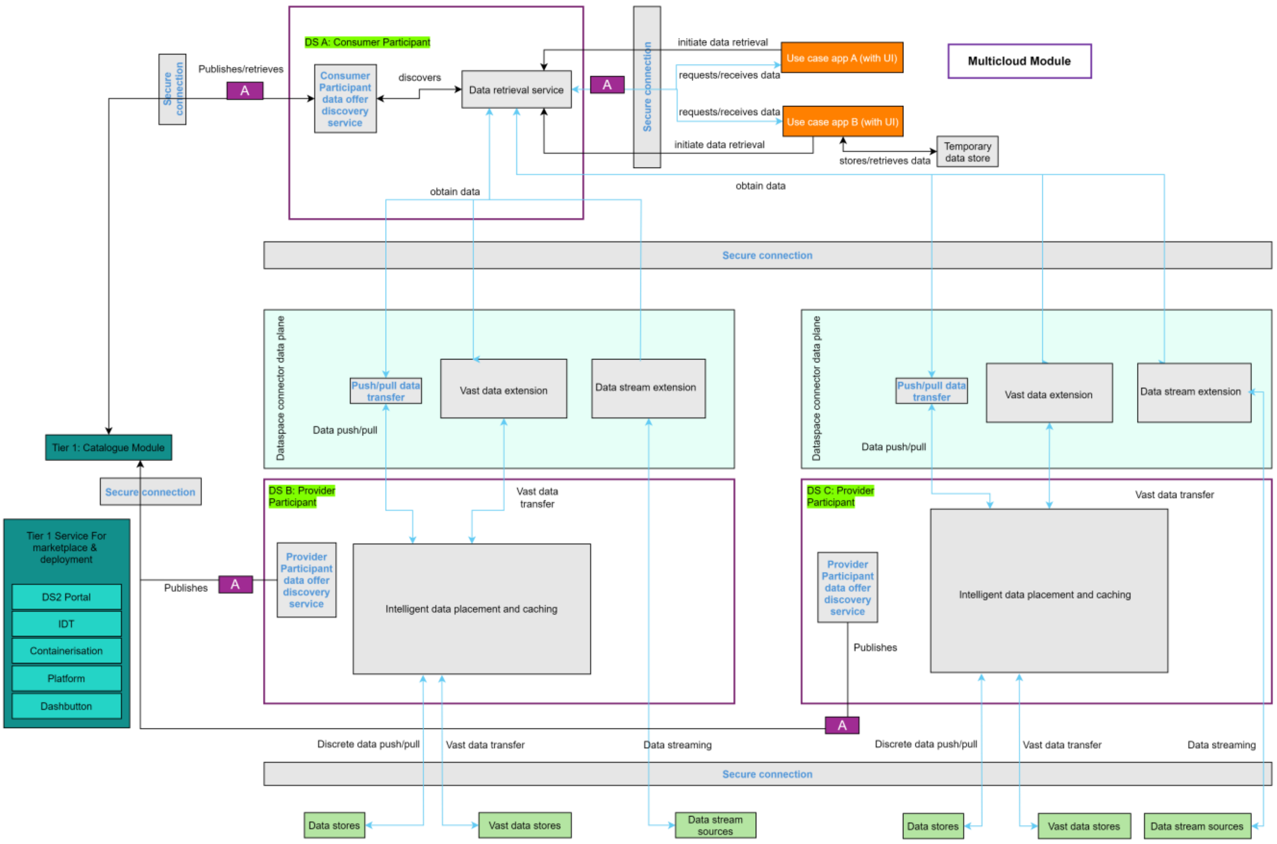
The module is specifically developed for operating across dataspaces enabling efficient data sharing from data stores through multiple dataspaces. When a dataspace consumer participant initiates a request for data for use case applications, MCL ensures that the requested data is efficiently delivered through one or more dataspace provider participant(s).

2.1.2.5 Across Dataspaces with Intermediary (where applicable)

N/A.

2.1.3 Component Definition

The figure below represents the actors, internal structure, primary sub-components, primary DS2 module interfaces at a high level.



When a use case initiates data retrieval request, it queries the data offer discovery service to determine which provider participant(s) can serve the requested data. To support vast amount and streaming data, two novel sub-components to the dataspace connector data plane are introduced. The intelligent data placement and caching sub-component analyses access patterns ensuring that frequently accessed various data stores (stored in multi-cloud infrastructure) is cached closer to the consumer participant

The figure below represents the actors, internal structure, primary sub-components, primary DS2 module interfaces, and primary other interfaces of the module at a detailed level.

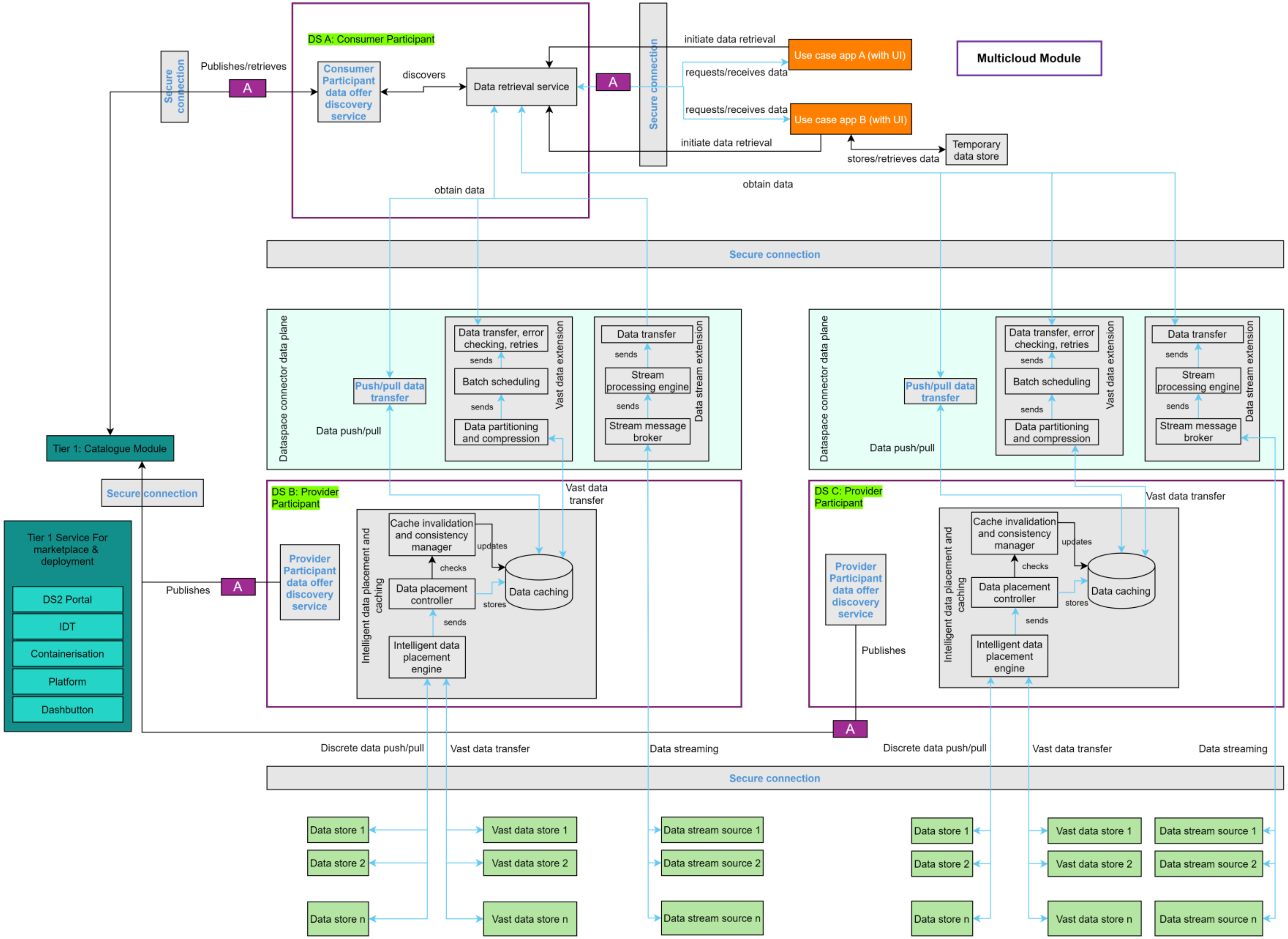


Figure 13: Schema for the Module

This module has the following subcomponents and other functions:

- **Tier 1: Catalogue module**: It securely stores a description metadata of the dataspaces and implements an interconnected search and retrieval system for a consumer participant to discover data and then relevant provider participant(s) end points.
- **Use case app:** These are the high-level applications of users such as those in the use cases of DS2. Such applications can require data from multiple data stores, which are presumed to be distributed to multi-cloud data storage infrastructures. Also, the data sharing happens through multiple dataspace provider participants. Each use case can directly consume the obtained data and/or temporarily store them in a local storage for combining the data arriving in batches (in case of vast data transfer) or processing in future.
- **Temporary data store:** A local data storage which are used by the use case apps to store data for (very) short term.
- **Secure connection:** This sub-component is responsible for ensuring data sharing takes place over secure connections (e.g., VPN, SSL/TLS) and comes from the DS2 Security Module (SEC).
- **Dataspace connector (data plane):** It facilitates secure and efficient transfer of data between participants in the DS2 ecosystem while ensuring compliance with agreed-upon data governance policies and handling data routing. In DS2, the data place supports three types of data sharing – discrete data, vast amounts of data, and streaming data.
- **Tier 1 Service for marketplace and deployment:** The full stack will be implemented as generically described elsewhere in this document.

Dataspace consumer participant:

- **Consumer Participant data offer discovery service:** This sub-component performs two tasks – (a) Publishes a description of the data offer of each participant to the Tier 1 catalogue module (for a centralised discovery) and (b) Enables a consumer participant within the DS2 ecosystem to discover the data offers for the use case applications. Furthermore, the data offer is published in the form of metadata (e.g., data type such as discrete, vast amount, or streaming data, accessibility conditions). This service exists as a background in DIGI's cloud-based Paradise platform and will be adopted to store and search metadata of participant data offers.
- **Data retrieval service:** This sub-component is triggered by the use case applications which require access to data stored in multi-cloud data stores. The dataspace consumer participant performs a discovery of the available provider participants and then proceeds to the retrieval of data stored from various data stores. This service allows dataspace consumer participants to request and obtain data needed for specific use case applications seamlessly. This service allows both push and pull type of data retrieval, supports multiple data type (such as discrete, vast amount, and streaming), and queries. Additionally, it ensures that data integrity and consistency are maintained throughout the retrieval process, providing reliable data access to user applications. The retrieved data may be directly used in such an application or may be temporarily stored in a data store. Integrity checks (hash verification) are performed especially for vast data transfer done through batches.

Dataspace provider participant:

- **Provider participant data offer discovery service:** It publishes a description of the data offer of each participant to the Tier 1 catalogue module for a centralised discovery by the consumer participant.

- **Intelligent data placement and caching:** This aims to optimise data storage and retrieval by strategically placing data across multi-cloud storage locations by employing predictive caching mechanisms. This component analyses access patterns from dataspace consumer participants and predict future data requests (based on specific parameters), ensuring that frequently accessed data is cached closer to the consumer participant. By doing so, it reduces latency and improves data retrieval speeds. The intelligent data placement strategy ensures optimal use of storage resources by distributing data based on access frequency, storage costs, and performance requirements. This sub-component is comprised of:
    - **Cache invalidation and consistency manager:** In case of data changes, this sub-component acts to invalidate or update outdated cached entries ensuring that stale data is not transferred. It will implement strategies like time-to-live settings, write-through caching, or cache coherence protocols.
    - **Data placement controller:** It enforces the decisions made by the data placement engine and handles the actual data movement in storage locations for data caching.
    - **Intelligent data placement engine:** This sub-component uses intelligent algorithms to determine where data should be cached. The placement decision is taken based on latency, access patterns from the consumer, data size, cost, network bandwidth, and other relevant factors.
    - **Data caching:** It covers the process of storing copies of frequently accessed data in a location closer to the consumer participant and the actual storage.

Dataspace connector data plane: In DS2, it supports three types of data transfer:

- **Push/pull data transfer:** Data can be delivered to consumer counterpart through this sub-component either via a push model (where the dataspace sends data automatically at intervals or when triggered) or a pull model (where consumer participant requests data when needed). While this is supported by default in the dataspace connector, it is needed to accomplish discrete or small amounts of data transfer.
- **Vast data extension:** Today, the dataspace connector data plane typically supports push-pull style of data transfer. DS2 introduces a novel sub-component called vast data extension to the data plane. It is designed to handle sharing of extremely large datasets stored across distributed data stores. It is composed of:
    - **Data transfer, error checking, retries**: Using this, the batch of data is transferred from a provider to a consumer participant. During the transfer, the provider side monitors for any errors or interruptions such as network connection lost. In case of an error detected, the sub-component retries the transfer or attempts to resume from the point of failure.
    - **Batch scheduling**: This sub-component schedules batch data transfers to run at specific times (e.g., every hour). The schedules are automatically developed based on the use case application’s needs. Scheduling can also occur during off-peak hours (should the use case need it) to minimise impact of the network and DS2 resources.
    - **Data partitioning and compression:** Extremely large datasets are split into smaller chunks or batches which are then compressed to save network bandwidth.
- **Data stream extension:** This extension provides robust capabilities for handling real-time data streams within the DS2 architecture for use case(s) that require(s) data

streaming. This is a novel sub-component introduced by DS2 for the dataspace connector data plane where data streaking is minimally supported in connectors today if at all. This component enables continuous data flows from various sources, such as IoT devices, sensors, and real-time applications. It supports high-throughput and low-latency data pipeline such as Apache Kafka, ensuring that streaming data is handled efficiently and reliably. This extension is composed of:

  - **Data transfer**: Using this, the stream of data is transferred from the provider to the consumer participant.
  - **Stream processing engine**: Processes continuous streams of data in (near) real time ensuring uninterrupted data flow, stateful processing capabilities, and ensure the stream is in right format needed by the use case applications. It can also be used for relatively simple data transformations and performing operations like data aggregation (if needed).
  - **Stream message broker:** The message broker serves as a buffer and pipeline between the streaming data sources and consumer participants. Data sources publish data to message queues or topics that act as the entry point to the streaming pipeline.

- **Data store:** These data stores represent data storage at the participant. Each participant data store may be designed to manage a variety of data types and formats, providing a robust and scalable storage solution. The subcomponent architecture supports multiple data stores leveraging multi-cloud environments, enabling data to be distributed and replicated across different geographic locations, enhancing accessibility and redundancy.
- **Vast data store:** Similar to the data store mentioned above, these are specific to storages with vast amounts of data.
- **Data stream source:** This refers to data sources that continuously produce data, such as IoT devices, sensors (such as video cameras), and real-time applications (e.g., weather apps).

2.1.4 Technical Foundations and Background

The above components will be buit upon the following:

- Eclipse dataspace connector will be used for data sharing and is one of the main components in the Figure 13. Further research will be done on how to introduce the two extensions in the data plane of this connector.
- Apache Kafka will be used for supporting data streaming. Kafka is a lightweight, scalable solution that includes stream message broker (ZooKeeper). In addition to that, Kafka stream will be used as the stream processing engine. It is also a lightweight framework and integrates seamlessly with Kafka.
- Apache Hadoop and Cassandra will be exploited for data caching. Hadoop distributed file system (HDFS) is a proven, scalable, fault-tolerant system for storing large amounts of data while Cassandra is a highly scalable NoSQL database that offers decentralised data distribution across multiple locations, making it suitable for intelligent placement across cloud and edge environments.

Subcomponent/Component	Owner	License
Eclipse dataspace connector	Open source	Apache License 2.0
Apache Kafka	Open source	Apache License 2.0
Nginx	Open source	Apache License 2.0
Hadoop	Open source	Apache License 2.0

Cassandra	Open source	Apache License 2.0

2.1.5 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
DS2 connector	Give To	Allows extensions for vast amount and streaming data transfer over a secure connection
Catalogue module	Give To	Metadata for provider data offer (GET from consumer)
Catalogue module	Receives from	Metadata for provider data offer (POST from provider)
Use case app	Give To	Initiates data retrieval request
Use case app	Receives from	Data from data retrieval service

2.1.6 Technical Risks

Risk	Description	Contingency Plan
Dataspace connector extension leads to non-interoperability	The two proposed extensions (vast data and streaming data) lead to non-interoperability with currently deployed dataspaces	Extensions will be developed as a modular service allowing the connector to function as is.
Discovery service not able search for provider participant dataspaces	Consumer participant discovery service not able to find provider participant at the beginning of the DS2 ecosystem deployment.	The discovery service will be exposed with a default URL, known to the consumer participant to initiate the data offer discovery process.

2.1.7 Security

Security Issue	Description	Need
Data leakage from the data stores	Unintended data leaks noticed from the data stores.	Security module will be used to store the data securely at rest.

2.1.8 Data Governance

Data Governance Issue	Description	Need
Connector data plane extensions	Multi-cloud module introduces two new extensions for supporting vast data and streaming data through dataspace connectors.	Multiple parties may contribute and share data through the two extensions, leading to unclear definitions of who owns the data and is responsible for its governance.

2.1.9 Requirements and Functionality

This module will be used in the following use cases:

City Scape ✓

Green Deal	✓
Agriculture	✓
Inter-Sector	✓

Their requirements and functions/extensions to achieve them relative to this module, specifically extracted from the use case are as per the table below noting that in many cases further discussion might need to take placed between pilot partner and module partner to determine if a fit or the scope of the precise fit.:

WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
Section 2.2 UC1.1	“share the data on energy consumption and other patterns because I want to save money”	Across dataspaces data sharing over secure connectivity.	R & D	M
Section 2.2 UC1.1	“Receive consumption information to optimize the production and create more attractive packages.”		R & D	M
Section 2.2 UC1.1	“Get data on awareness and value put on energy effectiveness to create better products / fit the needs”		R & D	M
Section 2.2 all UCs	“Sharing and gathering data from multiple sources and sectors”		R & D	M
	Use Case 2: Green Deal			
Section 2.2 UC2.1	“Data is shared and accessible via DIH AGRIFOOD DATA SPACE”	Across dataspaces data sharing over secure connectivity.	R & D	M
Section 2.2 UC2.2	“Relevant data sources to be obtained from both data spaces within the use case. “		R & D	M
Section 2.2 UC2.3	“Data about households (including flats) to be arranged including number of participants, building area and other relevant data. Dataset shared through the MOMS DATA SPACE”		R & D	M
Section 2.2 UC2.4	“MOMS is using the data from two sensors in the city”		R & D	M
	Use Case 3: Agriculture			
Section 2.2 UC3.1	“Retrieve data from DigiAgro DS where sensor data from crop owners is collected and stored”	Across dataspaces data sharing over secure connectivity.	R & D	M
Section 2.2 UC3.1	“Use AgroScience DS for data processing services and machine learning algorithms to analyse the collected data”		R & D	M
Section 2.2 UC3.2 and 3.3	“Integrate environmental data and weather forecast data seamlessly between the two dataspaces”		R & D	M

2.1.10 Workflows

The following sub-sections describe the sequence diagrams of the MCL Module.

2.1.10.1 Write data offer description

This feature provides the capability to write the description of data offer in JSON format to the Tier 1: Catalogue module. Figure 11 shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- Self provision the description of data offer in JSON format.
- Securely communicate the description to the Catalogue module which stores it into a local database.

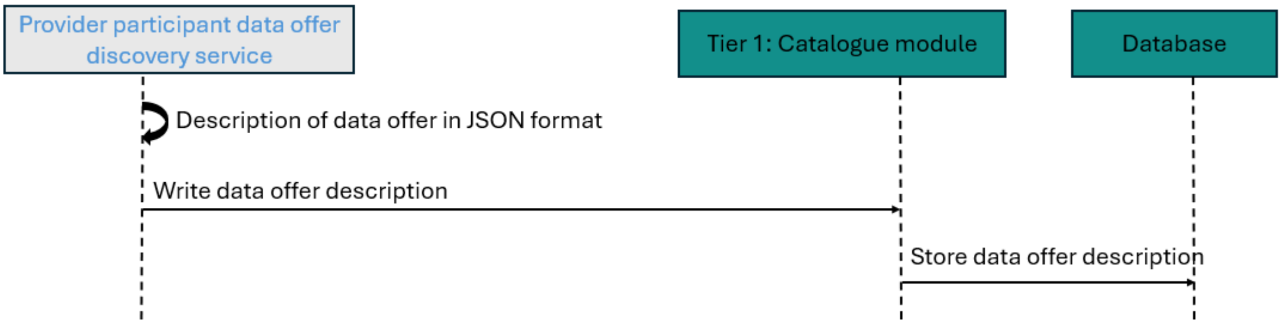

Provider participant data offer discovery service
Tier 1: Catalogue module
Database
Description of data offer in JSON format
Write data offer description
Store data offer description

Figure 11: Write data offer description sequence diagram

2.1.10.2 Read data offer description

This feature provides the capability to read or discover the provider participant's description of data offer in JSON format by a consumer participant. Figure 12 shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- Send read request for data offer.
- Receive and parse data offer received in JSON format.

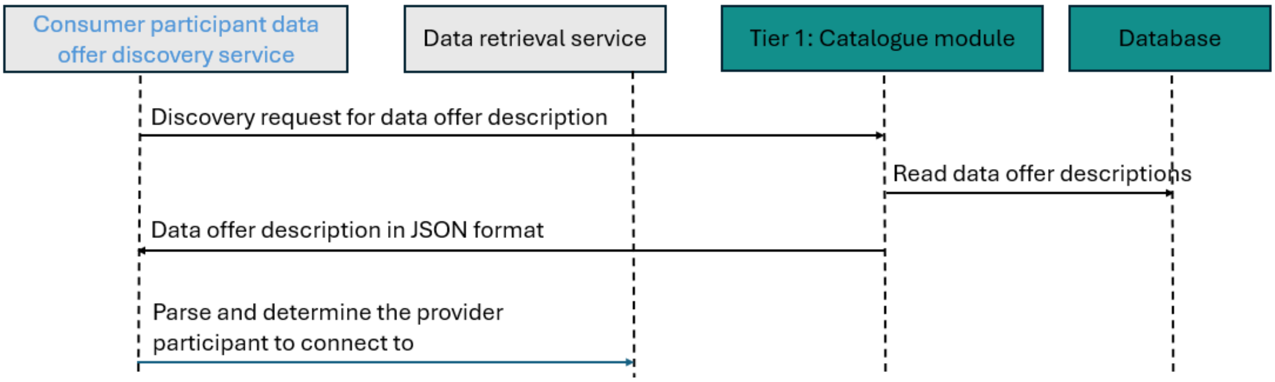

Consumer participant data offer discovery service
Data retrieval service
Tier 1: Catalogue module
Database
Discovery request for data offer description
Read data offer descriptions
Data offer description in JSON format
Parse and determine the provider participant to connect to

Figure 12: Discover data offer description sequence diagram

2.1.10.3 Data sharing using push/pull data transfer

This feature enables a use case application to obtain discrete data from data stores using push/pull data transfer of the dataspace connector data plane. It is assumed that the discovery steps shown in Figure 12 have already taken place. Figure 13 shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- Use case app A initiates the data retrieval request.
- Data retrieval services knows from discovery which provider participant can provide the requested data.
- Data retrieval service sends a read data request to the corresponding data store through the push/pull data transfer for obtaining the discrete data.
- Received discrete data is passed on to the use case app for its consumption.

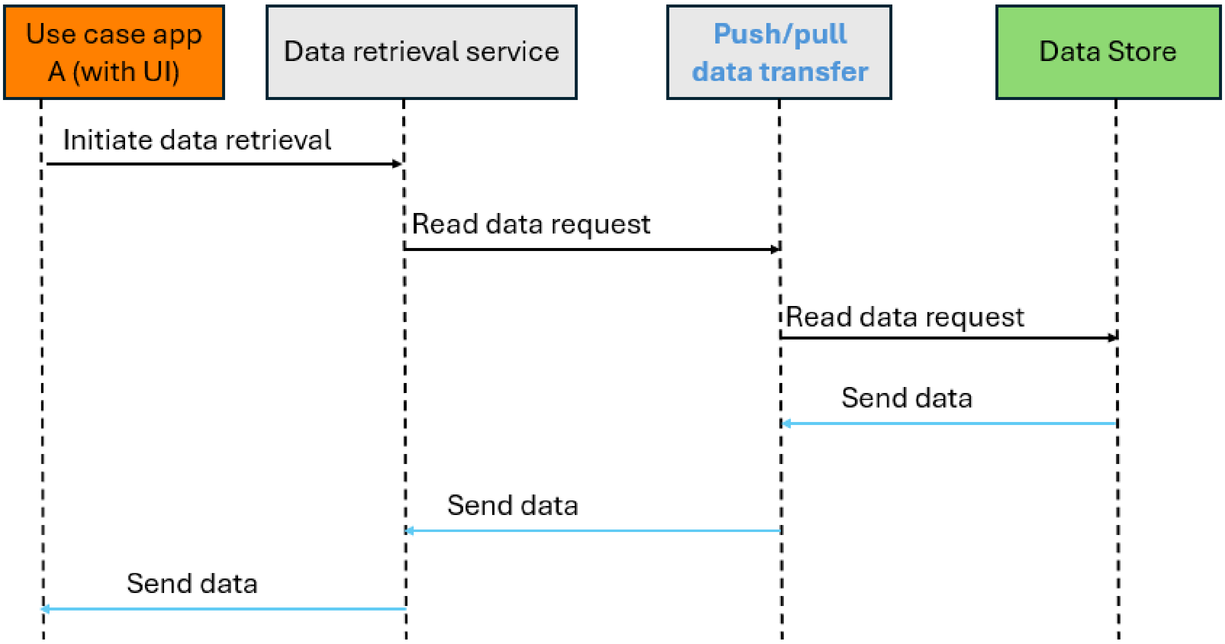


Figure 13: Data sharing using push/pull data transfer sequence diagram

2.1.10.4 Data sharing using data stream transfer

This feature enables a use case application to obtain streaming data from data stores using the data stream extension of the dataspace connector data plane. It is assumed that the discovery steps shown in Figure 12 have already taken place. Figure 14 shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- Use case app B initiates the data retrieval request for a specific topic for which the steaming is occurring.
- Data retrieval services knows from discovery which provider participant can provide the requested data.
- The data store is publishing the data for the specific topics to the data stream extension.
- Data retrieval service sends a read data request to the corresponding topic to the data stream extension for obtaining the streaming data.
- Received streaming data is optionally stored on a temporary data store before being processed at the use case app B.

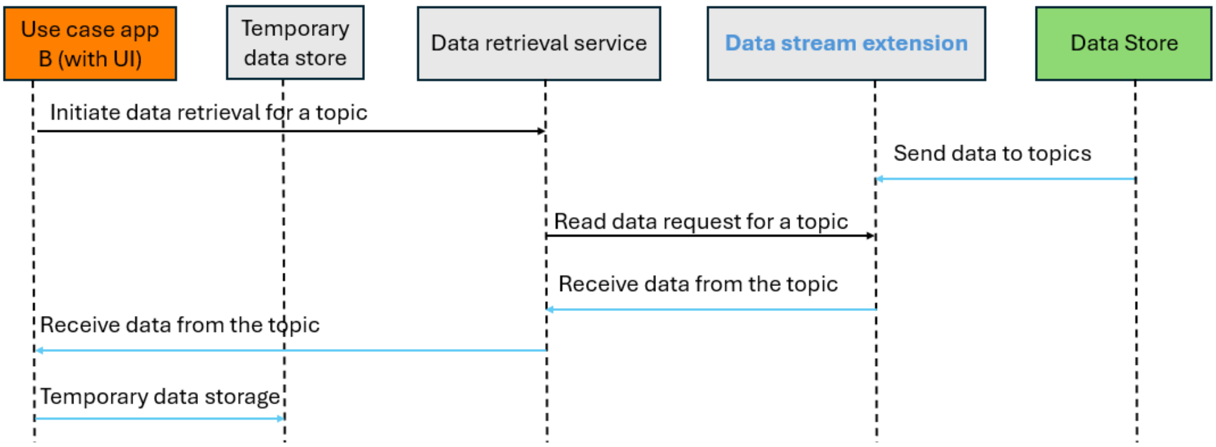


Figure 14: Data sharing using data stream transfer sequence diagram

2.1.10.5 Data caching

This feature enables intelligent data placement and data caching. Figure 15 shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- A vast data store sends data for caching to the intelligent data placement engine.
- The engine sends the data to cache to data placement controller, which in turn forwards the incoming data to the storage for caching.
- The controller also checks with the cache invalidation and consistency manager if the incoming data is up to date or is there any conditions which make the data inconsistent. If yes, then it updates the storage for data caching.

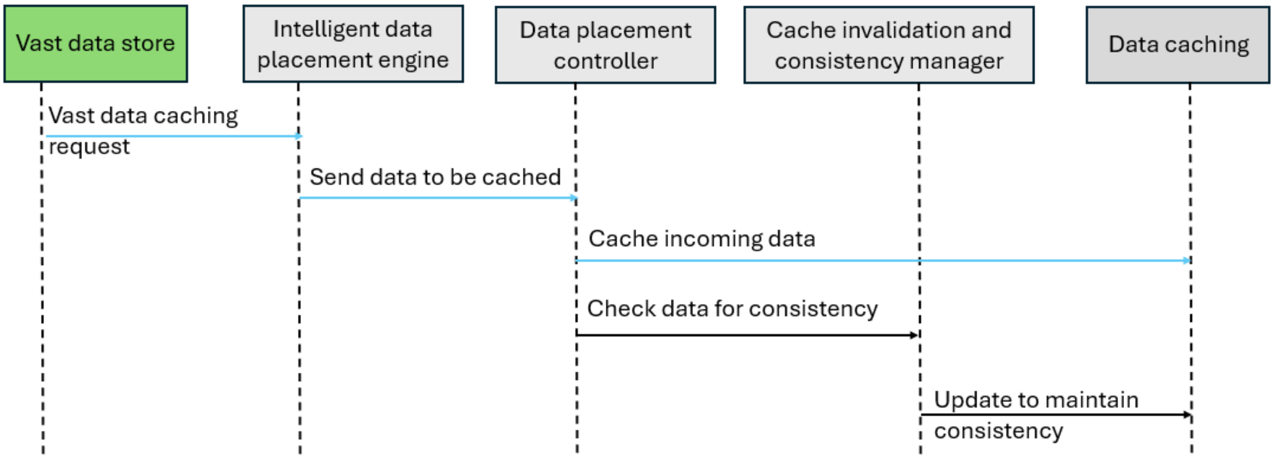

Vast data store
Intelligent data placement engine
Data placement controller
Cache invalidation and consistency manager
Data caching
Vast data caching request
Send data to be cached
Cache incoming data
Check data for consistency
Update to maintain consistency

Figure 15: Data caching sequence diagram

2.1.11 Role, Resourcing, and Milestones

Sub-component	Main Activity	M18	M24	M30	M36
Dataspace discovery service	Implementation and ensuring their integration with different dataspace participants	■			
Push/pull data transfer	Integrate it as a part of Eclipse dataspace connector	■			
Intelligent data placement engine	Develop the engine	■			
Data placement controller	Develop the controller		■		
Data caching	Implement the storage for data caching		■		
Cache invalidation and consistency manager	Implement it and complete integration of the intelligent data placement and caching			■	
Data partitioning and compression	Implement the sub-component		■		
Batch scheduling	Implement the sub-component		■		
Data transfer, error checking, retries	Implement the sub-component		■		
Stream message broker	Implement the sub-component	■			
Stream processing engine and data transfer	Implement the sub-component	■			
Data retrieval service	Implement the sub-component	■			
Integration of all sub-components and bug-fixing	Together with debugging, testing, and validation	■	■	■	■
Table Total/DOA Task Total/Resilience					

2.1.12 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
N/A			