Catena-X

THE FIRST OPEN AND COLLABORATIVE DATA ECOSYSTEM

Onboarding Guide: Data Integration Patterns & Tools

Release V1.2, August 2024

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages
Finanziert von der Europäischen Union
NextGenerationEU

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

1. INTRODUCTION

Catena-X is a decentral system, where data is exchanged peer-to-peer between two companies on demand. Those companies need to be able to provide and receive data in the right quality, semantic model, and data format. This requires companies to adapt and extend their existing data architecture.

This guide is meant to provide an overview of the different patterns for company internal data integration in the context of data exchange within Catena-X. It is intended for enterprise architects and executives in the context of data integration. Creating an understanding on the different options that companies may leverage to connect backend applications with a data space to provide and receive data, this guide further provides general and Catena-X use case specific advantages and challenges that need to be considered before choosing the pattern that fits best.

This guide is not an introduction to the general concept of data spaces, this has already been described in different whitepapers as referred below:

Whitepaper	Link
Open DEI	https://design-principles-for-data-spaces.org/download-gate/
IDSA	https://www.fraunhofer.de/content/dam/zv/en/fields-of-research/industrial-data-space/whitepaper-industrial-data-space-eng.pdf
Catena-X Operating Model	https://catenax-ev.github.io/docs/operating-model/why-introduction
Gaia-X Architecture Document	https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/

Table 1: Links to different whitepapers

Furthermore, this guide does not provide technology specific implementation details and technologies, as they would vary too much from company to company.

Starting with a high-level overview over the different components in the Catena-X data space, this guide walks through the different capabilities that a data provider/consumer needs to build up to technically participate in a data space.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

Following with the description of four main different data integration patterns, their advantages, challenges as well as their fit for the various Catena-X use cases.

After that the Catena-X facing data processing patterns are introduced. Those patterns are key to understand how data is shared via Catena-X and what is necessary to participate in use cases that make use of these patterns.

The subsequent chapter 'Patterns for Dataspace Connector Operations' evaluates the different options to operate the data exchange component of the data space.

With the section about Data Governance and Security important organizational constraints are elaborated for companies that introduce a Catena-X data integration solution.

Afterwards, the chapter Conformity Assessment and Certification gives insights into what needs to be done in terms of certification for a data integration solution.

Finally, some considerations about creating internal data products and external data offers are given.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

2. HIGH LEVEL DATASPACE ARCHITECTURE

While this whitepaper does not explain the details of how Catena-X works and which technologies are used, this chapter still briefly explains all concepts that are necessary to understand to select a suitable data integration pattern.

For a more technical overview over which components and concepts need to be implemented within each data provider/consumer, see the 'Deployment View' guide on the official Catena-X Code Repo[1]. Figure 1 illustrates the general concept of how the Catena-X data space works and exemplarily depicts the interaction of involved organizations with respect to data exchange.

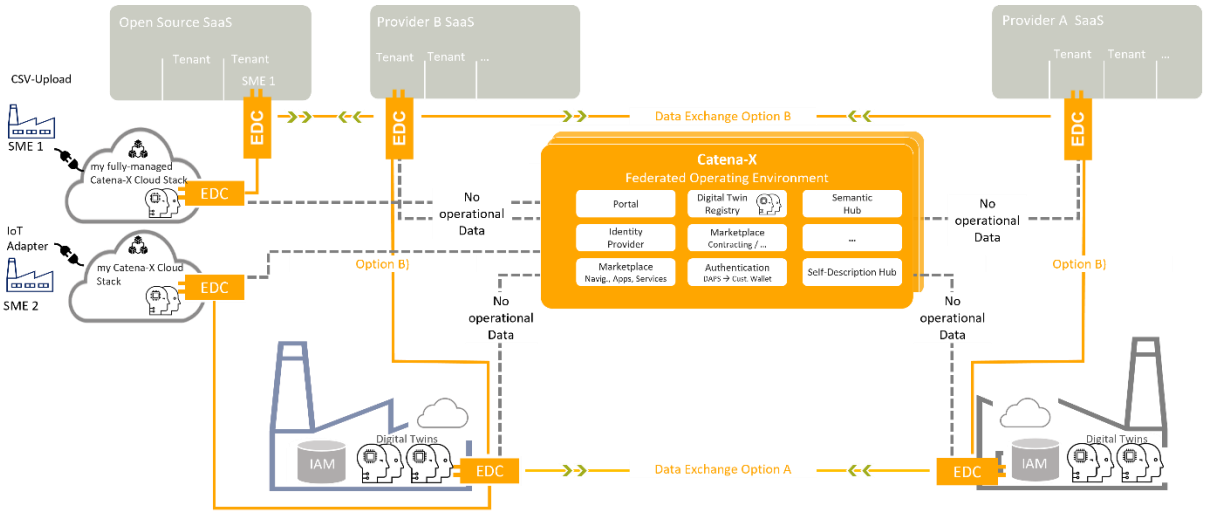

Figure 1: Catena-X Data Space – How Catena-X works / What holds us together? Example[2]

Participation in the Catena-X data space requires certain technical capabilities within organizations. Zooming in on the interaction of two involved organizations, Figure 2 puts the required internal and external capabilities in the context of the overall data space setup, which are described high-level in the subsequent paragraphs.

[1] https://eclipse-tractusx.github.io/

[2] The Digital Twin Registry for the Digital Twin pattern is depicted in this picture as centralized. This service is now decentralized and needs to be deployed once at every data consumer or provider.

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages
Finanziert von der Europäischen Union
NextGenerationEU

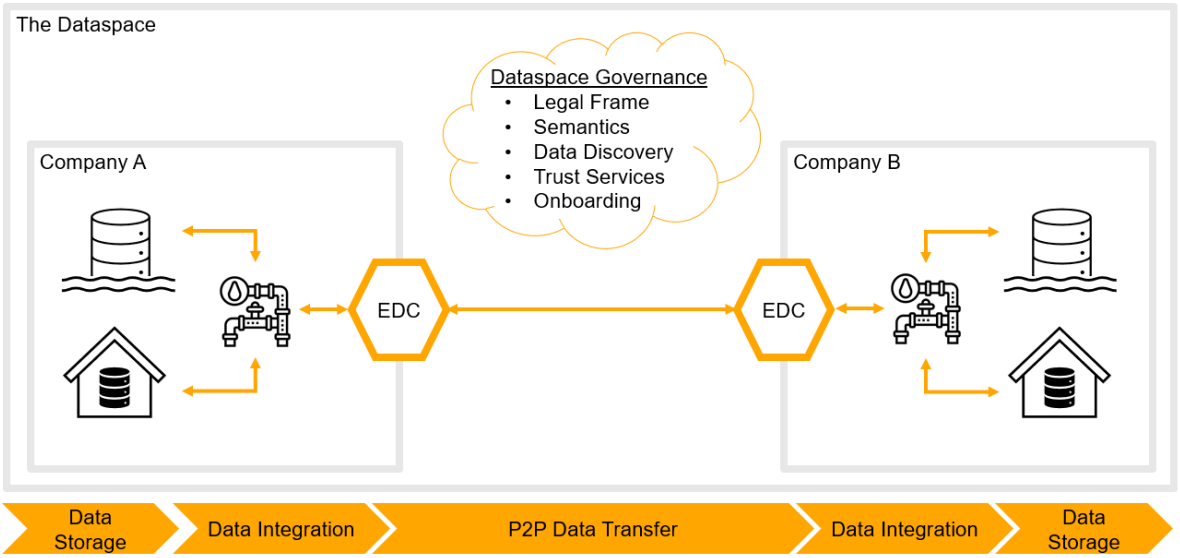

Figure 2: Catena-X Data Space Setup - Simplified

P2P Data Transfer

The Catena-X data space is decentral. This means, that there is no central storage system that collects and distributes data. Rather, data is exchanged between two companies on a peer-to-peer basis. This means, that each company needs a component that handles a) contract negotiations over data assets and b) the actual data transfer. Catena-X uses the 'Eclipse Dataspace Components/Connector' (EDC) for this purpose[3]. The EDC is the gatekeeper for incoming external data as well as outgoing internal data and thus plays an important role in the network.

Data Integration

Catena-X uses shared semantic models to enable interoperability between applications and partners. All data that is transferred via the Catena-X network needs to be transformed according to those semantic models before it is sent out. Similarly, received data from external partners first needs to be transformed into a format that backend systems can understand again. Furthermore, data might need to be combined from multiple source systems, depending on the Catena-X use case (see 'Data Storage' below).

Data Storage

The data that is exchanged within Catena-X most likely comes from existing source systems. These source systems differ from use case to use case. For some use cases, many different source systems will need to be identified. The challenge is to get the data in matching quality and granularity. Other use cases might only require a single source system, where data transformation

[3] https://projects.eclipse.org/projects/technology.dataspaceconnector

and integration are easier. Similarly, when receiving data, data might need to be distributed to various or just one target system. This heavily influences the choice of the data integration pattern.

Note: If a Catena-X "certified solution" from one of the app marketplaces is used, the components for data exchange and integration (e.g., EDC, transformation of semantic models) is part of the service offering and doesn't need to be implemented by a data provider or consumer. Additionally, at some point there will be certified solutions specifically for data integration and provisioning.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

3. TECHNICAL CAPABILITIES FOR DATA PROVISIONING

The previous chapter gave a high-level overview over companies' internal components as well as external components. This chapter focuses and deep-dives on the internal technical capabilities that a data provider/consumer needs to build up so that he can participate in the Catena-X network. Not all those capabilities are mandatory to have and depending on the data integration pattern, some are even obsolete. Figure 3 gives an overview over the mandatory and optional capabilities with mandatory capabilities being highlighted in orange.

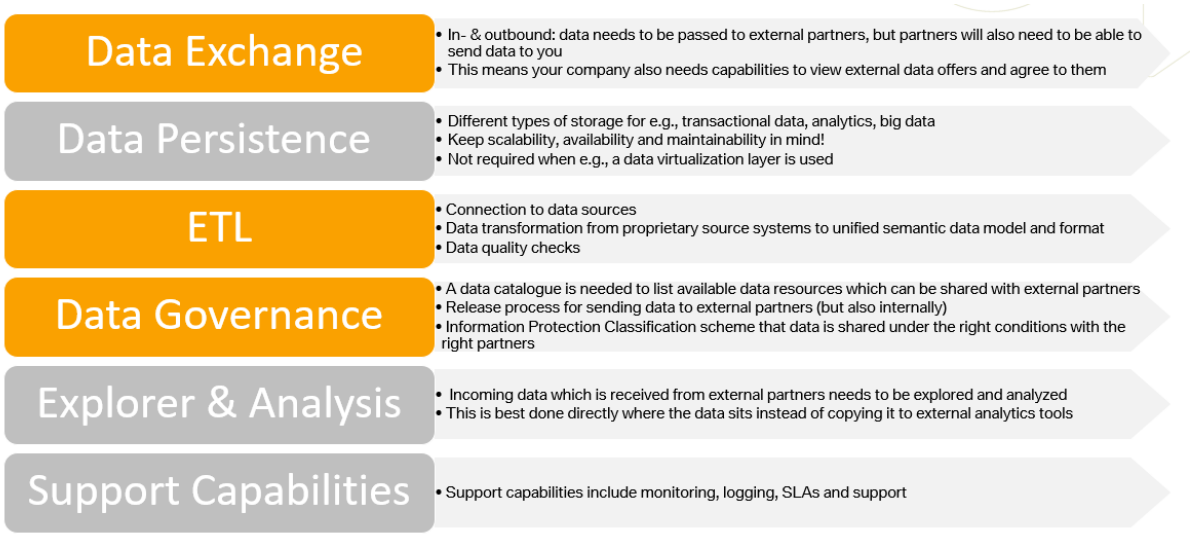

Figure 3: Data Integration Capability View – General (orange: mandatory; grey: optional)

Figure 4 puts these technical capabilities into context, demonstrating the interrelations.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

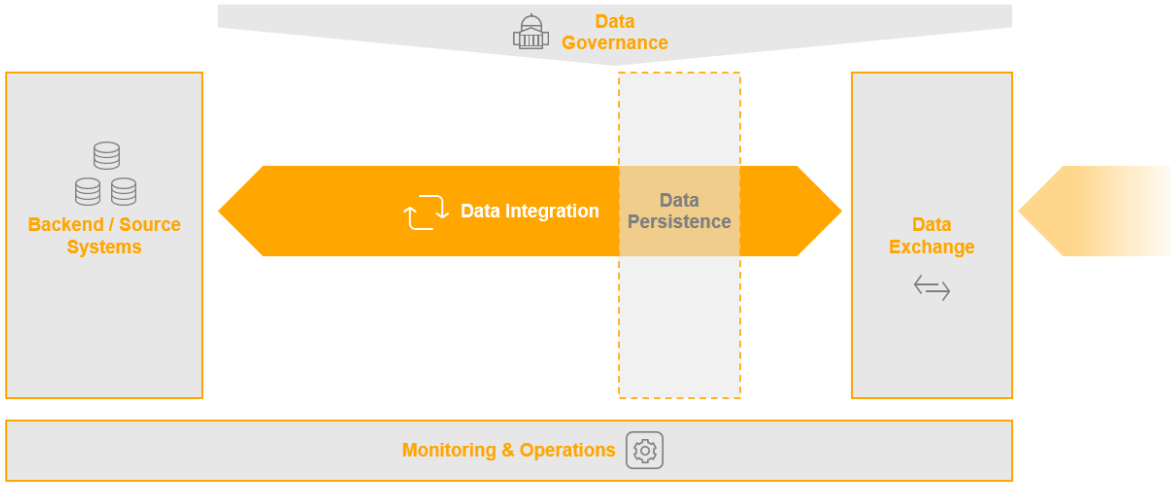

Figure 4: Data Integration Capabilities Architecture

3.1 Data Exchange

The data exchange capability encompasses all aspects directly involved in data exchange within the Catena-X network. This exchange is facilitated by Dataspace Connectors that act as bridges between different systems, ensuring secure and standardized data transfer. In Catena-X, the **Eclipse Data Space Connector (EDC)** is the reference implementation for this purpose.

The EDC is an open-source, modular connector designed to facilitate secure and standardized data exchange within the Catena-X network. It is architecturally separated into two main components: the control plane and the data plane. The control plane handles the negotiation and management of data contracts, ensuring that data access and usage comply with defined policies. The data plane is responsible for the actual transfer of data between systems, leveraging various protocols and extensions.

The EDC's modular architecture enables the development and integration of extensions to expand its capabilities and tailor it to specific use cases. These extensions can be used to implement the actual data exchange and can be categorized into two main types:

3.1.1 Provider-Agnostic Extensions:

- **S3:** exchange large datasets stored in object storage systems like S3 compatible services.[4]
- **HTTP REST:** leverage the REST API architecture for flexible data exchange.
- **FTP**: exchange files leveraging the FTP protocol.

3.1.2 Provider-Specific Extensions:

[4] Due to the long history of the Amazon S3 offering, several cloud providers have chosen to implement S3 API compatible storage offerings.

- **Cloud Provider Services:** Integrate directly with cloud-specific data services (e.g., Azure Data Lake Storage, Google Cloud Storage, Amazon S3) for seamless data exchange from or to a cloud environment.
- **Custom Protocols:** Utilize extensions for proprietary or specialized protocols used by specific partners or industries.[5]

3.2 Data Persistence

Note: Not every data integration pattern will need data persistence.[6] Concepts such as data fabrics / data virtualization work without such a caching layer.

The data persistence layer acts as an intermediate storage or cache before data is sent out or after data is received. It will need to handle different types of data, depending on the use case and data transfer mechanism, like filesystems, relational and other databases, streams, etc.

A data persistence layer implies that data is duplicated, and thus mechanisms to control data quality as well as lineage[7] are very important.

3.3 Data Integration

Data needs to be extracted from backend systems either on a regular basis or on demand. Depending on the Catena-X use case and company internal backend system landscape, data from different sources then needs to be combined to a new data asset to meet the Catena-X data needs. Furthermore, data needs to be transformed into the right Catena-X semantic model.

If no data persistence layer is used, this needs to be done "on the fly" when a request for data is received. If a data persistence layer is used, the data integration process to create new data assets for Catena-X can be done in stages:

1. extract data from source systems
2. store in the persistence layer
3. combine different data sets
4. adapt the semantic model

Note: If data is received from partners, it needs to be distributed to target systems. Before this can be done, the semantic model needs to be changed to match the target system, received data sets need to be split according to target systems and data needs to be ingested into the target system.

3.4 Data Governance

The data governance capability is a mixture between process and IT capabilities.

---

[5] https://eclipse-tractusx.github.io/

[6] Chapter 5 discusses in detail, which pattern does or does not entail data persistence.

[7] See sections **Error! Reference source not found.**, **Error! Reference source not found.**, 5.3.4, and 5.4.4 for Catena-X advantages on data quality, including lineage.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

First, a company needs to be able to – internally – create and manage data products / assets. This includes company internal release processes as well as tools to manage access to data. In a next step, processes (and tools) are needed to externally release data to partners. This includes the definition of access rights (which partners are allowed to access certain data) as well as usage rights (what is the partner allowed to do with the data). Access and usage rights will differ from data asset to data asset.

For data offered by external partners, processes and tools to view external contract offers and agree to them are needed. It needs to be made sure that only authorized employees can view said contract offers and agree to them. Furthermore, a data consumer needs to ensure that the usage rights set by the data provider are followed.

3.5 Explore and Analyze

Being able to explore and analyze incoming data is not a mandatory capability to have. However, it speeds up the process of integrating external data into a company's systems. Although the semantic models of the data that is to be received is standardized in Catena-X, it still might be, that the data differs from what is expected.

3.6 Monitoring and Operations

Receiving data from, as well as sending data to external sources in a network with horizontal as well as vertical competitors has several special implications.

First, a company should have a gap-less audit trail on which employee released which data to which partner (or received data). Furthermore, it should be logged, who (which external partner) accessed which data asset and when, to be able to prove compliance with e.g., antitrust laws.

Furthermore, existing incident response processes need to be extended by an external component: A partner needs to be able to inform a company that a data pipeline is broken, data quality issues occurred or that the access points / EDCs are not working. Ideally, this is not observed by the partner, but by internal monitoring tools that then inform partners about a possible disruption.

Regarding operations: Catena-X heavily relies on open-source software (OSS), also for the components that need to be deployed at each data provider/consumer company. Most companies usually buy managed software that might be based on OSS, but they don't manage, patch and update OSS software themselves. There will be a portfolio of managed solutions for components such as the EDC or Digital Twin Registry at the start of Catena-X. Nevertheless, especially large enterprises should be prepared to operate OSS software in a productive environment themselves, if the existing solutions don't fit their needs. This means that they need to be able to handle patching, updates and potentially contribute to an OSS project when they discover a bug or security issue that needs fixing. Finally, if a company needs adaptations to OSS code, they cannot simply request a new feature, and somebody will implement it (as it would be when customizing COTS software). Either they pay somebody to adapt the software for them or they do it themselves.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

4. SELECTION AND PREPARATION OF BACKEND SYSTEMS

Catena-X is designed around its ten initial use cases. Software vendors will provide business applications to calculate CO2 footprints along the supply-chain, automate recalls along multiple tier levels or help a recycler to understand how to recycle cells of a battery of a specific vehicle. The data that needs to be provided by the partners to create the data chains that power the different use cases will mostly come from legacy backend applications such as ERP, MES, or Shopfloor (SPS) systems. Those systems are sometimes older than 30 years. Furthermore, data for Catena-X use cases will most of the time come from various source systems, which need to be combined to provide the right data. Thus, to be a member of the Catena-X network, an ETL process needs to be established.

Technical data provisioning for Catena-X is a six-step process:

1. Select the use case for which data needs to be provided.
2. Understand the data needs based on the semantic model(s) of that use case.
3. Determine the backend systems from which data needs to be extracted.
4. Choose the data integration pattern suitable for the specific case.
5. Extract and prepare the data from backend systems according to the pattern.
6. Publish and register data in the Catena-X network.

Especially the effort and time to extract and prepare data from backend systems is not to be underestimated. Data from potentially multiple very old systems needs to be extracted and thus transformed from different proprietary data formats and semantics to a unified data asset. Data quality often varies, performance of the backend systems might be an issue if the amount of data needed in Catena-X is large, experts in old systems are hard to find or have retired, the requirements on response times or timeliness of data from source systems might present a challenge.

Experience has shown that the extraction and preparation of data from backend systems (Step 5) takes up to 80% of the time that is needed to – technically - provide data to a Catena-X use case. The other five steps take the remaining 20%. If no structures for technical data integration (such as a data lake or data mesh) are existing, it might take some time and effort to setup the first use case.

Note: This chapter only described the steps for a technical integration of data. Companies joining Catena-X will also need to adapt or create governance processes for the release of internal data or the retrieval of external data. Please refer to the governance process guide for more information.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

5. PATTERNS FOR DATA INTEGRATION

The following section describes different patterns for data integration in a wider, more practical sense. In the context of this guide, data integration comprises of the target/destination systems, data transformation and data persistence.

Note: Data integration always includes both ways:

- Data provisioning: extracting data from source systems, transforming it to Catena-X compatible data and then offering it to external partners.
- Data consumption: receiving data from external partners, transforming it into target system compatible data and then ingesting into the target system.

The options/patterns to operate a Dataspace Connector are excluded and described in a separate section in this guide called Patterns for Dataspace Connector Operations. Although the different data integration patterns can be combined with any patterns to operate a dataspace connector, some combinations make more sense and will be advised.

Each pattern is introduced by its general design characteristics, an architectural overview based on possible technical building blocks is given, the advantages and challenges mainly in the context of Catena-X are specified, and then the suitability for different use cases and participation scenarios is explored. The patterns described in this guide are simplified industry best practices which are extended and modified for Catena-X. The following sections are not meant to give a perfect introduction to e.g., data lakes or data mesh concepts. There are excellent guides for those by professionals, please read them for a more in-depth explanation.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

5.1 Pattern A: Central Data Asset Storage as an intermediate layer

5.1.1 Summary

With central data asset storage, data from connected backend systems is stored in a central repository, such as in a data lake or central data warehouse. Here, different kinds of raw data can be stored, processed, and combined into new data sets, while still being centrally available for any type of access. Although this comes at some costs and must be established through a proper implementation first, it provides good performance even in big data scenarios while ensuring that data quality is met, and data governance is fulfilled. It is interesting to note that data lake offers clear organizational isolation, meaning that each organization has its own data governance, etc. As such, the central storage solution is suitable for large Catena-X use cases, where big data applications are common or even analytical applications are applied, such as traceability, quality, or circular economy.

5.1.2 Design Characteristics

A central data asset storage – as the name indicates – combines and collects data from different backend systems and external sources into a central data store, such as a data lake or central data warehouse. In this central store, raw data can be combined and further processed to new data assets, distributed to target systems, or shared externally again. The storage type is not limited to unstructured data (e.g., XML, JSON, or Parquet) but a central storage hub can also be built for transactional data, no-SQL data, or streaming data.

Thus, the central data asset storage acts as an intermediate layer between a) two internal systems and b) internal systems and external partners. Data passes the central storage before it is ingested into another system or released to external partners. Once ingested into the central storage, data can be (semantically) transformed, data quality can be improved, or – as mentioned – different raw datasets can be combined into new data assets with higher value. As a result, data is potentially replicated multiple times.

Furthermore, there is central governance on top of the data. It can be centrally (in one place/tool, but obviously by different teams) managed who gets access to which data, which information protection classes apply to data, and to whom – externally – data is released.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

5.1.3 Architecture Overview

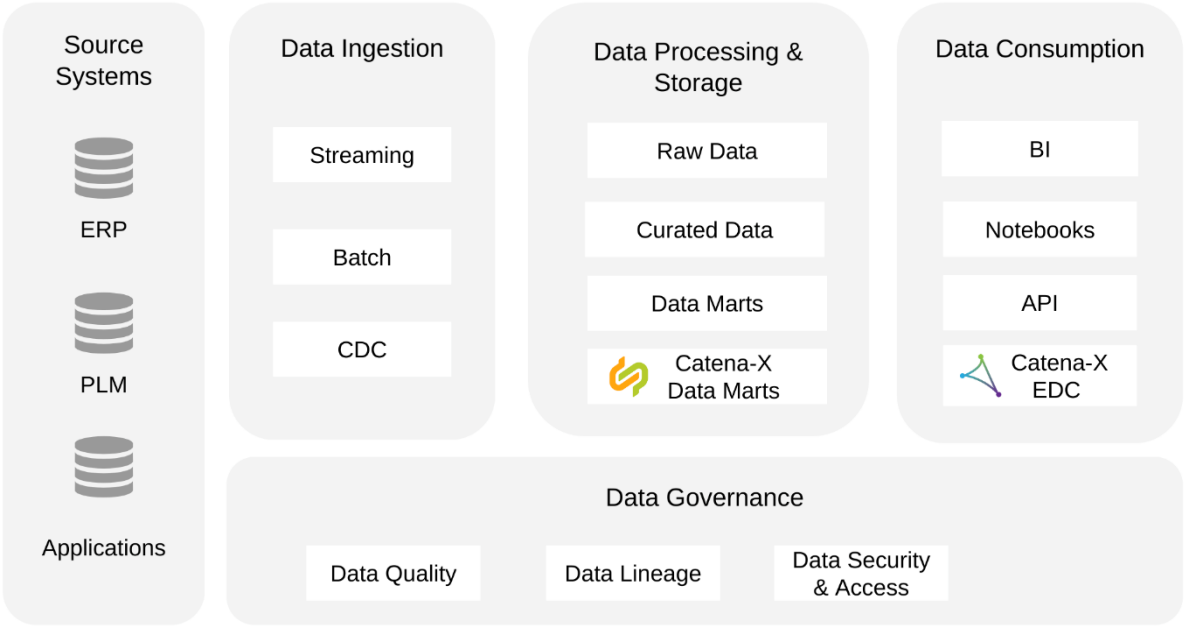

Figure 5: Architecture Overview of Data Lake

5.1.4 Advantages & challenges in the context of Catena-X

General advantages

- After the data pipelines that ingest data from a single source system into the central data storage are stable and data quality is good, it is very easy to distribute data to different target systems. Central storage makes it possible to have a "single source of truth", resulting in ease of maintenance and management of system and data. For example, if the API of the source system changes only one pipeline needs to be maintained and managed.
- Because all data is cached in the central data store, it can be transformed into a unified semantic model (or in the case of Catena-X into the Catena-X semantic model). This has two advantages: Data consumers will only need to understand one shared semantic and there is no need to do computationally intensive ad-hoc transformations into a target semantic.
- As there is one central place where data passes, tools for ETL, security, or monitoring can be unified and reused.
- Data lineage – tracking where data comes from and where it is coming to – is a doable task with a central data store. As data comes from a source system, is combined with other data, and then distributed to target systems all in one place, it can be kept track of where data flows, even externally.

Catena-X specific advantages

- Having data in the Catena-X semantic model "at rest" has huge benefits when it comes to providing larger amounts of data in a reasonable time, as no data transformation needs to happen ad-hoc.
- Data for Catena-X use cases will most likely come from different source systems. Being able to combine data into a specific, internal Catena-X data product before data is shared, is a big bonus.
- Also, having the capability to analyze and check quality of incoming data before it is distributed to target systems can be beneficial. Although there will be SLAs and predefined semantic models, it is not guaranteed that external data providers fully adhere to those rules.
- Extending an existing central data storage with Catena-X specific components (e.g., EDC) and external release processes is simple, as they can be centrally steered.
- By centralizing data exchanges, this pattern enables to reduce the amount of Dataspace Connectors to operate. While the central data store might present a honeypot for malicious actors from the outside, it also shields the different target systems with another layer of security and avoids that externally received data is directly ingested into target systems. This allows for additional integrity and security checks of external data. It also decouples the source systems from external requests which is good for both security and performance, as the load is not placed upon the source systems.

General challenges

- Building a central data store including the integration of the most valuable backend systems and creation of data governance structures comes with a high initial effort and invest, and should only be considered if a general strategic direction towards a data-driven company – not just for Catena-X – is to be reached.
- Each data asset/product in the central data store requires constant and proper governance, quality control and maintenance.
- Data is replicated across multiple systems. This means that a) storage costs increase over time and b) data needs to be kept in sync between source system and the central store which can be complex.

Catena-X specific challenges

- Catena-X use cases use a variety of different storage technologies. It won't be enough to set up e.g., a data lake for large unstructured data. It will also be necessary to handle streaming data or documents. In case the central storage does not support those storage and processing technologies, it creates additional effort to setup those technologies for potentially just a single use case.

5.1.5 Suitability for Different Catena-X Use Cases

- A central data asset storage is a viable solution when a large-scale participation in multiple use cases is intended. Central governance processes make the release of data to external partners easy. A central storage allows for the distribution of data to different targets in the enterprise.
- This pattern works for all Catena-X use cases that:
  - Require larger amounts of data.
  - Require data "on-demand" per pull from a partner.
  - Require data from multiple source systems.
  - Require complex queries on the data that otherwise would put a high load on source systems if connected directly.
- This pattern – if not existing in a company yet - is not advised:
  - If participation in a single use case is intended as the overhead to setup a central data store for this is just too big.
  - Real-Time/Streaming data is needed. Then the source system should be connected directly or via a streaming service.
  - To be provided data comes from a single source system. It might be easier to setup a reporting table with Catena-X semantics in that source system and create a direct connection.
  - Incoming data is intended for a single or few target systems. Storing data in a cache makes it harder to keep track of the data and follow providers' usage policies. It is easier to e.g., "delete data after 30 days" if data is only stored in one persistence layer rather than multiple layers.

Note: The following list is based on a subjective assessment of the various patterns fulfilling use case requirements regarding data integration.

Potentially suitable if aiming to participate in the following Catena-X use cases:

- Traceability
- Circular Economy
- Quality

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

5.2 Pattern B: Data Virtualization Layer / Data Fabric with direct access to backend systems, exposed through APIs

5.2.1 Summary

The data virtualization layer uses virtualization techniques to make data centrally accessible without moving it physically to any central repository. As such it combines the benefits of a central data access possibility of more advanced patterns like the data lake and data mesh at less implementational costs, as a simpler infrastructure is sufficient for its realization. Hence, the virtualization layer is comparably fast to set up while providing better overall performance than direct access solutions if data transformations are crucial for the respective application scenario. As such, the virtualization pattern is suitable for realizing simpler use cases which do not incorporate big data applications or analytics, such as the CO2 case.

5.2.2 Design characteristics

A data virtualization layer or data fabric, respectively, offers users a unified and technically abstract view for querying and manipulating data across a range of disparate sources. As such, it can be used to create virtualized and integrated views of data in memory rather than executing data movement and physically storing integrated views. Further, it provides a layer of abstraction above the physical implementation of data, such as location, storage structures, technology, access language, and APIs by connecting to different data sources and making them accessible from a single logical location. This makes data virtualization a technology-centric concept where data is made available via objective-based APIs, while it also allows to leverage metadata to drive recommendations.

5.2.3 Architecture overview

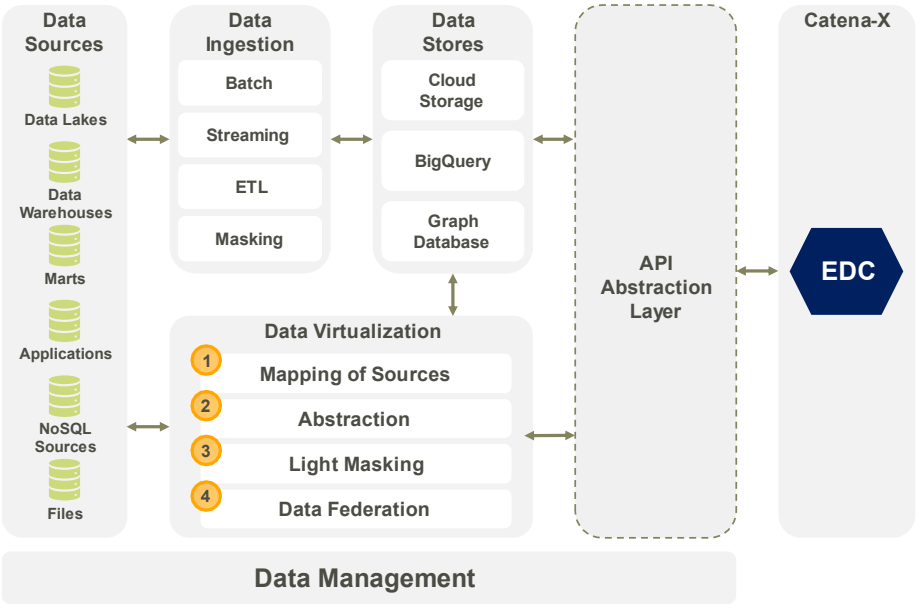

Figure 6: Architecture Overview of a Data Fabric

5.2.4 Advantages & challenges in the context of Catena-X

General advantages

- A virtualization layer removes the need for physical movement and storage of data as data must not be transferred somewhere. Instead, it is just virtually replicated and, thus, made accessible. Hence, due to non-required additional storage resources, cost advantages might be realized in comparison with more advanced solutions, such as data lakes or data meshes where data is replicated. Nevertheless, additional costs for on-the-fly calculations for data transformation and combination occur, which might eat up the cost benefits of reduced storage capacities.
- A central virtualization layer makes it possible to easily deploy enhanced data security and governance capabilities such as role-based access controls, encryption, masking, and obfuscation, as this can be handled centrally at the virtualization layer instead of individually for every accessed/connected backend system. The benefits of enhanced security can also be realized by a data lake concept. Moreover, as data passes through the virtualization layer the original data is secure, as these systems are decoupled from direct access.
- In general, the data access in one virtual data management layer is simpler to realize as the one to multiple physical systems such as in pattern D, as just one central connection is sufficient, instead of sole connections to multiple systems. Moreover, setting up a virtualization layer is relatively easy as less effort is required, since data does not have to be copied to a central destination (see pattern A) while also the more advanced data governance mechanisms of a data mesh are not required.
- Virtualization allows for a coherent data access from different data sources with no restrictions, as the virtualization layer serves as central access point for the connected original data sources.
- Upscaling is easy with virtualization, as simply adding additional backends to the virtualization layer poses not very much effort compared to the more advanced architectures of data lakes and data meshes, where additional data storage or enhanced governance solutions must be considered before corresponding systems are being connected.

Catena-X specific advantages

- Having a virtualization layer allows the user to quickly start a Catena-X use case no matter where the needed data is stored, since data sources can be easily and quickly virtualized and, thus, made accessible centrally.
- Through the addition of APIs, an additional intermediate layer is available between the dataspace connector and the backend system, which adds protection to the involved backend systems. Furthermore, reuse of data can be ensured for data exchange with other data spaces.

General challenges

- As data fabrics use virtualization techniques to directly connect to source systems, no historization of data is possible, unless virtualization incorporates a persistent data storage. However, extending the data virtualization layer paradigm into one with a persistent data storage solution such as a data lake (see pattern A) solves the data historization problems.
- Generally, latency poses a shortcoming of data virtualization as the virtualization layer does not contain a persistent data storage solution, meaning data must be requested from source systems. This challenge holds especially true compared to data lake solutions, which perform much better in such scenarios.
- As data is virtually made accessible, its original structure stays the same. Hence, data integration tools are required, e.g., for data curation or data quality enhancement, to make proper use of data and integrate the accessed data into useful data sets.
- Virtualization is not well suited for big data scenarios as respective latency issues due to increased backend system access are expected to be higher than with lower amounts of data. Moreover, analytics applications where combined data sets need to be created pose another shortcoming, again, due to the required backend access.

Catena-X specific challenges

- Data storage – before it is ingested into target systems - of incoming data from Catena-X is not foreseen within the pattern. However, this option is essential for preliminary data analysis or data quality checks.
- To improve response time, additional intermediate data storages need to be setup for in- and outbound data flow with Catena-X.
- The virtualization layer does not necessarily have the capability to transform data into Catena-X semantic models or from Catena-X semantic models into backend system semantics, requiring this to be handled on the backend level.

5.2.5 Suitability for different Catena-X use cases

- This pattern works for all Catena-X use cases that:
  - Require participation in only few use cases.
  - Require fast setup for data exchange with Catena-X.
  - Have limited resources / investment for setup of data exchange with Catena -X.
- This pattern – if not existing in a company yet - is not advised:
  - For participation in multiple use cases.
  - Where data for use cases comes from a multiple source-systems.
  - When incoming data is intended for multiple target-systems.

Note: The following list is based on a subjective assessment of the various patterns fulfilling use case requirements regarding data integration.

Potentially suitable if aiming to participate in the following Catena-X use case:

- CO2-Footprint

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

5.3 Pattern C: Data Mesh and a decentral approach

5.3.1 Summary

Data mesh patterns are a domain-oriented, decentralized approach to data sharing and integration. They are focused on a decentralized data ownership and architecture, and they treat the data itself as a product. This type of architecture has numerous advantages, among which the ease of up(scalability), clear ownership of the data and – consequently – a federated computational governance. In the case of Catena-X, since use cases fetch data from different domains, this can lead to ownership issues in that the relevant domain identification can be tricky and one might not instantly be able to figure out who is responsible to provide data access and other rights.

5.3.2 Design characteristics

The data mesh pattern is a people- and process-centric concept which enables a decentral approach to data integration. It is based on consumer-driven, late binding of loosely coupled, domain-oriented data sources, used to connect distributed data across different domains.

In essence, a data mesh is a network to exchange data about a business, where each domain that publishes data becomes a node in the data mesh. The analytical data provided by these domains is treated as a data product and the consumers (business analysts, data scientists, etc.) are treated like customers, defining the semantics of the data, which is made available via controlled data sets.

Data mesh follows a distributed system architecture. This requires a federated and global governance which is based on standardization (ecosystem thinking).

The responsibility is distributed to the people who are closest to the data (business domains), to support continuous change and scalability.

A peculiarity of the data mesh pattern is that it comes along with self-service models, meaning that it allows a rapid and reliable provisioning/maintenance of the infrastructure without having to rely on IT operations teams. For example, this pattern provides tooling that supports a domain data product developer's workflow and includes capabilities to lower the current cost and specialization needed to build data products (as there is no middle layer).

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages
Finanziert von der Europäischen Union
NextGenerationEU

5.3.3 Architecture overview

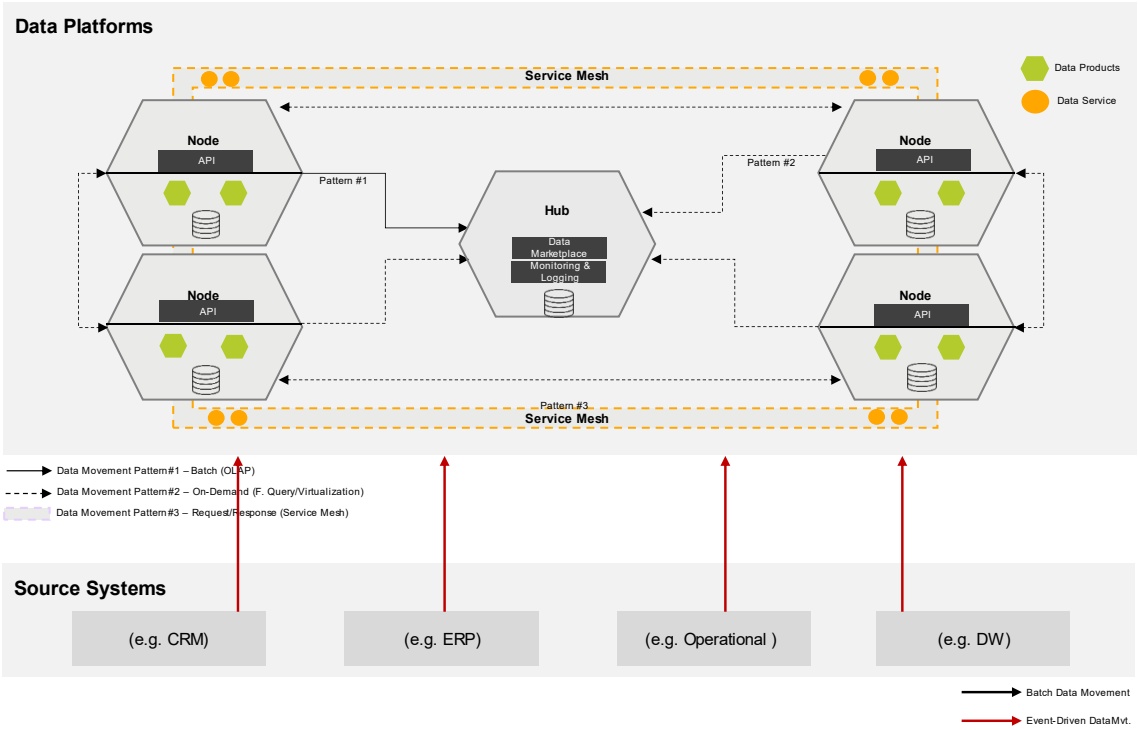

Figure 7: Architecture Overview of Data Mesh

5.3.4 Advantages & challenges in the context of Catena-X

General advantages

- Since the data producers publish their own data, there is no middleman in any step of the process, allowing domains to operate with a high degree of autonomy and ensure fast data exploration/preparation in the required format.
- Data integration and reconciliation can happen at later stages, and only if necessary. This translates into smaller upfront costs, incremental roll-out, and small to low risk. As a result, the costs to implement data meshes are lower compared to data lakes/data warehouses.
- All nodes in a data mesh can co-exist with legacy systems, both departmental and company-wide (such as existing data warehouses or architectures such as mainframes).
- Compared to the data virtualization and data lake patterns, the data mesh pattern encourages a decentralized and distributed approach for teams to manage data as they see fit, even though a global federated governance is in place.
- Data mesh architectures deal well with upscaling.

- The data owners in the organization are clearly defined at any point in time/any step of the process.

Catena-X specific advantages

- The clear physical isolation in data mesh allows hardware independence among the organizations on the Catena-X platform.
- In case of frequently changing requirements of use cases (e.g., due to market changes), a data mesh might provide the required adaptations in a fast manner, due to direct data ownership avoiding communication and process overhead.
- A data mesh supports the quick release of use cases by its flexible infrastructure characteristics & quick release process through direct data ownership.

General challenges

- The implementation and maintenance of a data mesh architecture could translate into higher costs of ownership and support. Furthermore, some nodes may lack skills and incentives.
- Unless published data is carefully curated and maintained, the data mesh will deteriorate into a set of disparate data lakes (also known as data puddles).
- As oftentimes domain-specific formats may be difficult for others to use, data cleansing and preparation may be required before the data exchange happens, making sure that no "join" information goes missing.
- Data mesh architectures require proper company-wide governance standardization, together with a supportive environment to ensure appropriate incentivization, SLAs and guidance.
- Security and access control must be implemented by every node independently.

Catena-X specific challenges

- Due to distributed data governance, there is not necessarily a central authority within a company involved in all data transfers, thus, information asymmetries may arise in the enterprise. This might even increase due to participation in the Catena-X data space.
- Data quality is not checked centrally. Thus, quality must be implemented at each node individually. This poses data quality issues as well as creating corresponding overhead for continuous data quality management as it is handled at the node-level. This might impose additional risk if dealing with external data exchange where data is not only required to be provided in a certain quality, but also received through Catena-X into the organization. This data needs to be checked for quality as well.

5.3.5 Suitability for different Catena-X use cases

- This pattern works for all Catena-X use cases that:
  - Require fast and optimized response times.

- Involve many players.
- Intend large-scale participation in multiple use cases, which require provisioning data from many (disparate) data sources and/or data types/formats.
- Are versatile, due to participation in use cases with frequent requirement changes (because of market changes), proof-of concepts and/or ad-hoc analysis.
- Require the integration of (new) third-party data stores, applications, or services. Suitable resources, skills, and investment, as well as commitment through company wide data / digital transformation strategy is ensured.

- This pattern – if not existing in a company yet - is not advised:
  - If participation in a single use case is intended as the overhead to setup a data mesh for this is just too big.
  - If the data to be provided comes from a single or a limited number of source systems. It might be easier to setup a reporting table with Catena-X semantics in that source system and create a direct connection.
  - Incoming data is intended for a single or few target systems. Storing data in caches makes it harder to keep track of the data and follow providers' usage policies. It is easier to e.g., "delete data after 30 days" if data is stored in a persistence layer.
  - The resources and the investment for Catena-X 's data exchange setup are limited.

Note: The following list is based on a subjective assessment of the various patterns fulfilling use case requirements regarding data integration.

Potentially suitable pattern if aiming to participate in the following Catena-X use cases:

- Traceability
- Circular Economy
- Quality

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages
Finanziert von der Europäischen Union
NextGenerationEU

5.4 Pattern D: Direct connection between single backend systems and Catena-X

5.4.1 Summary

As the name suggests, when setting up this connection, the backend system of an organization is directly connected to the Catena-X platform. This connection does not demand any intermediaries and hence can be set up very quickly. Although this option has its own challenges like lack of a central data governance option, it comes in handy for instances where a connection must be set up in no time. As such, the direct connection solution is suitable for use cases, where a single backend system needs to be connected with Catena-X and put in use to exchange information.

5.4.2 Design characteristics

Backend systems are directly connected to Catena-X without any intermediary structure while demonstrating this pattern. To combine raw data into new data sets, different backend systems must be accessed and different data sources with different semantics must be put together. Since there is no central governance for data, data is governed by/ on the different backend systems.

Furthermore, as there is no intermediate layer, different tools such as ETL, security and monitoring techniques must be adapted and configured for each backend system and cannot be reused.

5.4.3 Architecture overview

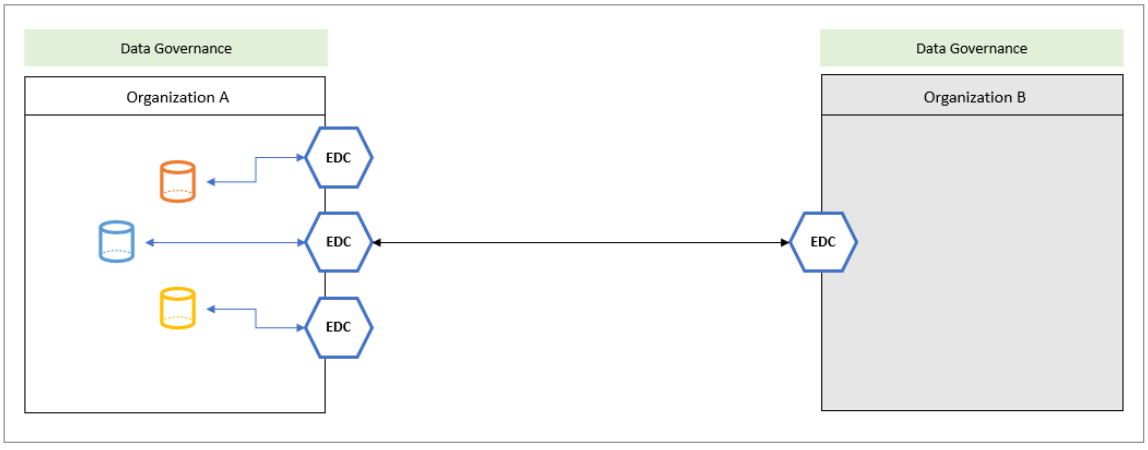

Figure 8: Architecture Overview of Direct Connection

5.4.4 Advantages & challenges in the context of Catena-X

General advantages

- As the pattern name suggests, it requires no intermediaries.
- Since there is no central (data) governance, it is quite simplified.

Catena-X specific advantages

- Direct connection between a backend system and Catena-X is a low initial investment and does not demand high-capacity. It also offers a fast setup timeline.
- Connecting a backend system directly to Catena-X provides the lowest latency, given there is just one source system involved.

General challenges

- Data transformation may not be possible as this needs to happen in source or target systems, which are usually not designed for transformation. Even if they can transform data, they are usually not designed to do it for large amounts of data.
- There can be potential latency issues if multiple source systems are accessed (requests must be forwarded to each backend system and waiting for the responses is required).
- Establishing proper security mechanisms induces a lot of effort as each backend system must be handled and considered separately.

Catena-X specific challenges

- Security – as backend systems are connected directly, security might be a major risk participating in the Catena-X data space, which needs to be carefully handled through respective security and authentication mechanisms.

5.4.5 Suitability for different Catena-X use cases

- This pattern works for all Catena-X use cases where:
  - Just a 1-on-1 or 1-to-few (not a 1-to-many) mapping between source system & use cases is needed.
  - There are limited resources / investment possibilities for setup of data exchange with Catena-X.
- This pattern – if not existing in a company yet - is not advised:
  - Where many-to-many mappings between use cases and source systems are needed.
  - If data transformation is essential, as transformations need to happen within the target or source systems.
  - In cases where historization of data is needed.

Potentially suitable pattern if aiming to participate in the following Catena-X use cases:

Note: This area will be detailed in a future version of this guide.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

5.5 Data Integration Patterns: Strengths and Weaknesses

The following table summarizes the strengths and weaknesses of each of the options. For a detailed analysis, refer to the chapter that follows.

	A. Central Data Asset Storage as an intermediary layer	B. Data virtualization Layer/Data fabric with direct access to backend systems	C. Data Mesh and a decentral approach	D. Direct connection between single backend systems
Performance	+	+	+	+
Operations/ Maintenance	-	+	-	+
Security	+	+	+/ -[1]	-
Data Transformation	+	-	+/-[2]	-
Capacity and Effort	+	+	+	+
Governance	+	+	-	-
Other Strengths	Unification of processes	Cost advantage	Scalability	Setup timeline

Table 2: Overview of Data Integration Patterns Strengths and Weaknesses

[1]Security is easy to manage as long as data comes from a single or limited number of source systems as security and access control must be implemented by every node independently.

[2] In case of frequently changing requirements of use cases (e.g., due to market changes), a data mesh might provide the required adaptations and data transformations in a fast manner, due to direct data ownership avoiding communication and process overhead. On the other hand, data transformation within a data mesh architecture could translate into higher costs of ownership and support. Furthermore, some nodes may lack skills and incentives.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

6. CATENA-X DATA PROCESSING PATTERNS

The last chapters focused on the patterns for organizing company-internal IT architecture to prepare for Catena-X data integration. Now the focus shifts to the Catena-X facing standards that need to be fulfilled, to participate in Catena-X use cases. Here, it is important to understand that standards and protocols differ between the use cases. Thus, the following chapter aims to provide an overview about the different patterns that are used to organize data transfers accordingly. Those alternative combinations of standards for data processing are defined as "Data Processing Patterns" in the scope of this document (see Figure 9).

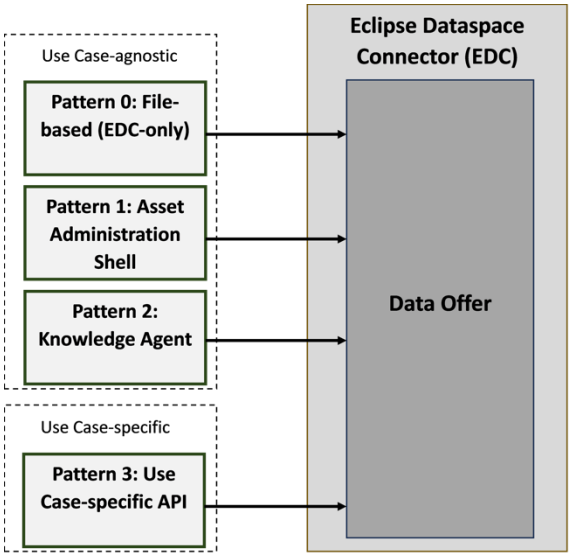

Figure 9: Catena-X Data Processing Patterns - Overview[8]

6.1 File-based (Dataspace Connector-Only)

This most simple pattern only utilizes a dataspace connector to transfer files in a bilateral exchange scenario. In this case, intermediate persistence such as an object store is used for further analysis by specific business apps (see Figure 10). In addition, Catena-X semantic models might be used to standardize information to be exchanged.

[8] Extract from the CX-0055 standard

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

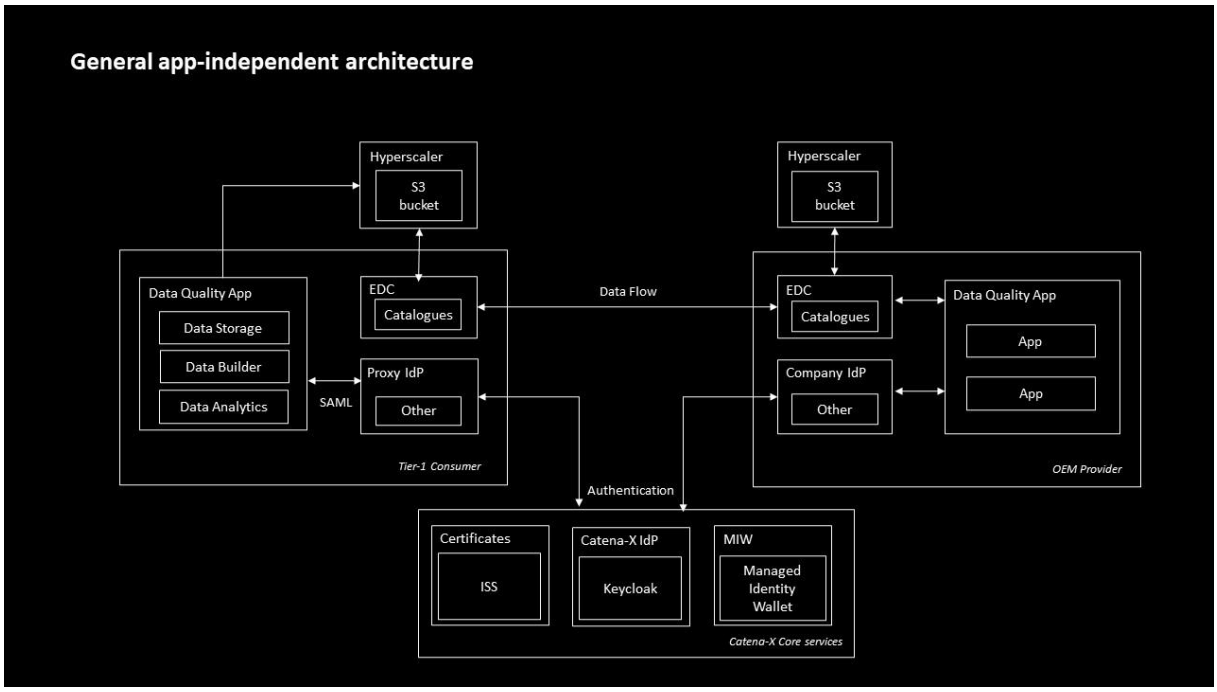

Figure 10: Example for file-based pattern (Quality Use Case[9])

6.2 Digital Twin/Asset Administration Shell

6.2.1 Semantic Models and Digital Twins

Data integration within the Catena-X data space is enabled by standardized semantic models (see step 2 of the six-step process in chapter 4). Semantic models are the basis for not only sharing data, but information. It is crucial that every partner in the ecosystem does understand which kind of data is required to participate in specific use cases and what the provided data means. Only then, business applications can be built on top of it. In other words, semantic models are the backbone for semantic interoperability.

Catena-X offers a set of these standardized semantic models. They are published in the standard library of Catena-X[10]. These semantic models are also available as open-source machine readable specifications in Eclipse Tractus-X[11].

A business application typically needs specific data (depending on the use case) about a specific vehicle, a specific component, or a specific product type. This asset-oriented gathering and providing of data is best supported by the digital twin pattern. A digital twin is representing an asset and serves as a single contact point for all data around this asset. "Asset" is a very generic term: it can be a vehicle, a gearbox but also more abstract entities like a fleet of vehicles, a vehicle type, an

[9] https://eclipse-tractusx.github.io/docs-kits/kits/Quality-Kit/Adoption%20View%20Quality%20Kit
[10] https://catenax-ev.github.io/docs/next/standards/overview
[11] https://github.com/eclipse-tractusx/sldt-semantic-models

organization etc. The only prerequisite is that this entity has so much value to the organization that this entity is considered an asset and therefore has a unique identifier.

A digital twin provides access to a collection of different aspects. Each aspect has a semantic model associated with it, so that the meaning of the data for this aspect is clearly defined. The data is not static but is synchronized with the different data sources depending on the data integration patterns (see chapter 5) chosen to implement the aspect. The collection of aspects can be extended any time. Registration and discovery of digital twins is done via a Digital Twin Registry.

In some cases, the data consumer does not know the identifier of the relevant data provider information - the business partner number (BPN) within the Catena-X dataspace - for the asset under consideration. Therefore, additional discovery services should be provided to find the relevant dataspace connector.

IEC 63278 standardizes the Asset Administration Shell. The Industrial Digital Twin Association[12] is the home of the Asset Administration Shell. The Asset Administration Shell specifies an interoperable digital twin implementation together with standardized APIs. Catena-X is using the Asset Administration Shell standard for digital twins (see CX-0002[13]).

6.2.2 Central Digital Twin Registry

For the first releases of Catena-X a central Digital Twin Registry was set into place as a central service. However, this has many disadvantages with respect to data sovereignty and scalability. Therefore, this approach is not recommended for data spaces.

The data provided via the aspects of a digital twin is not contained in the Digital Twin Registry itself. The registry just contains the endpoints and additional meta information. The data itself is made accessible via the dataspace connector of the corresponding data owner.

Note: A central digital twin registry today is no longer supported in Catena-X.

6.2.3 Decentralized Digital Twin Registry

In a decentralized Digital Twin Registry approach, every company participant needs to either (see also chapter 3.6):

1. set up their own Digital Twin Registry and ensure operation of the Digital Twin Registry. Open-source solutions for setting up the Digital Twin Registry can be considered to be used as basis.

[12] https://industrialdigitaltwin.org/
[13] https://catenax-ev.github.io/docs/next/standards/overview

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

2. subscribe to an operated (certified) Digital Twin Registry service.

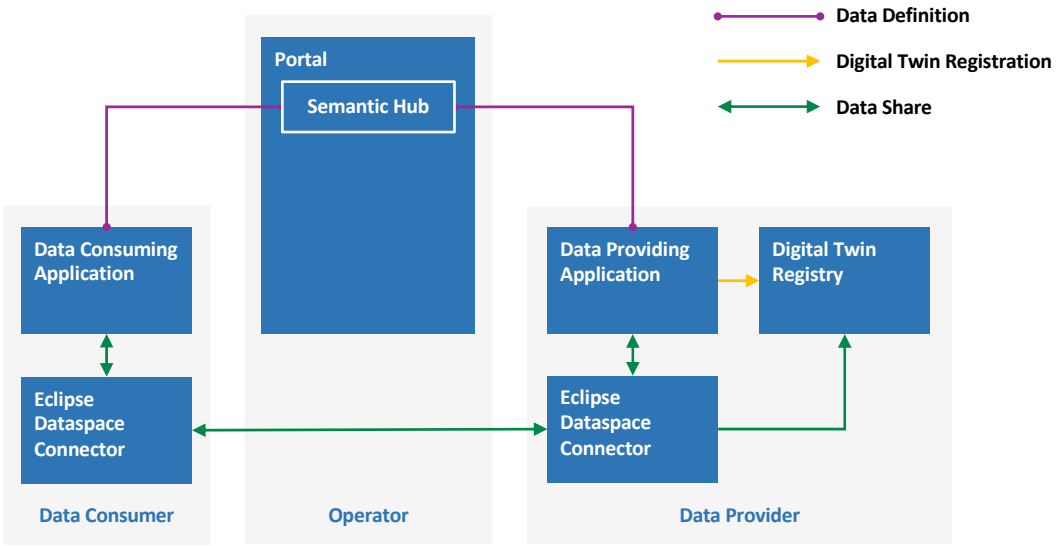

Figure 11: Pattern for decentralized Digital Twin Registry

Each data provider needs to register their decentralized Digital Twin Registry for discovery of the Digital Twin Registry itself. Either there is a central discovery service available at the Catena-X operating company or the Digital Twin Registry is registered as a data asset in the dataspace connector of the provider.

The Digital Twin Registry only contains meta data. Nevertheless, the Digital Twin Registry needs to scale. The goal is to have a digital twin for every product type and product instance for which data is exchanged in the Catena-X data space, which are all exposed through one single DTR.

Hence the recommendation is to have the Digital Twin Registry registered in the dataspace connector of the data provider. There might be dataspace connectors without a Digital Twin Registry, for example for exclusive data consumers. The aspects registered per digital twin typically are accessed via the same dataspace connector the Digital Twin Registry is registered at. However, this is not a prerequisite as Catena-X' original central Digital Twin Registry approach showed.

6.3 Knowledge Agents

The main objective of the Knowledge Agents (KA) data processing pattern described in this section is to create a state-of-the-art compute-to-data architecture for automotive use cases (and beyond) based on standards and best practices from Gaia-X and W3C.

The most important concepts needed for its realization are summarized in Figure 12.

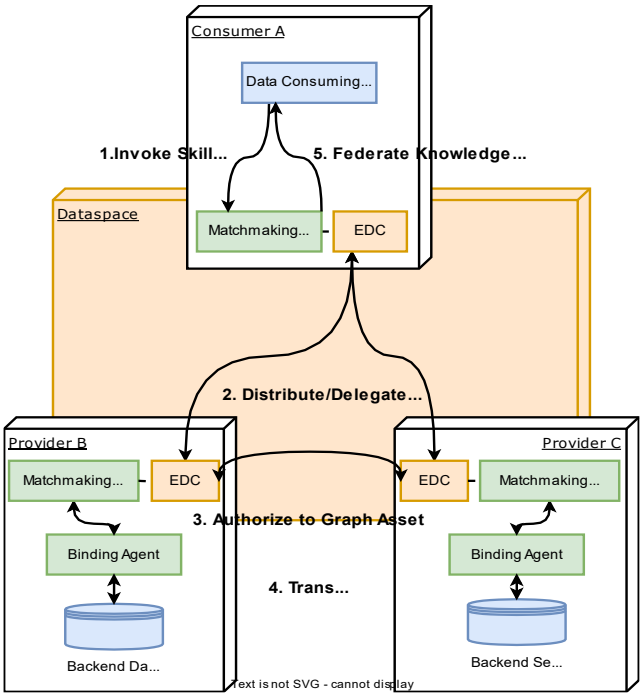

Figure 12: Overview Knowledge Agents Concept

The Data Consuming App in the figure serves the consumer by gathering, analyzing, and presenting the knowledge about business questions such as: How much of a certain material can be found in a specific vehicle series? It is assumed that the data which is needed to answer such questions is distributed over the network and cannot be found at one central place.

To help collecting the data over the network, Skills are introduced. A Skill is a pre-formulated query (or: procedure) with limited scope such as: List all vehicle series that contain material produced in a certain location. The Skill is used to access all federated data instances via a tenant and receive input in the form of a data set.

To obtain the correct results in a federated system, all participants of the Skill execution need to have a common understanding of their vocabulary. Relying on these conventions, an executor of a Skill can calculate which providers can contribute the necessary information and in which sequence requests must be performed, such that the resulting distributed operation works.

This coordinating task is taken over by the *Matchmaking Agent*, an endpoint that is mandatory for any KA-enabled data space participant. For that purpose, the Matchmaking Agent supports the SPARQL specification with the effect that the data space can be traversed as one large data structure. Hereby, the consumer-side Matchmaking Agent will – as driven by the built-in federation features of SPARQL - interact with the KA-enabled EDC to negotiate and perform the transfer of Sub-Skills which are partial expressions of the original SPARQL command to other data space participants.

In turn, upon successful transfer of the Sub-Skill, the provider-side Matchmaking Agent(s) will be activated by their respective EDC. The precondition for this activation is that the provider EDC first needs to offer a so-called Graph Asset. Graph Assets are a variant of ordinary Data Assets in the Catena-X EDC standard, while Data Assets typically refer to an actual backend system (e.g., an object in an Object Store, an AAS server, or a REST endpoint).

Graph Assets introduce another intermediary instance, the so-called Binding Agent. Simply put, the Binding Agent is a restricted version of the Matchmaking Agent (which speaks a profile, i.e., a subset of the SPARQL specification) which is translating Sub-Skills of a particular business domain (Bill-Of-Material, Chemical Materials, Production Sites, etc.) into proper SQL- or REST-based backend system calls. This scheme has several advantages:

- For different types of backend systems, business domains and usage scenarios, different Binding Agent implementations (Caching Graph Store, SQL Binding Engine, REST Binding Engine) can be switched-in without affecting the shared data space/semantic model and the largely immutable backend systems' data models.
- Access to the backend system is decoupled by another layer of security, such that additional types of policies (role-based, row-level and attribute-level access) can be implemented in the interplay of the Matchmaking and Binding Agents.

As mentioned earlier, essential for the realization of the idea is the creation, governance and discoverability of a well-defined semantic catalogue (the Federated Catalogue) which together with the data inside the Graph Assets form a Federated Knowledge Graph.

6.4 Use-Case specific API

In addition to the previously introduced data processing patterns, some Catena-X use cases utilize dedicated, custom APIs to steer the data transfer. In those cases, a business application is required to consume the specific API. There are also hybrid patterns that use a use-case specific API in combination with a Digital Twin approach. The authors of this document advise that use case specific APIs conflict with the idea of realizing use case agnostic data provisioning and consumption procedures. With this approach, scaling the network becomes more difficult and costly for data providers and consumers as they participate in more than one use case. Thus, it is recommended to avoid or transition from use case specific APIs to one of the three patterns introduced above.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

7. PATTERNS FOR DATASPACE CONNECTOR OPERATIONS

Data integration patterns can be combined with different ways of operating the external gateway – the dataspace connector – each with its pros and cons. The following operational modes are to be understood in the context of company internal operations and not for the entire Catena-X ecosystem. First, it is possible to implement just one central dataspace connector in a company which is responsible for routing all incoming and outgoing data access. It is also possible to set up a distributed connector operations landscape, meaning that there are several connectors deployed which are connected to just some data sources each. A company might even connect to multiple data spaces, beyond Catena-X that might require different approaches to data space connectivity. Each of these operational modes has advantages and shortcomings. Further information regarding the setup and configuration of dataspace connectors can be found in the Tractus-X Connector KIT.

Setup and connector operations need to be separated from data pipeline implementation efforts. Data space connectivity, independently from the specific use case, represents a workload that is typically provided as a service to internal teams, that is either hosted on its own or sourced from a certified Enablement Service Provider. As data entering and leaving company boundaries comes with requirements on the data security and governance sides (see Chapter 8), there should be a central responsibility in place, ensuring that mechanisms for auditability, lineage, security and observability of data exchanged are met by all entities requiring data space connectivity.

Data pipelines, that integrate connector instances and their data assets with backend systems on the other hand, are part of a specific use case. As such, implementation logic and requirements are with the entity responsible for its implementation and operational readiness.

7.1 When to use a dataspace connector

The dataspace connector is a component for sovereign cross-company data exchange. It shall be used whenever information or data needs to be exchanged between two separate legal entities. The focus lies on between legal entities. A data provider creates contract offers for data which a data consumer agrees to. A bilateral agreement has therefore been made. If a Catena-X certified business application is used by a data provider or consumer, this app typically comes with its own EDC. While all data transfer cross company happens via this EDC, it is not necessary to also use it for company internal data ingestion from e.g., a backend system to this newly purchased app. For ingesting data from own source systems to own business applications, regular ingestion mechanisms should be used as the data-transfer happens within one's own legal entity.

Also, in the case of larger global enterprises, spanning multiple business units or even businesses, the dataspace connector and data transfer via a data space is typically used for exchange with entities external to this group of entities, such as suppliers and customers. Internally, data is exchanged as per the enterprise's internal data management and governance structures.

7.2 Single Connector Operation

7.2.1 Description

A single-connector operation is characterized by just one single dataspace connector within a company or legal entity which is connected to all required internal data sources and is responsible for the entire external data traffic. Such an operational structure might be beneficial if the overall number of connected systems and use cases is small, and requirements to identity, access and ownership of data are simple enough to be represented by a single EDC instance and a corresponding team administering it.

With this approach it is important to ensure non-functional requirements and service-level objectives (SLOs) are defined and met by the single-connector architecture to meet targets for availability, scalability, transaction throughput etc. as all company use cases will depend on the operational readiness of this single connector. This operational readiness should be included and tested extensively as part of the company's connector release management.

7.2.2 Advantages

- Architectural elements such as load balancers, auto-scaling and failover capabilities, API gateways and web application firewalls can help protect dataspace connectors from operational failures. Setting up just one single connector per company yields less setup and configuration effort and can be done rather quickly (within 2 to 3 days).
- Maintaining just one dataspace connector is easier and quicker in terms of updating and bug fixing as just one component must be considered. Additional components to manage and monitor a fleet of connector instances are not needed.
- By implementing just one single connector less computational resources are required, although its requirements can be considered as rather low for server infrastructure in general. However, a single dataspace connector binds less baseline resources compared to a connector fleet in a distributed manner.
- A single connector can be more easily tailored to meet company specific requirements on data management and governance as well as controlled to ensure that corresponding mechanisms are enforced across backend systems and use cases, such as auditability of all negotiations and transfers occurring with external entities.

7.2.3 Challenges

- As the dataspace connector is still in its early product lifecycle phase, performance and latency issues might arise due to many or big data requests which are to be handled simultaneously. Load test extensively.
- Maintaining and configuring contract offers for all use cases and data sources on just one connector might be challenging, due to issues with ownership and responsibility of data as well as security concerns – every actor who has access to the control plane's API key credential can modify and delete any data exposed by the company across all use cases.
- In the case of connector defects, all operations are affected. Malfunctions which are limited to one single use case might cause disturbances across all use cases, as there is no isolation boundary between them. The connector acts as single point of failure for cross-organizational data exchange.

7.3 Distributed Connector Operation

7.3.1 Description

A distributed connector operation is characterized by several dataspace connectors within a company which are all connected to different (groups of) data sources, thus reducing the number of connected systems per connector compared to the single-connector mode. Such a structure is beneficial if many different data sources must be accessed as use case specific configuration and maintenance effort per connector are reduced as there are several of them available.

With this approach it is important to not have dataspace connectors being hosted and maintained by arbitrary teams across the company. Having a number of teams, for example those owning the respective use cases, all having to handle connector architectural design, configuration and maintenance is not operationally efficient and requires a skillset that is likely not broadly available across the company. Instead, even with a distributed connector approach and favoring multiple connector instances for use case isolation and individual configurability, there should still be one team owning this multi-connector environment, that provide and maintain it as internal multi-tenant service for teams to request and consume.

There has to be central observability into connector versions used, infrastructure configuration, as well as fulfilment of data security and governance requirements that apply across data transfers entering or leaving the company or legal entity (see early chapters of this doc). Handling a multi-connector environment as one central workload allows for enforcement of such mechanisms.

7.3.2 Advantages

- Having specifically implemented dataspace connectors per use case allows for better access control to contract offers, negotiations and transfers as these can be managed on each connector in a use case specific manner.

- Performance and latency issues come with a reduced blast radius as connectors connect to less data sources, which reduces the risk of overloading and thus facilitates proper connector operation without issues.
- In case of bugs and defects, other connectors might remain unaffected; thus, their operation can still be resumed, and use case applications remain up and running.
- Connector extensions can be implemented to enable use case specific requirements without increasing the complexity of other connector instances and their respective use cases and data routing mechanisms.

7.3.3 Challenges

- Setting up several dataspace connectors comes with additional setup cost in terms of time, effort, and resources needed, which might be even higher for maintaining several connectors in case of necessary updates or with bugfixes. As part of an internal connector-as-a-service workload, management components should be included to simplify and scale operations across a growing fleet of connectors to address this challenge. For example, Kubernetes comes with capabilities to make updates and configuration changes in bulk as well as automatically handle specific operational events. Defined runbooks for both incident and patch management help streamline and accelerate operational tasks. Having one team owning workload resiliency, availability and operational excellence helps with overall organizational efficiency in managing dataspace connectors.

7.4 Multi-Dataspace Connectivity

7.4.1 Description

As Catena-X and related initiatives such as Manufacturing-X mature, the requirement emerges for legal entities to connect to multiple data spaces simultaneously. Requirements to connectivity might not be identical between data spaces. Architecturally, companies should define connectivity to a data space as a workload that is provided once per data space (see section “Distributed Connector Operation”). For Catena-X, there would be either one or multiple dataspace connectors that follow Catena-X' standards and requirements. For a different data space, a different connector might be used, whose requirements would be fulfilled by providing a separate, potentially multi-tenancy connectivity workload with a team owning its internal functionality.

Concludingly, there should be one workload or application that provides an internal service for dataspace connectivity per data space in which a company participates for use case teams to consume. The team owning the respective connectivity workload is responsible for ensuring operational readiness and for selecting the right architectural pattern(s) to use. Separate data pipelines are needed for every data space that has to be supported by a specific use case as semantic models and standardization requirements to the data exchanged might differ between data spaces.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

8. DATA GOVERNANCE AND SECURITY

This chapter takes on a technical view on data governance and security, including audit trails, logging, and data quality control. The topics covered are relevant independently of the data integration and processing patterns that are implemented within a company. However, based on those patterns, an implementation varies and might become easier or more difficult. This chapter is not about operations of the core network components, it exclusively targets the decentral components that are deployed at each individual participant company. Furthermore, this chapter does not cover all aspects of data governance and security. It will evolve over time and new aspects will be added as requirements emerge.

8.1 Data Governance and Release

Data governance and release is mainly an organizational topic and there is a complete separate guide on it on the Catena-X association homepage: https://catena-x.net/en/catena-x-introduce-implement/onboarding. Therefore, this chapter will focus on the implications for IT and data integration and only those governance aspects that are required to understand the technical side are mentioned.

As releasing data to, as well as consuming data from external partners comes with a greater responsibility for security and safety, not one single person should be able to execute those tasks. Thus, it is advised to either implement a four-eyes-principle or even a data exchange board that decides about external data transfers.

For the time being, there are little to no tools to connect arbitrary backend systems with dataspace connectors as part of data pipelines, for either data provisioning or data consumption without the involvement of IT personnel. This also means that IT can act as a proof point to check if it is allowed to release or consume certain data before data pipelines are implemented. Developers should work based on tickets, where also the check for data release by a board or manager could be tracked. After a data pipeline has been implemented, a User Acceptance Test (UAT) should be done to validate if in case of data provisioning the right data is provided under the right conditions (usage policy) to the right parties (access policy). Or, in case of data consumption, that the consumed data is stored in the right way and sent to the right backend systems.

Both organizational data governance processes as well as IT tasks such as the creation of assets or connection of backend systems with the EDC should be automated and integrated into the participant's system landscape. Depending on an organization's data maturity and established data management and governance mechanisms, including creation and templating of data pipelines, it is alternatively possible to start with manual processes such as Jira tickets or emails for documentation and visibility, to learn about obstacles and internal challenges, before implementing governed, automated processes.

Independently of the pattern that is implemented, an audit proof documentation of data that is released and consumed is required. In the target picture, this might become easier with a centralized toolchain as part of a data lake or data virtualization layer, as data release or consumption can be tracked and logged at a central instance.

Depending on the use case that a company wants to participate in, there might be usage restrictions expressed in usage policies, such as „this data shall only be used for the analysis of quality issues". If data consumed is first stored in an intermediate layer such as a data lake, the data consumer needs to track and enforce this usage policy and needs to make sure that data isn't consumed by applications or individuals that don't match the policy.

Administrative access to data pipelines, dataspace connectors and Digital Twin Registry needs to be limited to authorized personnel, so that only those can access the underlying databases, containers, and other resources who have a justified business need for it. Also, it might make sense to separate connector instances either with separate deployments or with namespaces within a cluster to guarantee segregation between two business units that aren't allowed to access each other's data (see section "Distributed Connector Operation"). A segregation might also be necessary to allow for separate accounting and cost control of used resources across different business units. Note that multiple connector instances can still operate under a single BPN(L), and that this separation is only for internal process reasons, such as authentication and access.

8.1.1 Access to data assets

Data provider view

Depending on the pattern that is used, limitations for creating contract offers could be inherited from the access rights to the data product (i.e., only a specific role can create a contract offer for that data product). Keep in mind that the creation of a contract offer isn't enough: The data pipeline that connects the storage layer with the connector still needs to be implemented individually. Also, the dataspace connector doesn't have the capability to restrict actions to certain roles, as this is highly individual to each company. Company internal IT would need to extend the connector with a rights and roles concept that fits the company.

Data consumer view

Access to existing contract offers by providers can't be limited to specific roles. This means that – through the connector – every authenticated actor can potentially agree to every contract offer for which the provider's access policies are fulfilled. If restrictions on usage (e.g., only for the quality department) are imposed by the data provider, the data consumer needs to consider those when storing the data in a data persistence layer from which the data is later accessed or when ingesting data directly from the dataspace connector into a business application.

A company should separate access to the connector from access to the actual data. Everyone can see all contract offers, but only authorized stakeholders can access an actual data asset and create the data pipeline that ingests data into the storage layer. Also, there should be a four-eyes principle implemented that does not allow the same individual that built the data pipeline to also deploy and

modify it in production, presumably in accordance with the company's existing processes for production releases of IT systems.

Access rights to the newly created data product are set accordingly and only people with the right role can access the data product at the storage layer. If a business application is used for data consumption, the roles concept of the application should take usage policies and potential separation of data by consumer role into account.

Example:
Company A creates a contract offer for CO2 values of brakes. They specify that this data shall only be used for CO2 calculation purposes and that the data should not be passed on to the quality department. In the access policy, Company B is defined as the only authorized consumer. At this point in time, every employee of Company B who has access to a Company B dataspace connector can see this contract offer and can agree to it. The purchasing department of Company B now wants to access this data asset of Company A, to use it in two company internal CO2 reporting tools and creates a ticket for the IT department, to create a data pipeline that consumes the data asset and stores the data in the data lake of Company B, to then concludingly distribute the data to the two reporting tools. Along with the new data product, the asset's usage policy (data shall only be used for CO2 calculation purposes and that the data should not be passed on to the quality department) needs to be recorded. An employee of the purchasing department becomes the owner for the newly created data product, usually supported by the team owning data management and governance at the company who support individual business teams with their use cases and data products.

Now an employee of the quality department requests access to the data product of CO2 data. The purchasing department – as the business owner of this data product - needs to decline this request, as it would violate the asset's usage policy imposed by the data provider.

In a second scenario, an employee of the quality department, who has access to a Company B dataspace connector sees the CO2 contract offer and would like to consume the data. He creates a ticket for the IT department to build a data pipeline. The IT department needs to check the usage policy and decline the request, as it would violate the policy imposed by the data provider. In case the IT department does not feel capable of deciding on such requests, as they don't understand the specifics of business-related policies, a governance board for cross-organizational data transfer could be established to make those decisions.

This example shows that, depending on a company's internal processes for data provisioning and data consumption, different checks and proof points need to be implemented and enforced so that the legal framework of Catena-X is not violated.

8.2 Audit Trails and Logging

This part of the guide focuses on requirements regarding audit trails, logs, and log retention with respect to transferred data – independently of the data integration pattern in place. You also might want to keep an audit trail of created contract offers, policies and contract agreements. However, this is rather static data, since contract offers are only created once and hence might be logged differently than operational data of ongoing data access. Furthermore, through UATs and IT processes, the creation of assets, policies and contract offers is already documented.

Both consumed as well as provided data should be logged for various reasons:

a) Antitrust-Law: Although, through access policies, it should be impossible for competitors to access certain data, there are still humans involved. Also, there are use cases where competitors in fact do exchange data. It will be important to prove who had and hadn't had access to data in case of doubts.
b) Use-Case-Relevance: There might be use cases where you want to prove that data was transferred to a partner: e.g., a quality alert that should trigger a recall was sent in time.
c) Information Protection: Based on the sensitivity and required level of protection of the data, access needs to be logged. Access logs of data with a higher level of protection should (and in some cases must) be monitored proactively for anomalies. Although hacks should be highly unlikely due to the decentral nature of the network, in case of a hack, a company will need to be able to trace, which information was leaked, from where it was accessed and when.

8.2.1 Types of Logs

Different types of logs can be distinguished, for which different retention policies and periods apply. The types of logs in the context of Catena-X are listed below. Considerations on log retention policies and periods are discussed in a subsequent chapter.

(Extended) Server Logs

This log type covers all the logs that software components – typically containers, APIs, databases etc. produce. This can include stack traces and exceptions, depending on the log level. Extended Server Logs can be used for debugging, security or monitoring. The transferred data itself can't and shouldn't be logged completely, as the volume of data is too large even for a short-term duration (This would essentially duplicate the data with each request).

(Error) Logs for Customer Support

Logs for customer support are designed to be readable and understandable by non-experts. In the context of Catena-X this could include failed and successful data transfers and contract negotiations. The goal of this type of log is to allow customer support to tell the customer (in this case the data consumer), what went wrong and what to do or whom to contact next.

Audit Logs

The who, from where, when, what. Audit logs aim at creating an understanding of the actions that

happened on a business level (compared to server logs, which explain what happened on a technical/code level). In the context of Catena-X, audit logs can be distinguished in two categories.

Audit logs for data exchange: Those contain all the information about access to data assets from external partners as well as access to external assets. This can include the BPN of the partner or the connector and asset IDs of the executed query on the data asset.
Audit logs on assets and contracts: Those contain information about created and agreed contracts, changes of policies and assets, deleted assets or contracts etc.

Archive of Contract Offers and Agreements

The dataspace connector has a database of all contract offers and agreements. However, this data should also be archived for long-term storage due to requirements from antitrust or compliance. The archive should also contain the access and usage policies for each contract offer and agreement to prove who had and hadn't had access to specific assets.

8.2.2 Log Extraction

Logging must happen at multiple points due to the different available information at the various technical components.

Connector Data and Control Plane

The control plane of the dataspace connector stores information about the identity of the data consumer that agreed to a certain contract as well as all contract and offer details. This data is stored in a dedicated database. This means that it's not absolutely necessary to also log all contracts - it should be enough to access the database, if details about offers or agreements are required.

The data plane is the access point that is called every time a data transfer is initiated. To check for contract validity, the data plane interacts with the control plane on every call. Technically, both components can be used to log e.g., date and time of the request, BPN(L), connector ID or IP of the requestor and the details of the accessed asset.

Data Pipeline to Source/Target

Log the exact query that was executed on the data source as well as header information, the first lines of results, the exact semantic model, or other metadata.

Note: It's possible to simplify this logging model by passing information from the data/control plane to the data pipeline (or vice versa) and only log information from one component. This however requires additional development as the EDC does not support metadata propagation out of the box.

8.2.3 Log Retention Duration

From a data protection point of view, logs should only be kept until the purpose for which the logs are intended is fulfilled. I.e., if logs are needed to resolve customer requests in case data transfers

fail, the logs for data transfer should be kept for the maximum duration that is needed until a customer typically detects the issue, opens a ticket and customer support resolves the issue. In other words, data protection defines the maximum (not minimum) duration that logs should be kept. When implementing a log retention policy for the different types of logs, a company should look at the duration of their internal processes, typical response times to certain events (e.g., debugging, ticket resolution times) and then define for how long they want to keep the different types of logs.

The periods for which access to data must be logged will vary from company to company. It is always a good idea to consult the data protection, information security and/or anti-trust/compliance departments for advice on how to implement a suitable log retention policy for Catena-X. The list below is only a proposal for the maximum log retention duration for different types of log files and by no means a strict guideline that needs to be followed:

- Extended server logs: 7-14 days
- Logs for customer support: 30 days or until the issue is resolved
- Audit logs: up to 6 months
- Archive of created contract offers and contracts: up to 10 years

8.3 Data Quality

Monitoring the quality of data that is distributed to as well as received from partners is essential if the business shall trust the decisions that it makes based on that data. Data quality is a challenge with respect to cross-company data transfers. Data providers should proactively monitor the quality of the data they provide, and data consumers should check their received data for validity and conformance. Depending on the use case, Catena-X standards define additional rules that can be used for this purpose, such as with product carbon footprint (PCF) and its PCF rulebook.

Catena-X also provides possibilities to check for data quality with its semantic models. If described correctly, those models not only define the meaning of a certain attribute, but also which data types or values are allowed. Data engineers can use the JSON schema SAMM (formerly known as BAMM) models to implement customized data quality checks of received or provided data. With checks for data semantics in place, which are also needed for Catena-X conformity assessment (see Chapter 9), further data quality checks can be performed for the values of a provided or received data asset.

Plausibility checks should be included into data pipelines between dataspace connectors and participants' IT systems, that, while performing the transfer between connector data asset and backend system not just transform data between applicable schemas, but also verify that values contained lie within plausible ranges, follow expected patterns and are not anomalous compared to related records. Partners also need to define processes to remediate data quality issues – from preventing an EDC data asset from being refreshed to involving responsible teams to perform more in-depth analysis of data quality issues, where the right approach depends on the use case and criticality of the data being exchanged. In general, the owner of the provided data is responsible for data accuracy and quality for others to consume, both internally as well as externally.

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

9. CONFORMITY ASSESSMENT AND CERTIFICATION

The topic of data integration has to be considered in the context of standardization and certification to some extent. Thus, the following guiding questions shall help to get a better understanding of what needs to be done by which role.

Why do we need standards and certification in Catena-X (in general)?

See Catena-X' webpage[14] for further details on standardization and certification. There, you can also find the latest standards. The Catena-X operating model[15] gives more general advice about the different roles and the principle of standardization/certification itself.

Why do we need standards for data integration?

Providing and consuming data is one of the core elements of participating in the Catena-X data space. It needs to be ensured that data is supplied in the right format to ensure that use cases work as expected in an interoperable manner and the data-chain is not interrupted.

Who is affected by certification against these standards?

Standards are relevant for all data space roles as they define how the Catena-X ecosystem works. However, only Enablement Service Providers, that offer commercial solutions for technical Catena-X onboarding (so called Enablement Services), need to certify their solutions against these standards[16]. Companies that just want to provide or consume data can either source one of the existing enablement services of a commercial provider or build a solution themselves. In the latter scenario, those self-developed services do not need to be certified (as long as they are not offered as a commercial offering to other data space participants).

I want to offer single components as an Enablement Service, what should I do?

Companies require decentral software components to participate in Catena-X use cases. Those components, such as the Dataspace Connector (CX-0018) or the Digital Twin Registry (CX-0002), need to be compatible with other services of the data space and the current Catena-X release. Thus, standards dictate minimum requirements for reference implementations (FOSS or COTS) for

[14] https://catena-x.net/en/catena-x-introduce-implement/standardisierung

[15] https://catena-x.net/en/about-us/operating-environment-1

[16] https://catena-x.net/en/catena-x-introduce-implement/offering-a-catena-x-solution

these important components. Companies that want to provide commercial offerings for enablement services need to be certified against the corresponding standards.

I want to offer a solution for data integration as Enablement Service, what should I do?

Solutions for data integration usually go beyond offering a single service as they orchestrate all required components for data integration. This includes two clusters of services:

- **Company-individual data integration solutions**, e.g. backend systems, data lakes, data pipelines (referred to as "Data Integration Patterns", see chapter 5).
- **Catena-X standardized components/services**, e.g. enablement services or APIs relevant for specific use cases (referred to as "Data Processing Patterns", see chapter 6).

Relevant for standardization and certification is only the second category, as company-individual solutions can hardly be standardized. In this case a whole group of standards will be relevant for certification depending on the respective Data Processing Pattern.

The triangle standard CX-0055 was developed exactly for this purpose, as it associates Catena-X' Data Processing Patterns with the respective standards. By certifying against CX-0055, the Enablement Service Provider can choose against which patterns the solution shall be certified. Use cases that work with respective patterns can then be addressed by the certified solution.

Which standards are relevant for me?

The standards that are relevant for certification are continuously reworked and therefore subject to ongoing changes. Thus, the latest version of the "Modular System[17]" for certification needs to be considered as well as Catena-X' Conformity Assessment Framework Handbook.

[17] https://catena-x.net/en/catena-x-introduce-implement/certification

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages

10. OUTLOOK

10.1 Terminology

Dataspace Connector Contract Offers

In simple words, this represents the conditions under which a data provider offers a specific data set to data consumers. It describes:

a) the content that the consumer will receive (e.g., CO2 data for a vehicle)
b) the contractual obligations that the consumer needs to fulfil (e.g., don't pass data on to third parties).

Data Pipeline

While the connector contract offer provides the endpoint under which data can be retrieved, it doesn't provide any data itself. The endpoint needs to be connected to the actual data asset. This is done by data pipelines.

Internal data products/assets/sets

There are various terms with slightly different meaning depending on the context. In this guide they all refer to the actual data asset that is accessed via the data pipeline and offered through a contract offer.

10.2 General Considerations

- The internal data sets should be scoped and built in a way, that one data set reflects exactly one so called "sub-model" or semantic model. This allows that one data pipeline only needs to connect to one data set to extract data and the whole scheme can be used.
- One connector contract offer reflects one data asset that a data provider wants to offer to a group of data consumers (access policy) under the same conditions (usage policy). If a data provider wants to offer the same data asset to two data consumers but with different conditions (usage policies), he would need to create two separate contract offers, one targeted to each data consumer.
- A data asset is static and complex to build up. Meaning that it's not easy to create a data asset "on the fly" and it is usually also not easy to maintain many similar data assets. It would be hard to create one data asset for each customer. This implies that not all potential customers of a data asset should be able to access the whole asset. Thus, row (or column) based access control needs to be implemented by data pipelines.
- It might make sense to split data assets into several subsets so that e.g., subset A only contains vehicles or parts from a certain geographic region or from a certain vehicle type/part number. The benefits for this are:
    - faster response times because data sets are smaller

  - faster response times depending on from where in the world data is accessed
  - avoids additional filters in the data pipeline.
- Data pipelines - in their simplest form - could just be a SQL query with a "SELECT FROM WHERE" statement that selects data based on the BPN of the consumer in the WHERE clause. The BPN of the customer can be retrieved from the token that the data consumer must show to the provider. Of course, a data asset would need to contain either the BPN directly or there has to be a translation service (see example below).

10.3 A Concluding Example

A Tier-1 company A wants to offer CO2 information of its products to all its 8 OEM customers.

Option 1:
Company A creates one data set that contains CO2 data for all its products. Company A creates a contract offer that allows the 8 OEMs to access this data. As one product is only sold to one OEM, company A needs to implement a data pipeline that only returns those CO2 values for products that are sold to the customer that consumes the data. If such a filter wouldn't be implemented, OEM 2 could access data from OEM 1 and vice versa. This has the advantage, that only one data set needs to be created and maintained. However, the customer names need to be part of the data set so that row-based access control can be implemented. Furthermore, the Catena-X identifier of the consumer (BPNL) needs to be mapped to the company identifiers used in the data set.

This concept is advised if there will be many distinct consumers.

Option 2:
Company A creates one data set for each OEM containing only the CO2 information of products that are sold to this OEM. Company A then creates one contract offer specific to each OEM and one data pipeline specific to each OEM. While this is much more effort on creating data assets, contract offers and data pipelines, it has the advantage, that no additional filtering mechanisms are needed.

This concept is advised if there are only few distinct customers.