Catena-X

THE FIRST OPEN AND COLLABORATIVE DATA ECOSYSTEM

Onboarding Guide: Data Integration Patterns & Tools

Release V1.1, May 2023

Finanziert von der Europäischen Union
NextGenerationEU

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

TABLE OF CONTENTS

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 must 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 Whitepaper	https://catena-x.net/fileadmin/user_upload/Publikationen_und_WhitePaper_des_Vereins/CX_Operating_Model_Whitepaper_02_12_22.pdf
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.

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.

The subsequent chapter 'Patterns for EDC Operations' evaluates the different options to operate the data exchange component "EDC".

Afterwards, the chapter "industry insights" explains how different providers of integration services implement the Catena-X concepts.

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.

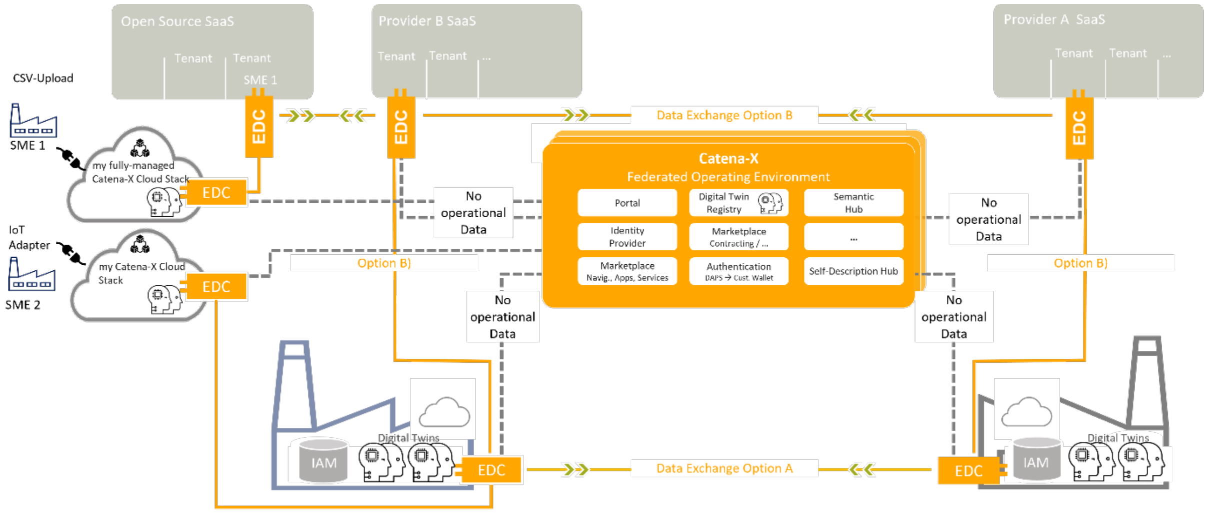


Figure 1: Catena-X Data Space – How Catena-X works / What holds us together? Example

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/

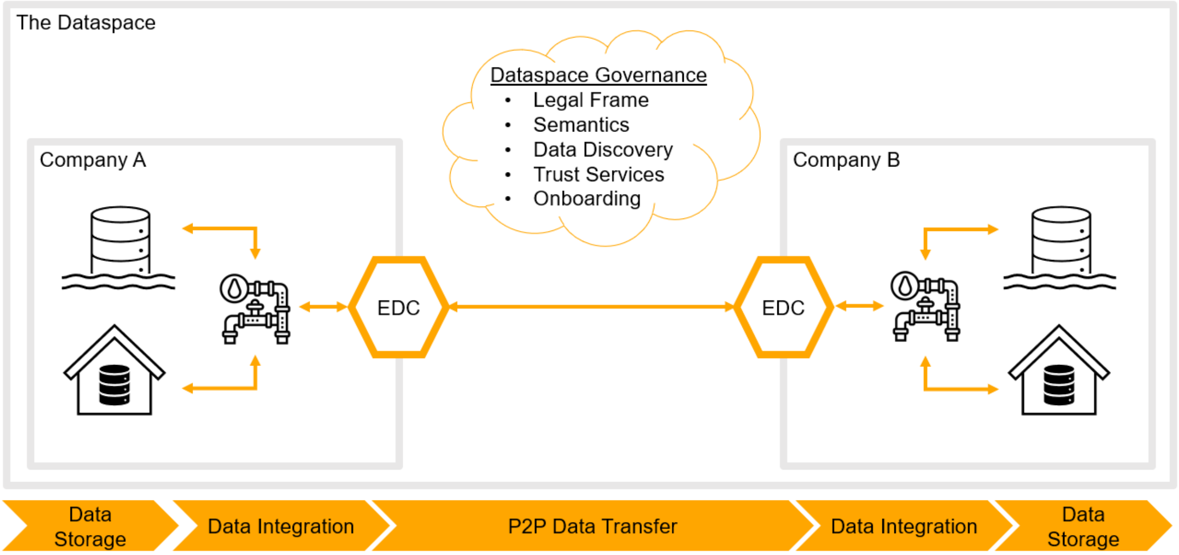


Figure 2: Catena-X Data Space Setup - Simplified

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' for this purpose[2]. 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 send 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 interval. Other use cases might only require a single source system, where data transformation and

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

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

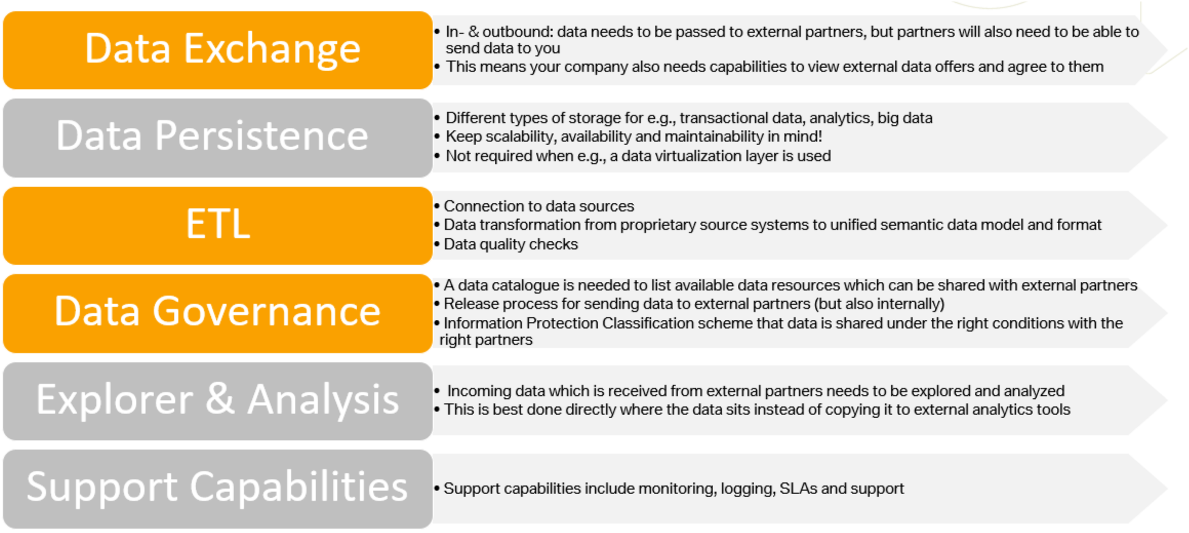
Note: If a Catena-X “certified solution” from one of the app marketplaces is used, the components for data exchange and integration (e.g., EDC, transformation of semantic models) is part of the service offering and doesn’t need to be implemented by a data provider or consumer. Additionally, at some point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

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 focusses 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.



Data Exchange
- In- & outbound: data needs to be passed to external partners, but partners will also need to be able to send data to you
- This means your company also needs capabilities to view external data offers and agree to them

Data Persistence
- Different types of storage for e.g., transactional data, analytics, big data
- Keep scalability, availability and maintainability in mind!
- Not required when e.g., a data virtualization layer is used

ETL
- Connection to data sources
- Data transformation from proprietary source systems to unified semantic data model and format
- Data quality checks

Data Governance
- A data catalogue is needed to list available data resources which can be shared with external partners
- Release process for sending data to external partners (but also internally)
- Information Protection Classification scheme that data is shared under the right conditions with the right partners

Explorer & Analysis
- Incoming data which is received from external partners needs to be explored and analyzed
- This is best done directly where the data sits instead of copying it to external analytics tools

Support Capabilities
- Support capabilities include monitoring, logging, SLAs and support

Figure 3: Data Integration Capability View – General (orange: must have; Grey: optional)

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

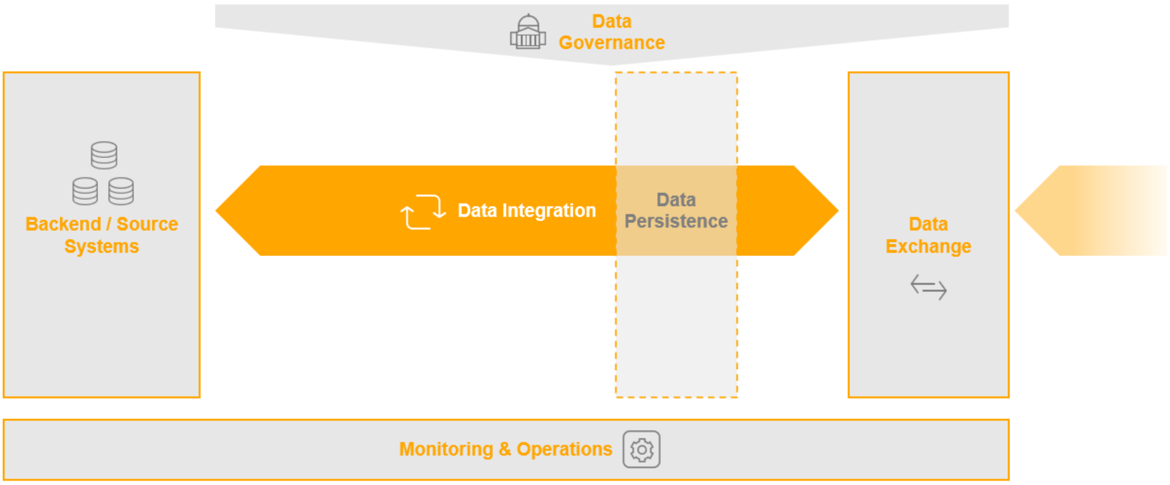


Figure 4: Data Integration Capabilities Architecture

3.1 Data Exchange

The data exchange capability – as the name indicates – covers all areas that are directly involved in data exchange.

Before data can be exchanged, it needs to be made available to external partners (data consumers) so that they can find it. This is done by registering the data that shall be shared at dedicated data discovery services. It will be possible to fine tune, which external partners are allowed to see which data, to guarantee antitrust compliance by design. Similarly, a data consumer needs to be able to communicate with data discovery services to discover the relevant data.

After a data consumer discovered the data, he wants to consume, the actual data exchange needs to be initiated. All typical exchange protocols could be used for that, depending on the type of data that is exchanged: S3, HTTP REST, Kafka Stream, JDBC (Note: Currently only the first two are implemented by a technical solution).

While there are a plethora of different options and vendors to implement the other capabilities, for this one there is currently only one technical alternative: The EDC. This will for sure change later, when there are multiple implementations and vendors available, so that one could choose between e.g., an EDC for edge devices or serverless EDCs for event driven architectures.

This capability is also the only one where there are reference implementations developed by the Catena-X consortia. An overview over those can be found in the documentation of the open-source repository[3].

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

3.2 Data Persistence

Note: Not all data integration pattern will need data persistence. 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 send out or after data is received. It will need to handle different types of data, depending on the use case and data transfer mechanism: Filesystems, (No) SQL-Databases, Streams, etc.

A data persistence layer implies that data is duplicated, and thus mechanisms to control data quality and lineage 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 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 datasets
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.

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 for 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 gapless 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 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 a 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 yet. 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 adaptions 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.

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 models 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.

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

As Catena-X is just about to start, the patterns and advice is based on the experiences and preliminary considerations that Catena-X consortia members have made in their preparations for data integration. The patterns are not industry tested and bulletproof, but they will improve over time.

The options/patterns to operate an EDC are excluded and described in a separate section in this guide called Patterns for EDC Operations. Although the different data integration patterns can be combined with any patterns to operate an EDC, 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.

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

5.1.1 Summary

In the 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 scenario. Here, different kinds of raw data can be stored, processed, and combined to new data sets, while still being centrally available for any 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 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 up for transactional data, no-SQL data, or streaming data.

Thus, the central data storage always 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.

5.1.3 Architecture Overview

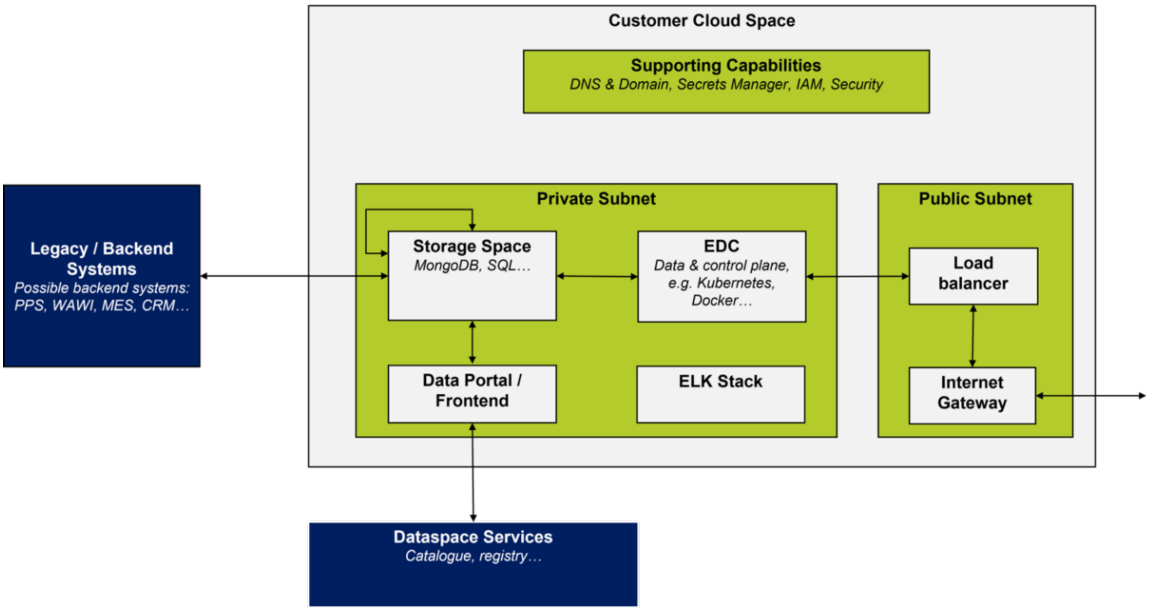


Figure 5: Architecture Overview of Data Lake

For a more detailed and technical architecture, please refer to Figure 11 in Appendix A.

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.

Finanziert von der
Europäischen Union
NextGenerationEU

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

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 to 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 (EDC) and external release processes is simple, as they can be centrally steered.
- As long as EDC and other dataspace specific components are not available as a managed service, operating it once centrally is more efficient than decentrally at multiple departments, because knowledge needs to be built up and maintained just once. Also, the code of the components might change frequently, which means only a one-time additional effort for updates and maintenance.
- 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.
- Data Lineage – as mentioned above might become especially important in the context of Catena-X, as data consumers might need to be able to delete data after a certain period or need to assess if they are allowed to pass on data to a third party. To fulfil this task, they need to know where the data that they received from external sources is stored and used.

General challenges

- Building a central data store including the integration of the most valuable backend systems and creation of data governance structures is a very 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 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. This creates additional effort to setup new storage technologies for potentially just one-use case.
- Participating in multiple use cases and thereby using one pipeline/EDC for the data exchange with Catena-X, logic requirements of different use cases for data provisioning needs to be aligned to ensure correct registration.

5.1.5 Suitability for Different Catena-X Use Cases

- A central data 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 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 semantic 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 the usage policies. It is easier to "delete data after 30 days" if data is only stored in one 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

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 or DCM (demand and capacity management).

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 makes 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

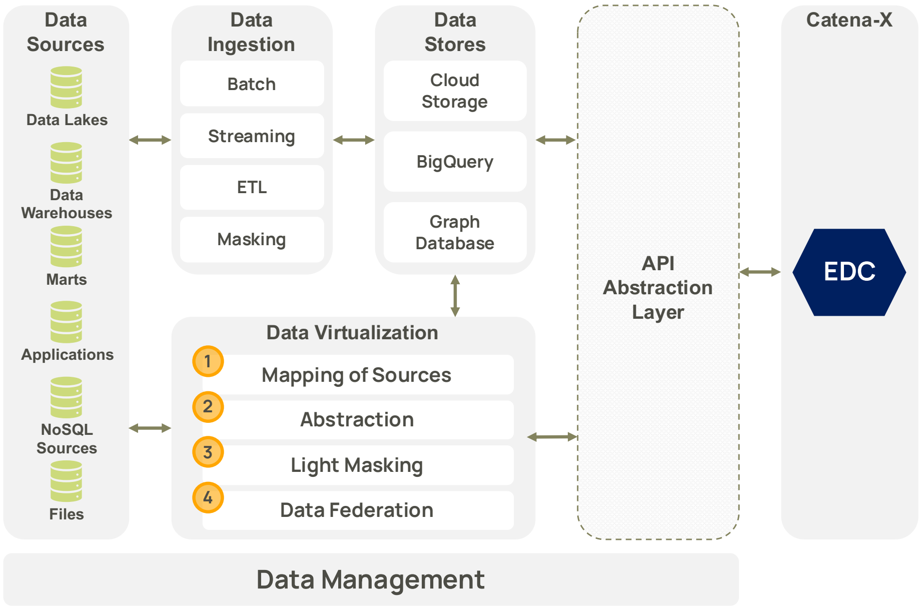


Figure 6: Architecture Overview of a Data Fabric

5.2.4 Advantages & challenges in the context of Catena-X

General advantages

5.2.5 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 / 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, these source systems are decoupled from direct access, hence more secure.
- 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 here instead of sole connections to multiple systems. Moreover, setting up a virtualization layer is relatively easy as less effort is required since data do 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 a coherent data access from different data sources with no restrictions because the virtualization layer serves as a central access point for the connected original data sources.
- Upscaling is easiest 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 the 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 EDC 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 at 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 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 increase response times additional intermediate data storages need to be setup for in- and outbound data flow with Catena-X.

5.2.6 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 cases:

- Demand and Capacity Management
- CO2-Footprint

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 (i.e., the data remains under control of the central IT organization). In case of Catena-X, since each use case fetches data from different domains, this can lead to ownership issues in that the relevant domain recognition can be tricky and one might not instantly be able to figure out who is responsible to provide data access and other rights. As such, the data mesh pattern is suitable for realizing use cases like Quality.

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/ maintaining of the infrastructure without having to rely on the IT operations team. 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 (no middle layer).

5.3.3 Architecture overview

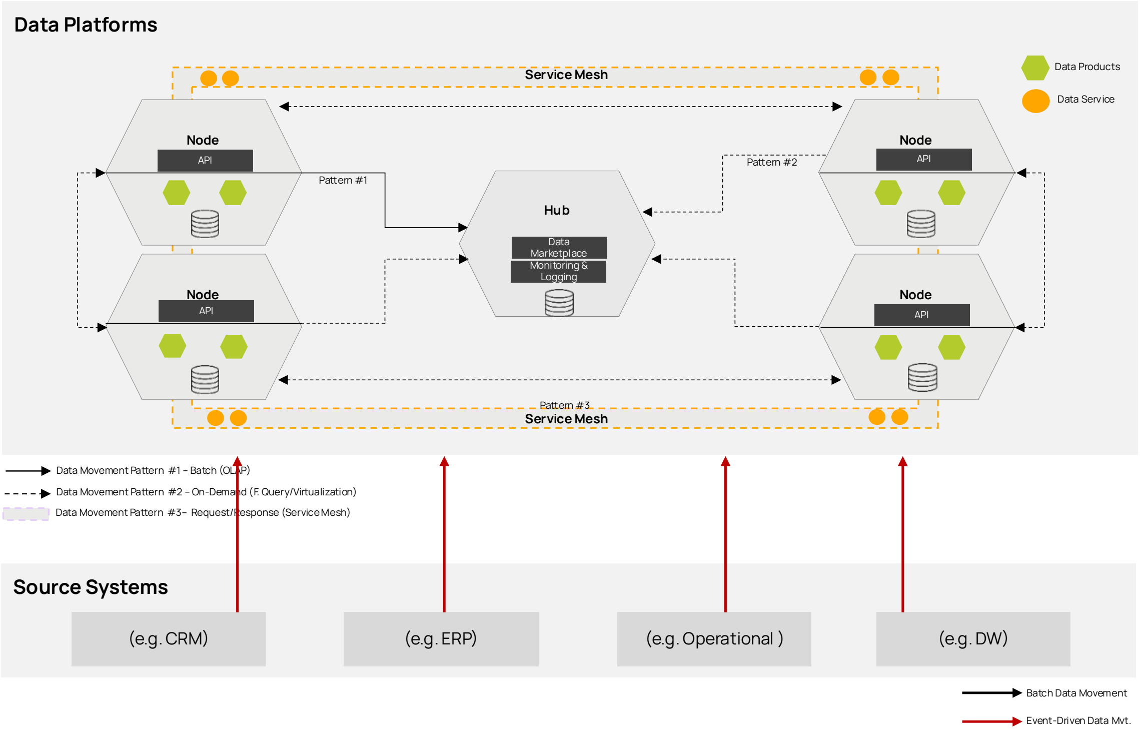


Figure 7: Architecture Overview of Data Mesh

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 stage, 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 easily co-exist with legacy systems, both departmental and company-wide (e.g., with existing data warehouse or encapsulation of unusual architectures such as mainframes).
- Compared to the data fabric 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 in 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 Catena-X platform.
- In case of frequently changing requirements of use cases (market changes), a data mesh might provide the required adaptions 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.
  - Involves many players.

  - Intend large-scale participation in multiple use cases, which requires provisioning data from many (disparate) data sources and/or data types/formats.
  - Are versatile, since the 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 central data store 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 semantic 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 the usage policies. It's easier to "delete data after 30 days" if data is only stored in one layer rather than multiple layers.
  - 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

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

5.4.1 Summary

As the name suggests, while setting up this connection, the backend system of an organization is directly connected with 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 central data governance option, it comes in handy for instances where the 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 the platform and put in use to exchange the information, for example in demand and capacity management or the CO2-Footprint.

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 to new data sets, different backend systems must be accessed and then 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.

Further, 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

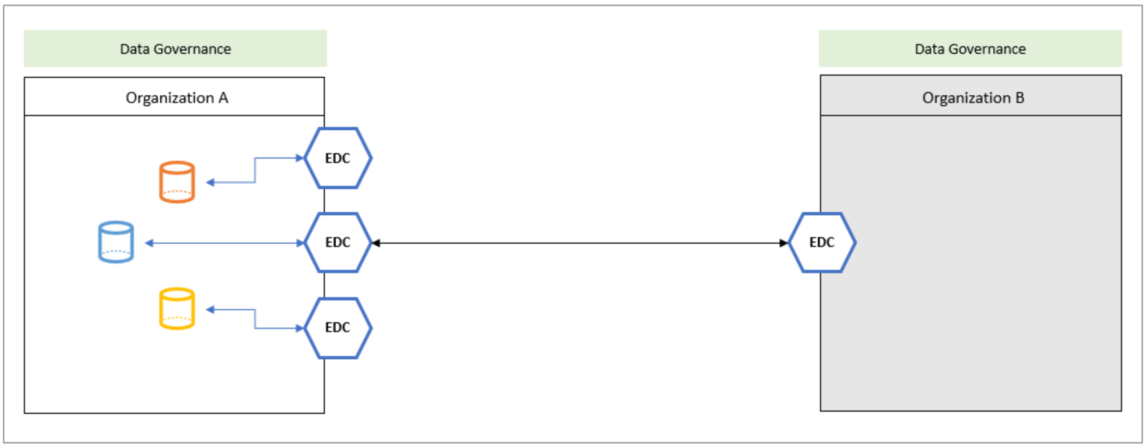


Figure 8: Architecture Overview of Direct Connection

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 the backend system directly to Catena-X requires the lowest latency, given there is just one source system is involved.

General challenges

- Data transformation may not be possible as they need to happen in source or target system, which are usually not designed for transformation. Even if they can transform data, they are usually not designed to do it for big 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 mapping between use case and source system is needed.
  - If data transformation is essential, as transformations need to happen in 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 the 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 Patterns Strengths and Weaknesses

[1] Security is easy to manage as long as the 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 (market changes), a data mesh might provide the required adaptions and data transformation 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

Finanziert von der
Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

6. PATTERNS FOR 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 data provided 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 Catana-X[4]. These semantic models are also available as open-source machine readable specifications in Eclipse Tractus-X[5].

A business application does not only need data, but a business application typically also 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 kind of 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 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 to 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. Additionally, a discovery service for digital twin registries is needed in the Catena-X data space since digital twins have different owners and in the future every data provider will have its own Digital Twin Registry.

The Asset Administration Shell[6] specifies an interoperable digital twin implementation together with standardized APIs. It will become an IEC standard[7] very soon. Catena-X is using the Asset Administration Shell standard for digital twins.

[4] https://catena-x.net/de/standard-library
[5] https://github.com/eclipse-tractusx/sldt-semantic-models
[6] https://industrialdigitaltwin.org/
[7] IEC63278

6.1 Central Digital Twin Registry

For the first releases of Catena-X a central Digital Twin Registry was set into place. However, this has many disadvantages with respect to data sovereignty and scalability. Therefore, in the future a decentralized approach will be realized.

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 some meta information. The data itself is made accessible via the EDC of the corresponding data owner.

6.2 Decentralized Digital Twin Registry

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

1. set up its 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.
2. subscribe to an operated (certified) Digital Twin Registry service

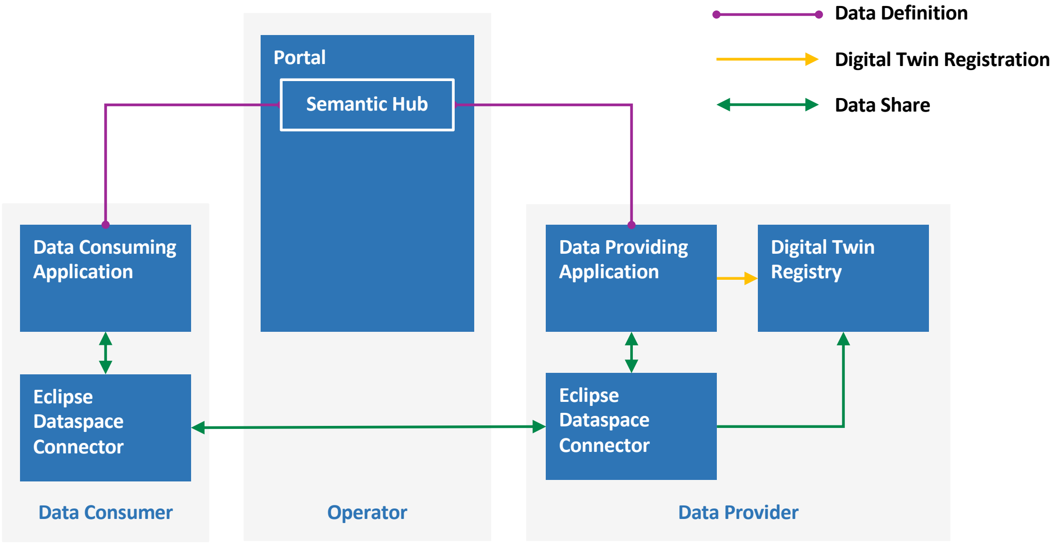

Data Definition
Digital Twin Registration
Data Share
Portal
Semantic Hub
Data Consuming Application
Eclipse Dataspace Connector
Data Consumer
Operator
Data Providing Application
Digital Twin Registry
Eclipse Dataspace Connector
Data Provider

Figure 9: Pattern for decentralized Digital Twin Registry

The API for registering of digital twins etc. does not change compared to the central Digital Twin Registry approach. The only parameter changing is the endpoint of the Digital Twin Registry that is used for registration of the digital twins.

Each data provider needs to register its 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 EDC asset in the EDC of the data provider.

The Digital Twin Registry does not contain data but just 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.

A company may also decide to run several digital twin registries. Since digital twins are organized around assets the asset type might be a criterion for splitting digital twin registries. Usage of different data sources to implement an aspect of a digital twin is not a criterion for splitting because hiding the data source is exactly the goal of the digital twin pattern. It would also be counterproductive although not impossible to split a single digital twin and register different aspects of the asset in different digital twin registries.

In a simplified pattern the recommendation is to have one Digital Twin Registry per EDC. There might be EDCs without a Digital Twin Registry, for example for mere data consumers. The aspects registered per digital twin typically are accessed via the same EDC the Digital Twin Registry is registered at. However, this is not a prerequisite as the central Digital Twin Registry approach shows.

7. PATTERNS FOR EDC OPERATIONS

The aforementioned data integration patterns can be combined with different ways of operating the external gateway – EDC – with a variety of pros and cons on its own. Here, it is noteworthy that 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 possible to implement just one central single EDC in a company which is responsible for routing all incoming and outgoing data access and transfer. However, it is also possible to set up a distributed EDC operations landscape, meaning that there are several EDCs implemented which are connected to just some data sources each. Each of these operational modes has advantages and shortcomings, however, please note that the EDC is still in an early phase of its product lifecycle, hence, some of the pros and cons might change as the product gets more mature. Further information regarding the set-up and configuration of EDCs can be found in the chapter “How to EDC” guide.

Note: Setup and EDC operations need to be separated from data pipeline implementation efforts, which might differ significantly.

7.1 When to use an EDC (and when not)

The EDC 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 app is used by a data provider or consumer, this app 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 ingest mechanisms should be used as the data-transfer happens within the own legal entity.

7.2 Single EDC Operation

7.2.1 Description

A single EDC operation is characterized by just a single EDC within a company 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 can be considered as rather low.

7.2.2 Advantages

- Setting up just one single EDC per company yields less setup and configuration efforts as it can be done rather quick (within 2 to 3 days).
- Although EDCs can be configured to stay up-to-date automatically, manual updating might be preferred by the hosting company. Thus, maintaining just one EDC is easier and quicker in terms of updating and bug fixing as just one component must be considered.

- By implementing just one single EDC less computational resources are required, although the requirements can be considered as rather low for server infrastructures in general. However, a single EDC binds less resources compared to many EDCs in a distributed manner.

7.2.3 Challenges

- As the EDC is 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.
- Maintaining and configuring contract offers for all use cases and data source on just one EDC might be challenging due to issues with ownership and responsibilities as well as respective security concerns.
- In case of EDC defects, all operations are affected, meaning that even malfunctions which are limited to only one use case can cause disturbances.

7.3 Distributed EDC Operation

7.3.1 Description

A distributed EDC operation is characterized by several EDCs within a company which are all connected to different data sources, thus reducing the number of connected systems per EDC compared to the single EDC mode. Such a structure might be beneficial if many different data sources must be accessed as corresponding configuration and maintenance effort per EDC are reduced as there are several of them available, each connected to less systems as it would be with just one single EDC.

7.3.2 Advantages

- Having specifically implemented EDCs per use case allows better access control to contract offers as these can be managed on each EDC in a use case specific manner.
- Performance and latency issues are less likely as EDCs may connected to less sources, which reduces the risk of overloading and thus ensures proper EDC operation without issues.
- In case of bugs and defects other EDCs remain unaffected, thus their operation can still be performed, and the use cases kept up.
- EDC extensions can be implemented to enable specific use case requirements without increasing the complexity of other EDCs and their respective use cases and data routing mechanisms.

7.3.3 Challenges

- Setting up several EDCs comes with additional setup costs in terms of time, effort, and resources, while these costs and efforts are even higher at maintaining several EDCs in case of necessary EDC updates or with bugfixes.

Finanziert von der Europäischen Union
NextGenerationEU

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

8. LOCAL OPERATIONS

This chapter takes a technical view on data governance and release, as well as audit trails, logging, and data quality control. The topics covered are relevant independent of the pattern that is implemented within a company. However, based on the pattern, the implementation varies or might become easier or more difficult. This chapter is not about operations of the core network components, rather it targets the decentral components that are deployed at each individual company. Furthermore, this chapter by no means tries to cover all aspects of IT operations. It will evolve over time and new aspects will be added as soon as they are shaped.

8.1 Data Governance and Release

Data governance and release is mainly an organizational topic and there is a complete separate guide on this topic on the association homepage: https://catena-x.net/en/catena-x-einfuehren-umsetzen/einfuehrung-von-catena-x. Therefore, this chapter will focus on the implications on 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 the EDC (the so-called 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 the data pipelines are implemented. The developers should work based on tickets, where also the check for data release by a board or manager could be tracked. After the 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 send 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 system landscape over time. However, it is advised to start with manual processes such as Jira tickets or E-Mails for documentation and learn about the obstacles first, before implementing automated processes.

Independent 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 be easier with a centralized tool such as a data lake or a data virtualization layer, as also 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 the usage policies, such as „the data shall only be used for the analysis of quality issues“. If the data consumed is first stored in an intermediate layer such as a data lake, the data consumer needs to track the usage policy and needs to make sure that data isn't consumed by applications or individuals that don't match the policy.

The administrative access to data pipelines, EDC & registry needs to be limited to developers so that only those can access the underlying databases, containers, and other resources. Also, it might make sense to separate EDC instances either with separate deployments or with namespaces within a cluster to guarantee segregation between two business units that aren't allowed to access data from each other. A segregation might also be necessary to allow for separate accounting of used resources to different business units. Note, that those connectors can still all operate under the same BPNL, and that the separation is only for internal process reasons.

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 EDC still needs to be implemented individually. Also, the EDC currently (and in the future) doesn't have the capability to restrict actions to certain (company specific) roles, as this is highly customized within each company. Company internal IT would need to extend the EDC with a rights and roles concept that fits the company.

Data consumer view

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

A company should separate access to the EDC from the access to the actual data. Everyone can see all contract offers, but only developers can access a 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 let the same developer that build the code also deploy it to production.

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 app is used for data consumption, the roles concept of the app should take usage policies and potential separation of data by role into account.

Example:
Company A creates a contract offer for CO2 values of breaks. It specifies that the 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 consumer. At this point in time, every employee of Company B who has access to the Company B EDC can see the contract offer and can agree to it. The purchasing department of Company B now wants to access the data 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 distribute the data to the two reporting tools. Along with the new data product, the 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.

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

In a second scenario, an employee of the quality department, who has access to the company B EDC 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. 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 could be created.

This example should show 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 focusses on requirements regarding audit trails, logs, and log retention with respect to transferred data – independent of the data integration pattern in place. You also might want to keep an audit trail of created contract offers, used policies and contract agreements. However, this is rather static data, as you e.g., only create a contract offer once and might be logged differently that operational data of constant data access. Furthermore, through UATs and IT processes, the creation of assets, policies and agreements 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 exchange data. It will be important to proof, 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 proof, 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 higher level of protection should (and in some companies 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.

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 here. Considerations on log retention policies and periods are discussed in a subsequent chapter.

(Extended) Server Logs
This type contains all the logs that software components – typically containers, APIs, data bases etc. – produce. That 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 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/ who 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, the connector and asset ID or the executed query on the data asset.

Audit Logs on assets and contracts: Those contain information about created and agreed contracts, changes on policies and assets, deleted assets or contracts etc.

Archive of Contract Offers and Agreements
The EDC has a data base of all contract offers and agreements. However, this 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 proof who had and hadn't had access to specific assets.

8.2.1 Log Extraction

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

EDC Data and Control Plane
The control plane of the EDC 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 data base. This means that it's not absolutely necessary to also log all contracts, but it should be enough to access the data base if details about created 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, ConnectorID 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 X lines of results, the exact semantic model, or other metadata.

Note: It's possible to simplify the logging model by passing information from the data/control plane to the data pipeline (or vice versa) and only log information at 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 for is fulfilled. I.e., if logs are needed to resolve customer requests in case data transfers failed, 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 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 concrete 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 contract: 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 has always been a challenge and still is one with respect to cross company data transfers. Data providers should proactively monitor the quality of data that they provide, and data consumers should check the received data for validity and conformance.

Catena-X already provides some 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.

Finanziert von der Europäischen Union
NextGenerationEU

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

9. INDUSTRY INSIGHTS

9.1 SAP Insights

SAP Integration Suite

SAP aims to support customers in their journey from enterprise-centric to network-centric business models. Therefore, SAP is working on providing integration capabilities into new federated data spaces which are seamlessly embedded into customers' existing SAP integration strategy.

SAP Integration Suite is SAP's Enterprise Integration Platform as a Service (EiPaaS) product offering enterprise-grade integration capabilities such as process integration, API-led integration, or event-based integration. It supports every form of integration, SAP-to-SAP, SAP-to-non-SAP, non-SAP-to-non-SAP, business-to-government, and business-to-business. Important to note are the non-functional qualities such as Cloud and hybrid deployment, large-scale operations, reliability, scale, security, support, and more. This makes SAP Integration Suite the perfect framework for an SAP-hosted Dataspace Connector, especially for message routing, message transformation, and usage control functionality.

Consequently, SAP intents to contribute to and make use of the open-source codebase of the Eclipse Dataspace Connector to provide a Dataspace Connector capability to SAP customers as part of SAP Integration Suite.

SAP's Strategy regarding Eclipse Dataspace Connector

Many SAP customers have articulated that they do not want a plethora of independent integration services from SAP but rather expect a coherent integration suite to provide what they need. SAP Integration Suite addresses these needs, being SAP's EiPaaS offering. It is not a monolithic service but offers several capabilities such as Cloud Integration or API Management under one umbrella. Thus, it provides a commercial and technical frame for these capabilities which can be used by customers easy and coherent but at the same time allows them to activate the capabilities they need.

Following this existing pattern, SAP plans to provide a Dataspace Connector as new capability of the SAP Integration Suite as sketched below. In addition, SAP envisions an offering as managed service for small and medium-sized enterprises (SMEs).

Finanziert von der Europäischen Union
NextGenerationEU

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

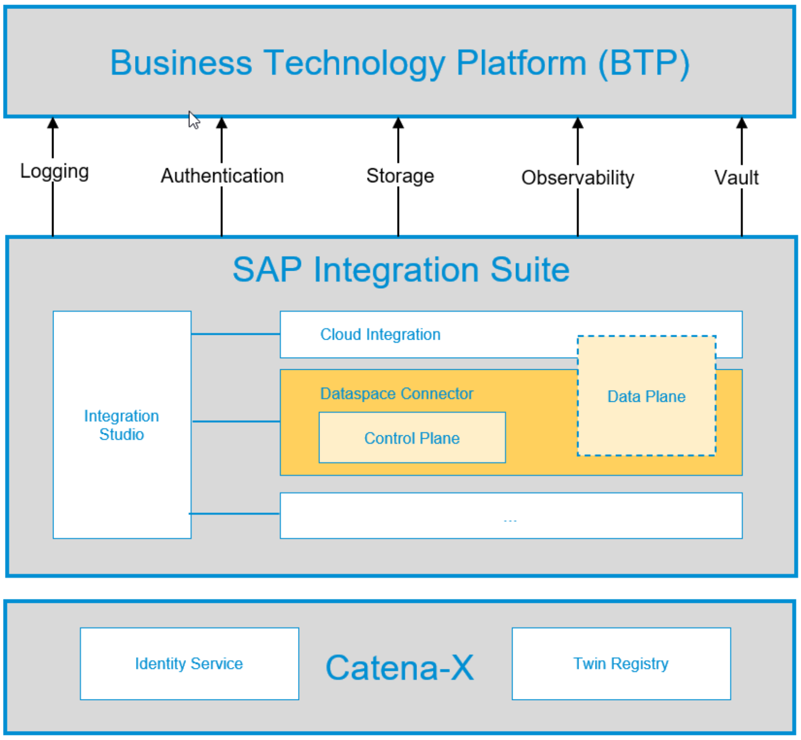


Figure 10: Dataspace Connector as capability of SAP Integration Suite

This means that Dataspace Connector is intended to become a managed EiPaaS capability of the SAP Integration Suite. It is planned to make use of services of the SAP Business Technology Platform (BTP) and to integrate with Catena-X. The plan is that it will be able to be used by SAP Integration Suite customers according to their needs with a couple of advantages:

For now, the intention for the Dataspace Connector is the following:

- It will be built on the open-source Eclipse Dataspace Connector software.
- The Dataspace Connector is planned to be fully operated and managed by SAP, and the service level agreements of SAP Integration Suite will apply.
- It will be managed in a single well-established place, the Integration Studio, like any other integration capability in SAP Integration Suite offering a guided user interface to connect to dataspaces.
- It is planned to offer strict runtime segregation for data sovereignty with built-in enterprise-grade scalability and security.
- It will reuse services (incl. existing set-up) of the SAP Business Technology Platform for backing the Dataspace Connector.
- Through the Cloud Integration capability of SAP Integration Suite, not only multiple connectivity options to backend applications and other technologies are offered by more than 200 adapters but also enhanced mediation capabilities, such as message transformation.

- The Cloud Integration capability should be directly used as a data plane of the Dataspace Connector, e.g., as HTTP data plane. This will enable a simple and efficient data exchange.
- In future, SAP can envision to plug in other data planes such as SAP Event Mesh or SAP Data Warehouse Cloud according to further needs.

9.1.1

- Since SAP Integration Suite acts as an intermediary, the following patterns are intended to be supported behind the scenes:
    - Pattern B. (Data virtualization Layer/Data fabric with direct access to backend systems) – this is directly supported by connecting the Dataspace Connector capability through the Cloud Integration capability with the respective backend applications or technologies. If the asset is not available on the backend in the correct format, Cloud Integration can transform it efficiently on the fly during the data exchange using XSLT or graphical mappings. With a rich set of mediation features the data can be further enriched or modified. To make the transformation easy for business users the whole mediation flow can be modeled in a graphical editor.
    - Pattern A. (Central Data Asset Storage as an intermediary layer) can be supported behind the scenes – this can be supported by connecting the Dataspace Connector capability through the Cloud Integration capability with the Data Asset Storage, for example in SAP Data Warehouse Cloud (SAP DWC). SAP can also envision to plug in SAP DWC or other technologies as data plane.
    - Pattern C. (Data Mesh and a decentral approach) can technically be supported similarly to Pattern A with the SAP Integration Suite acting as intermediary.

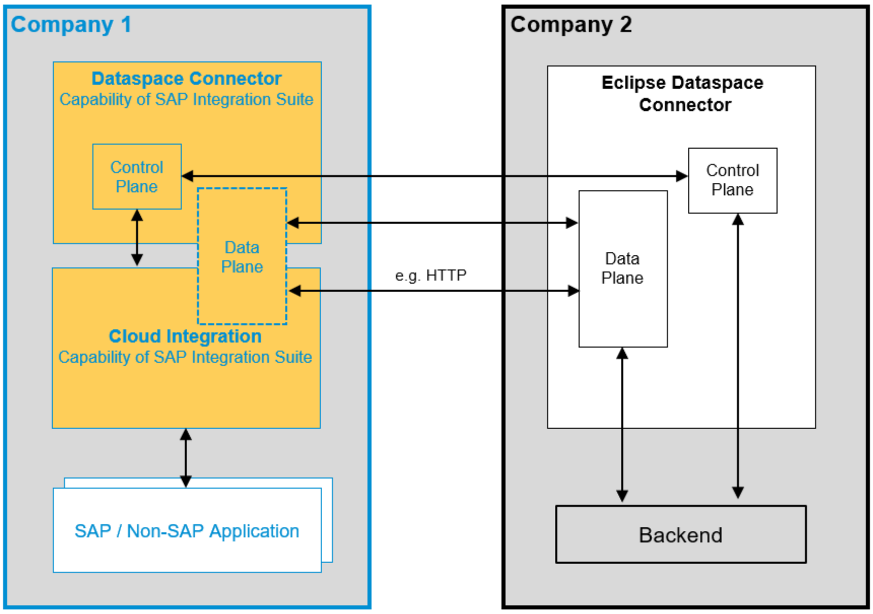

Company 1
Dataspace Connector
Capability of SAP Integration Suite
Control Plane
Data Plane
Cloud Integration
Capability of SAP Integration Suite
SAP / Non-SAP Application
e.g. HTTP
Company 2
Eclipse Dataspace Connector
Control Plane
Data Plane
Backend

Figure 11: Data Integration via Dataspace Connector in SAP Integration Suite

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

9.2 T-Systems Insights

T-Systems offering for your backend & EDC integration

A necessary precondition to participate in Catena-X is the capability of companies to provide and/or consume data. As described in previous chapters, this task requires dedicated mechanisms to extract data from backend systems, process/transform this data and supply it via EDC to Catena-X (and vice versa). T-Systems helps your company to build an end-to-end solution which comprises various integration scenarios in different shades of complexity based on your company-IT architecture and your goals in the scope of Catena-X.

Why is backend integration so difficult?

Exchanging data via data spaces beyond company scope is still a new topic for the whole automotive industry. Thus, there is not much knowledge available about possible pitfalls along the way to a full backend-integration, leading to the following typical questions:

- Data Extraction:

9.2.1 How can I set up data exchange between my IT systems and Catena-X with fast response time, minimal load on backends, small cost of operation etc.? How can I manage access rights and adhere to company security requirements? How can I provide extracted data in the correct format?...

- Data model transformation:

9.2.2 How can I derive a mapping/binding of my company semantic model to Catena-X semantics? How can I keep this mapping up to date? How can I match objects and properties across different IT-systems and different data spaces?...

- Catena-X connectivity:

9.2.3 How can I provide semantic models compliant to Catena-X standards? How can I deploy and configure EDC? How do I register data offers? How do I define policies?...

T-Systems can assist you to answer these questions via dedicated portfolio elements which are introduced in the following paragraphs.

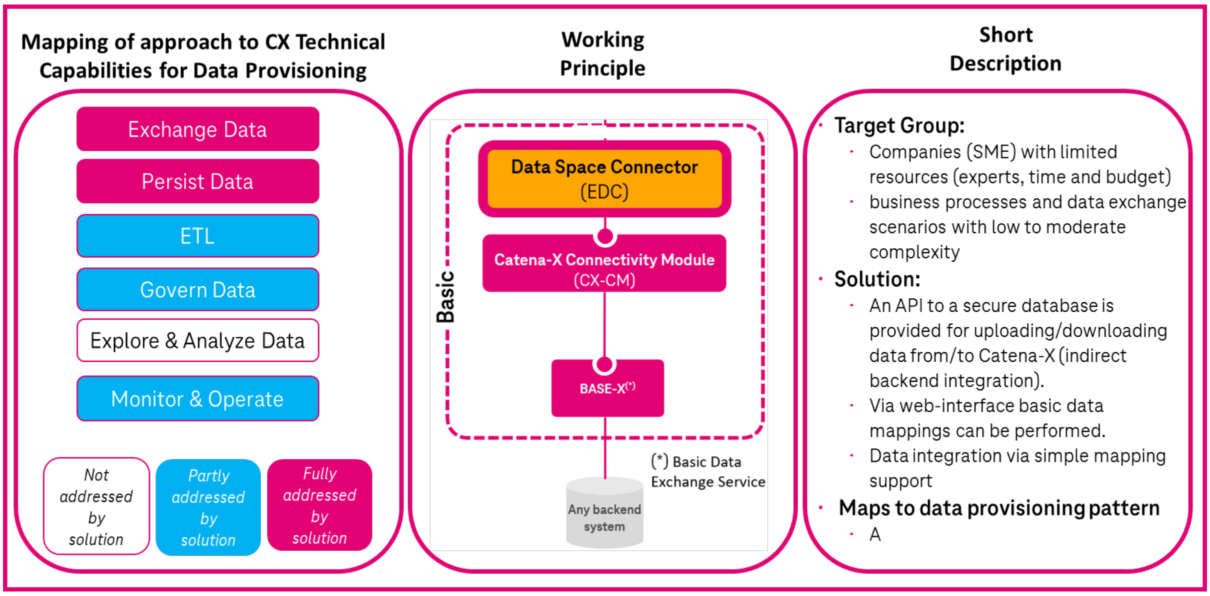
Catena-X Backend Readiness Workshop

As a starting point, companies need an understanding of the technical implications of becoming an active participant in Catena-X data space. Thus, we offer an initial workshop to provide knowledge of Catena-X architecture, its most important components, and generic options for backend integration. Furthermore, a technical deep-dive is included which aims at analysis of your individual company situation in terms of Catena-X target use cases and respective in-house data sources. The desired result of the workshop is to generate first solution ideas and pathways for Catena-X backend integration. The workshop can be held in two days but can be adjusted based on individual needs.

To implement solution ideas generated in this workshop, ready to use software modules are offered. Those modules may be combined to achieve the best value for the customer. In the following paragraphs available modules are described and bundled as packages which address different types of customers and needs for backend integration.

Basic Integration Package

The basic integration package addresses companies with rather limited resources in terms of expert knowledge, time, and budget. Furthermore, less complicated business processes and data exchange scenarios are expected. The approach consists of a Basic Data Exchange Service for Catena-X (Base-X) and a Catena-X Connectivity Module (CX-CM).


Mapping of approach to CX Technical Capabilities for Data Provisioning
Exchange Data
Persist Data
ETL
Govern Data
Explore & Analyze Data
Monitor & Operate
Not addressed by solution
Partly addressed by solution
Fully addressed by solution
Working Principle
Basic
Data Space Connector (EDC)
Catena-X Connectivity Module (CX-CM)
BASE-X(*)
(*) Basic Data Exchange Service
Any backend system
Short Description

- **Target Group:**
  - Companies (SME) with limited resources (experts, time and budget)
  - business processes and data exchange scenarios with low to moderate complexity
- **Solution:**
  - An API to a secure database is provided for uploading/downloading data from/to Catena-X (indirect backend integration).
  - Via web-interface basic data mappings can be performed.
  - Data integration via simple mapping support
- **Maps to data provisioning pattern**
  - A

Figure 12: Basic Integration Package Overview

Basic Data Exchange Service for Catena-X (Base-X)

This approach fills the gap between the Simple Data Exchanger (SDE)[8] and a full backend-integration. SDE can be used to provide data via predefined CSV files. This approach works well when companies only want to upload data occasionally. Nevertheless, if consuming data is required, or if automated processes to steer data transfer shall be established (e.g., publish a data set in Catena-X automatically when a certain lifecycle state has been reached) SDE does not provide the necessary capabilities.

Base-X provides companies an API to a secure database for uploading/downloading data from/to Catena-X. Thus, it is a solution which does not require a direct backend integration and therefore decreases complexity (less implementation and maintenance effort for interfaces). Base-X can also be a solution to connect backend systems with a small market diffusion where no off-the-shelf interfaces/connectivity modules are available (e.g., own developments, special-purpose software).

[8] T-Systems also offers the Simple Data Exchanger as a managed service in combination with automated setup of EDC (see chapter Connect Choice below)

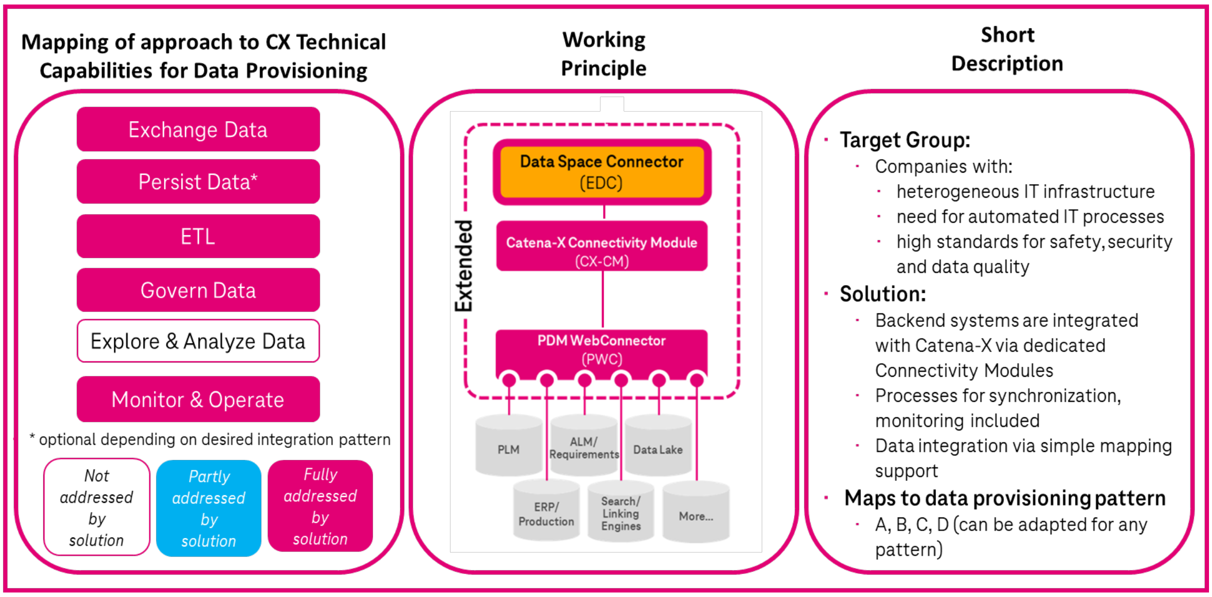
To transform the company/backend data model to Catena-X data model a simple mapping support is offered via a graphical user interface (GUI).

Catena-X Connectivity Module (CX-CM)

The CX-CM can be understood as a set of services which form the gateway for exchanging data with Catena-X. Therefore, it is an integral part of all packages that T-Systems offers. It orchestrates all components for data transfer (in particular EDC, AAS submodel server and respective backend service) and can be used to customize the dataflow in conjunction with underlying business processes (e.g., if data in a backend system is changed, a service is triggered which automatically updates respective twins in Digital Twin registry).

Extended Integration Package

The extended Integration Package addresses companies with a heterogeneous IT infrastructure (larger number of IT-Systems of different providers) and a need for automated IT processes as well as high standards for safety, security, and data quality. The approach focuses on the PDM WebConnector as one of the flagship integration products of T-Systems with a high maturity and industry diffusion.



Mapping of approach to CX Technical Capabilities for Data Provisioning

- Exchange Data
- Persist Data*
- ETL
- Govern Data
- Explore & Analyze Data
- Monitor & Operate

* optional depending on desired integration pattern

Not addressed by solution | Partly addressed by solution | Fully addressed by solution

Working Principle

Short Description

- **Target Group:**
  - Companies with:
    - heterogeneous IT infrastructure
    - need for automated IT processes
    - high standards for safety, security and data quality
- **Solution:**
  - Backend systems are integrated with Catena-X via dedicated Connectivity Modules
  - Processes for synchronization, monitoring included
  - Data integration via simple mapping support
- **Maps to data provisioning pattern**
  - A, B, C, D (can be adapted for any pattern)

Figure 13: Extended Integration Package

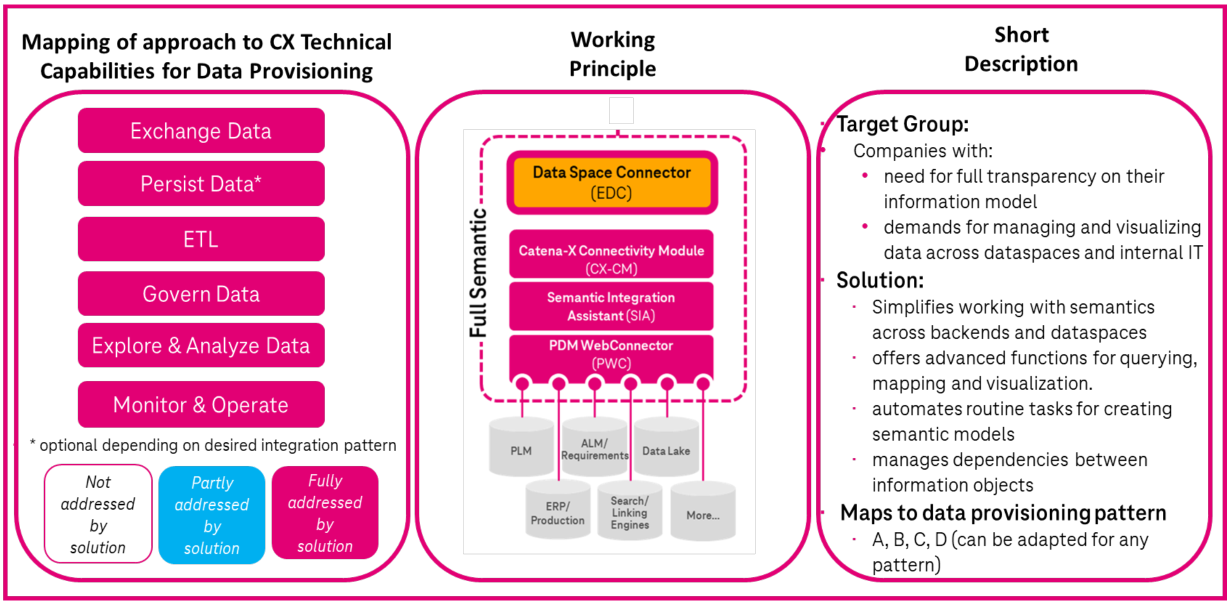
PDM WebConnector (PWC)

The product PWC is an Enterprise Middleware solution which can be used to extract data from backend systems, iot platforms etc. via dedicated Connectivity Modules. It offers a holistic concept for enterprise system integration, migration, and collaboration. It is specialized in the integration of standard as well as highly customized PLM and enterprise systems across the company value chain (e.g., engineering, production, sales). The built-in administration, monitoring and security mechanisms significantly reduce risks and overhead and increase efficiency. By using PWC you are ready for the future: cloud readiness, Big Data ingestion, IoT connectivity and much more!

- Service based integration of systems, distributed locations and collaborative partners
- Process driven data and information exchange
- Wide range of supported PLM, ERP, and other enterprise systems (SAP, Teamcenter, Windchill, 3DX, Aras, Doors, etc.)
- Powerful process engine designed for high-performance data processing
- Monitoring tool for process and system supervision
- Administration tool for easy administration of users, rights, and configurations
- Multi-PLM-Client for seamless system access

Semantic Integration Package

In addition to extended backend integration package, this bundle is directed to customers who need advanced support to align their complex enterprise information model with Catena-X, including advanced visualization, mapping and versioning of information models and model dependencies.



Mapping of approach to CX Technical Capabilities for Data Provisioning

Exchange Data

Persist Data*

ETL

Govern Data

Explore & Analyze Data

Monitor & Operate

* optional depending on desired integration pattern

Not addressed by solution | Partly addressed by solution | Fully addressed by solution

Working Principle

Short Description

Target Group:

Companies with:

- need for full transparency on their information model
- demands for managing and visualizing data across dataspaces and internal IT

Solution:

- Simplifies working with semantics across backends and dataspaces
- offers advanced functions for querying, mapping and visualization.
- automates routine tasks for creating semantic models
- manages dependencies between information objects

Maps to data provisioning pattern

- A, B, C, D (can be adapted for any pattern)

Figure 14: Semantic Integration Package

Semantic Integration Assistant (SIA)

Companies are increasingly facing the challenge of not losing overview about the multitude of information models in their IT landscape with their respective interrelations. Now with Catena-X and other data spaces this task becomes even more difficult (e.g., to manage various of AAS submodels for Catena-X in the future). To achieve this overview, Enterprise information models are derived by architects to show which objects are kept in which systems and how they interrelate. Currently, this consolidated view is created “by hand” with tremendous modelling and maintenance effort (e.g., if versions of IT-systems change).

SIA is an assistance system for digitizing management of enterprise information models. In essence, this tool allows to simply create a unified view on semantics across different backend systems and dataspaces like Catena-X. It offers advanced functions for querying, semantic mapping, and visualization. Furthermore, it automates routine tasks for creating semantic models and helps to manage dependencies between information objects across different systems in an interactive knowledge graph.

Complementary to the introduced packages an Eclipse Dataspace Connector (EDC) is necessary to participate in the Catena-X dataspace. T-Systems also offers dedicated packages to assist rollout of the EDC at your company.

Connect Starter: Flexible Eclipse Data Space Connector as a Managed Service

The Eclipse Dataspace Connector is the core component in a data space that technically enables data exchange between data provider and data consumer with end-to-end sovereign data protection and is built upon IDSA and GAIA-X standards. As the EDC is developed as an open-source project, it doesn’t provide commercial and downloadable software artifacts, that can be operated easily with customer friendly in an out-of-the-box pre-configured bundle. Furthermore, a connector only consisting of a core has barely any use. An extensionless connector can’t even communicate with other components on its own. Therefore, the connector needs value-adding extensions to incorporate better functionalities in it for broader uses and to further solve true business challenges. T-Systems is providing a IDSA certified and GAIA-X compliant, Eclipse Dataspace Connector (EDC) as a managed services for connection to Catena-X. For more information: https://dih.telekom.com/en/products

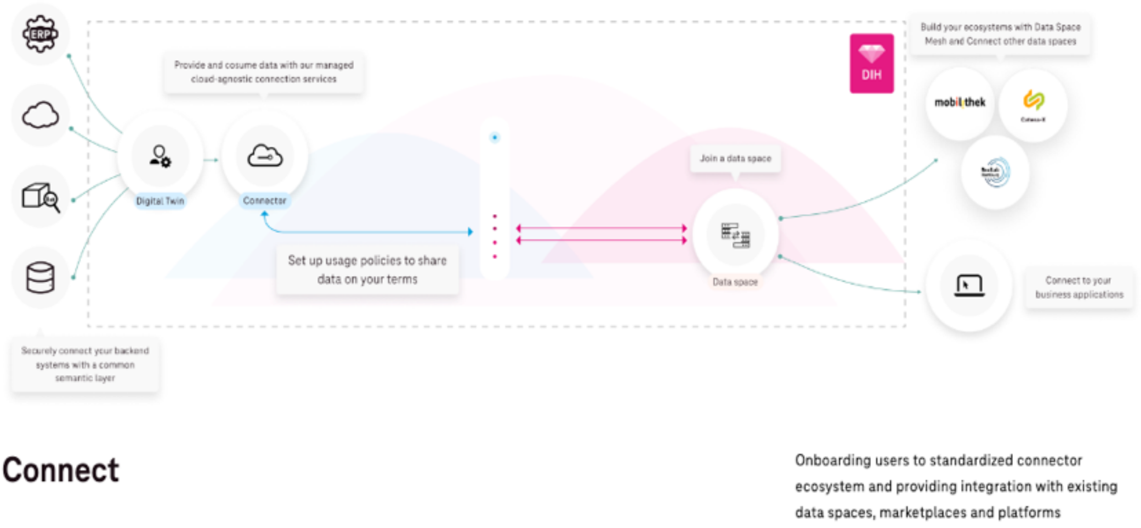
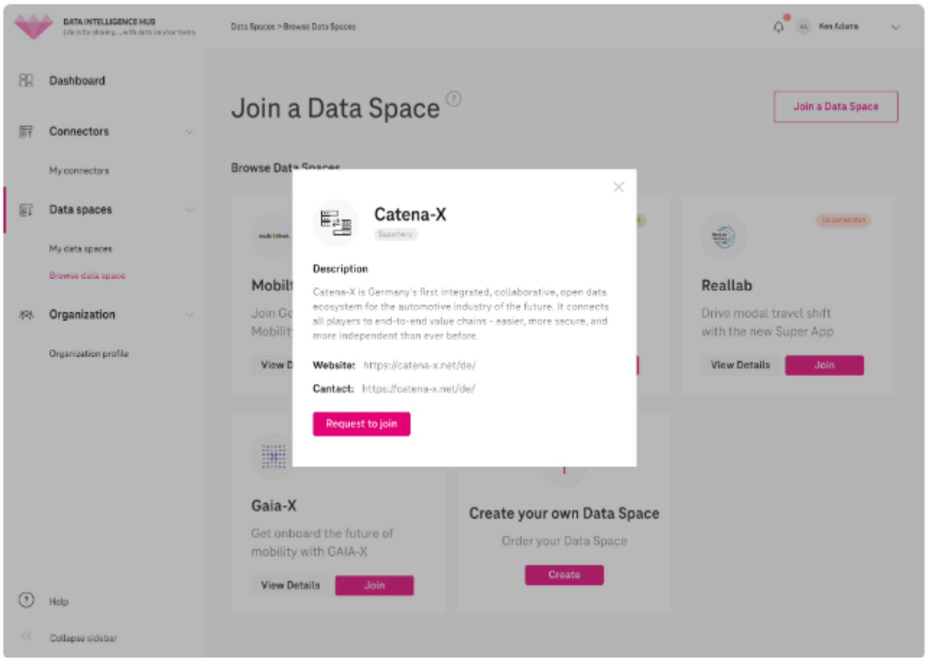

ERP
Provide and cosume data with our managed cloud-agnostic connection services
DIH
Build your ecosystems with Data Space Mesh and Connect other data spaces
mobilithek
Digital Twin
Connector
Join a data space
Set up usage policies to share data on your terms
Data space
Connect to your business applications
Securely connect your backend systems with a common semantic layer
Connect
Onboarding users to standardized connector ecosystem and providing integration with existing data spaces, marketplaces and platforms

Figure 14: Connect Starter User Flow

Connect Choice: Simple Data Exchanger (SDE) as a Managed Service

But integrating all the data sources is just not enough to participate and sustain in Catena-X. To achieve the full potential of your data and gain tangible business value quickly, a complex infrastructure is needed to engage with Catena-X ecosystem. We have taken away this pain of cumbersome, resource- and time-intensive technical onboarding onto Catena-X and have automated the process and simplified it into a few minutes. Simple Data Exchanger (SDE) Is an open-source software that allows companies to exchange data with other organizations in the Catena-X ecosystems, meaning to provide and to consume data. The application is particularly tailored to exchange Digital Twin Submodel Data in the formats that are defined & standardized by the Catena-X association. It also supports the definition of access & usage policies to assure Data Sovereignty and control over your data. For this, the key components Eclipse Data Space Connector (EDC) and Digital Twin Registry (DTR) will be used.



Catena-X is Germany's first integrated, collaborative, open data ecosystem for the automotive industry of the future. It connects all players to end-to-end value chains - easier, more secure, and more independent than ever before.

Website: https://catena-x.net/de/

Contact: https://catena-x.net/de/

Figure 15: Auto deployment of managed service

The automated process is directly integrated with T-Systems' data sharing application in Catena-X – the Simple Data Exchanger (SDE), which is accessible and triggered directly through the Catena-X marketplace. Our specially designed IDSA certified and GAIA-X compliant standard managed services solutions for Catena-X are architectured, designed and developed in a way to provide the customers with utmost flexibility and lowest of the entry barriers. With our managed services for Catena-X, we strive to make sure that it is not the technology and/or usability which stops you from getting connected to Catena-X. The bundle comes pre-packaged with EDC, SDE and DAPS in various configurations to get you connected to Catena-X and start data sharing in efficiency, in minutes and user friendly, and more importantly highly focused on keeping the requirements of automotive enterprises of all sizes in mind. For more information: https://dih.telekom.com/en/products

T-Systems as your partner for backend & EDC integration

T-Systems is the IT Market leader Automotive in D-A-CH region with 2.500 Experts dedicated to automotive business with experience of more than 15 years in the context of Backend Integration projects. Furthermore, our portfolio comprises software solutions for integration and migration of PLM/ERP/MES systems (and many more) spanning multiple vendors. These tools are used by companies worldwide including Catena-X members.

We work for Catena-X since the beginning as one of seven initial pioneers. Thus, we can offer our customers a deep knowledge about the entire Catena-X data flow from backend systems over data space connectors to the design of the semantic layer. Furthermore, we are practically involved in different Catena-X use cases and therefore do not only understand IT-side but also domain-specific requirements.

Our backend integration products cover most of the Catena-X requirements “off-the-shelf” for efficient backend integration by offering a configurable solution (GUI, low code) for optimized maintenance and operation cost. This allows our customers to adopt and scale efficiently to new data exchange scenarios, backend systems and data spaces. Furthermore, costs for technical Catena-X onboarding and operation are reduced and more plannable by decreasing software development efforts and automation of manual tasks in operation.

Finanziert von der Europäischen Union
NextGenerationEU

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

10. OUTLOOK

10.1 Data Registration

Note: This chapter is still in development and will be updated with one of the next versions of the data integration pattern guide.

This chapter shall provide a description of how data is made available to other Catena-X partners on a technical level and how the three concepts

- EDC Contract Offers
- Data Pipelines
- Internal data products/assets

can be connected and combined in a meaningful way.

The following proposal is meant to be a best-practice and might not reflect the current technical state of the various components.

10.2 Terminology

EDC Contract Offers

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

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

Data Pipeline

While the EDC 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/datasets

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.

10.3 General Considerations

- The internal datasets should be scoped and built in a way, that one dataset reflects exactly one so called "sub-model" aka. semantic model. This allows that one data pipeline only needs to connect to one dataset to extract data and the whole scheme can be used.
- One EDC 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 that easy to create a data asset "on the fly" and it is usually also not easy to maintain too 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 the datasets 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 its 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 or there's a translation service (See Example below).

10.4 An 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 dataset that contains CO2 data for all its products. Company A creates a contract offer that allows the 8 OEMs to access the 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 OEM1 and vice versa. This has the advantage, that only one dataset needs to be created and maintained. However, the customer names need to be part of the dataset so that a row-based access control can be implemented. Furthermore, the Catena-X Identifier of the consumer (BPN) needs to be mapped to the company identifiers used in the dataset.

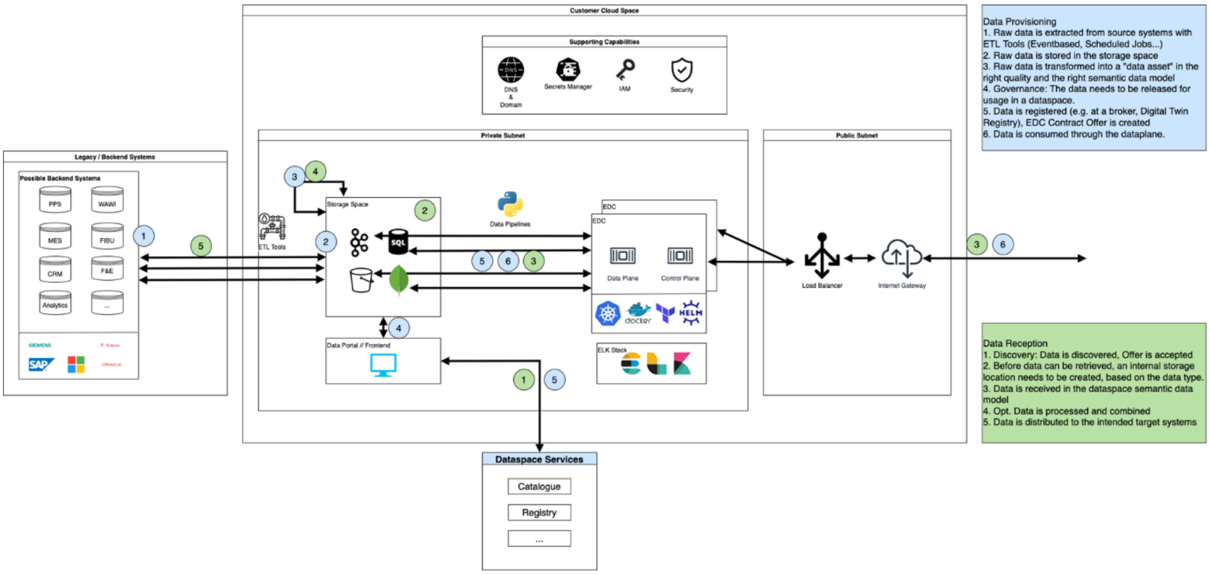
This concept is advised if there will be many distinct consumers.

Option 2:
Company A creates one dataset for each OEM containing only the CO2 information of products that are sold to each OEM. Company A then creates one contract offer specific for each OEM and one

data pipeline specific for 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.

11. APPENDIX A



Data Provisioning
1. Raw data is extracted from source systems with ETL Tools (Eventbased, Scheduled Jobs...)
2. Raw data is stored in the storage space
3. Raw data is transformed into a "data asset" in the right quality and the right semantic data model
4. Governance: The data needs to be released for usage in a dataspace.
5. Data is registered (e.g. at a broker, Digital Twin Registry), EDC Contract Offer is created
6. Data is consumed through the dataplane.

Data Reception
1. Discovery: Data is discovered, Offer is accepted
2. Before data can be retrieved, an internal storage location needs to be created, based on the data type.
3. Data is received in the dataspace semantic data model
4. Opt. Data is processed and combined
5. Data is distributed to the intended target systems

Figure 16 Data Lake Architectural Overview (detailed)

Finanziert von der Europäischen Union
NextGenerationEU

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

12. GLOSSARY

Term	Description
EDC	Eclipse Dataspace Connector: Open-source IDS connector designed to easily integrate different parties. The EDC requires a protocol implementation for policy enforcement among participants. Moreover, it implements the International Data Spaces standard (IDS) as well as relevant protocols and requirements associated with Gaia-X. However, the connector will be extensible so that alternative protocols can be supported.