FEDeRATED REFERENCE DATA SHARING ARCHITECTURE

under development

Serves as Annex 1 to

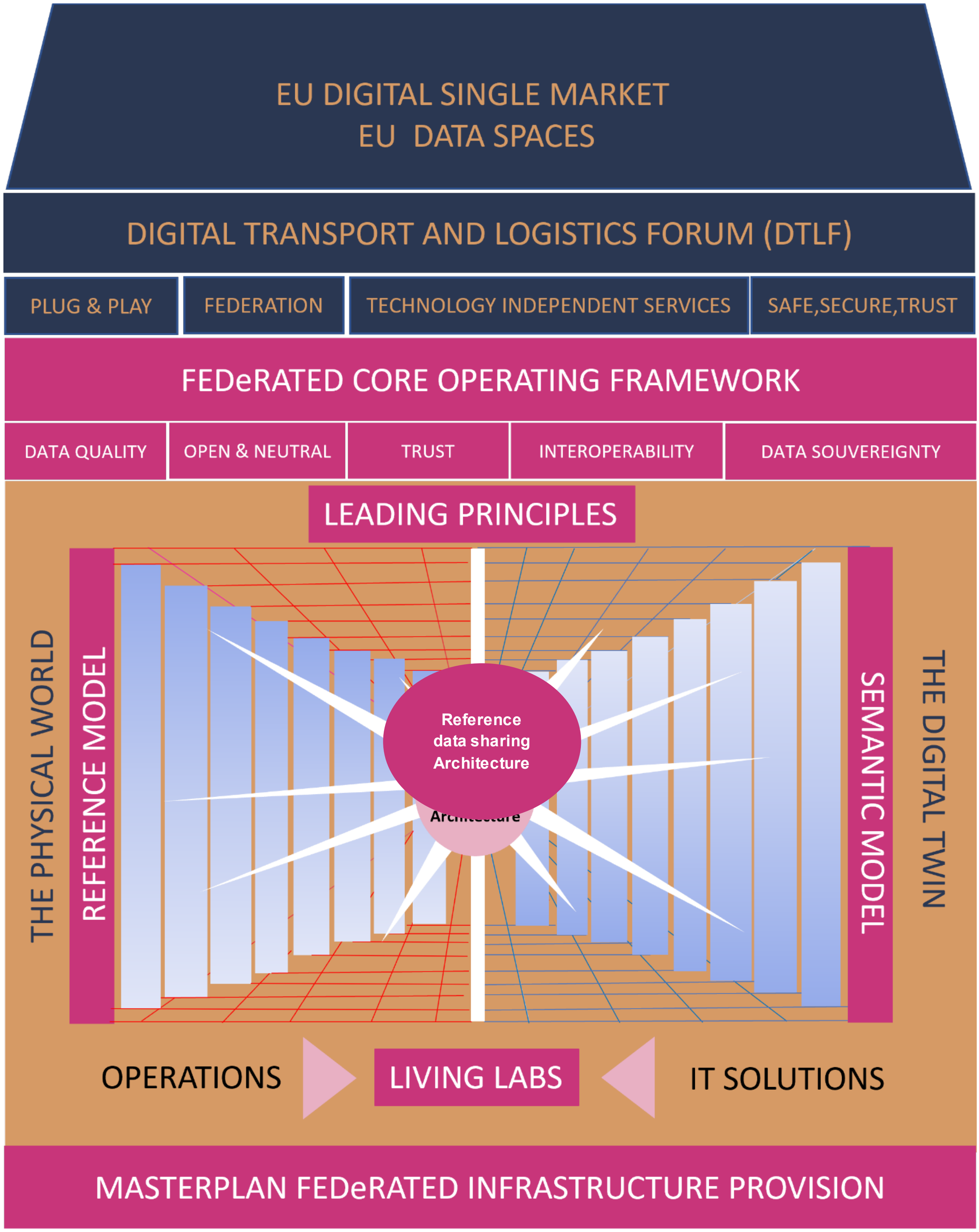
FEDeRATED MILESTONE 10

27 June 2022

www.federatedplatforms.eu

FEDeRATED
NETWORK OF PLATFORMS

Co-financed by the Connecting Europe
Facility of the European Union



This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the FEDeRATED project consortium and can in no way be taken to reflect the views of the European Union.

EXECUTIVE SUMMARY

Towards an EU data sharing grid for logistics and freight transport

The FEDeRATED Reference Architecture aims to enable the DTLF (Digital Transport and Logistics Forum) concept of a federated network of platforms to be develop as a technology grid enabling all parties in freight transport and logistics to share data according to the European Interoperability Framework (EIF). The *DTLF federative network of platforms* policy approach is based on 4 Building Blocks: 1) plug & play, 2) federation, 3) independent technology services and 4) safe, secure, trust. In practical terms this policy approach can be explained as a policy impulse to develop a future proof E*U data sharing grid for logistics and freight transport* enabling Distributed Data Resources (DDR) - i.e. IT systems/platforms that provide or use data aimed at delivering services - to connect with another. This EU *data sharing grid* would enable millions of IT systems/platforms to draw data at some times and supply data at other aimed at providing tailor made services to all participants, including compliance with legislation.

Overarching principle: Data sovereignty – data at source

The DDR embodies one of the basic principles of the EU Data Policy, the EU DTLF and the FEDeRATED Core Operating Framework (COF)[1], being data sovereignty; - more in particular data at source, pull data made available through a publish and subscribe approach for both data holders and users. This should be established in combination with the need for an open, neutral and trusted digital grid, and enabling interoperable data distribution of high quality. The consequence being that on a local or national scale, DDR's will be empowered to scale their activities onto an overarching – interoperable - EU grid. Very complex to plan for, orchestrate and keep in balance. Innovative and transitional at the least.

Top down (Reference Architecture) versus bottom up (LivingLabs)

The FEDeRATED mission is to develop a validated Masterplan and prototype to showcase how any DDR can sustainably participate in an E*U data sharing grid for logistics and freight transport* for business and compliance operations. Business reality should validate the digital technology policy vision and vice versa. To do so:

- The DTLF Building Blocks and FEDeRATED Core Operating Framework[2] were elaborated in 37 Leading Principles, 16 Technical components, and a list of Platform Services translated into this Reference data sharing Architecture.

- Various FEDeRATED partners were challenged to develop their business cases for data sharing in a LivingLab (LL) putting the Reference Architecture to the test.

[1] *FEDeRATED Milestone 2 Interim MasterPlan*

[2] *FEDeRATED Milestone 1 The Vision*

Co-financed by the Connecting Europe Facility of the European Union

The confrontation between reality and policy defines the Reference Architecture potential for upscaling.

The framework of the FEDeRATED Reference data sharing Architecture is illustrated in figure 1

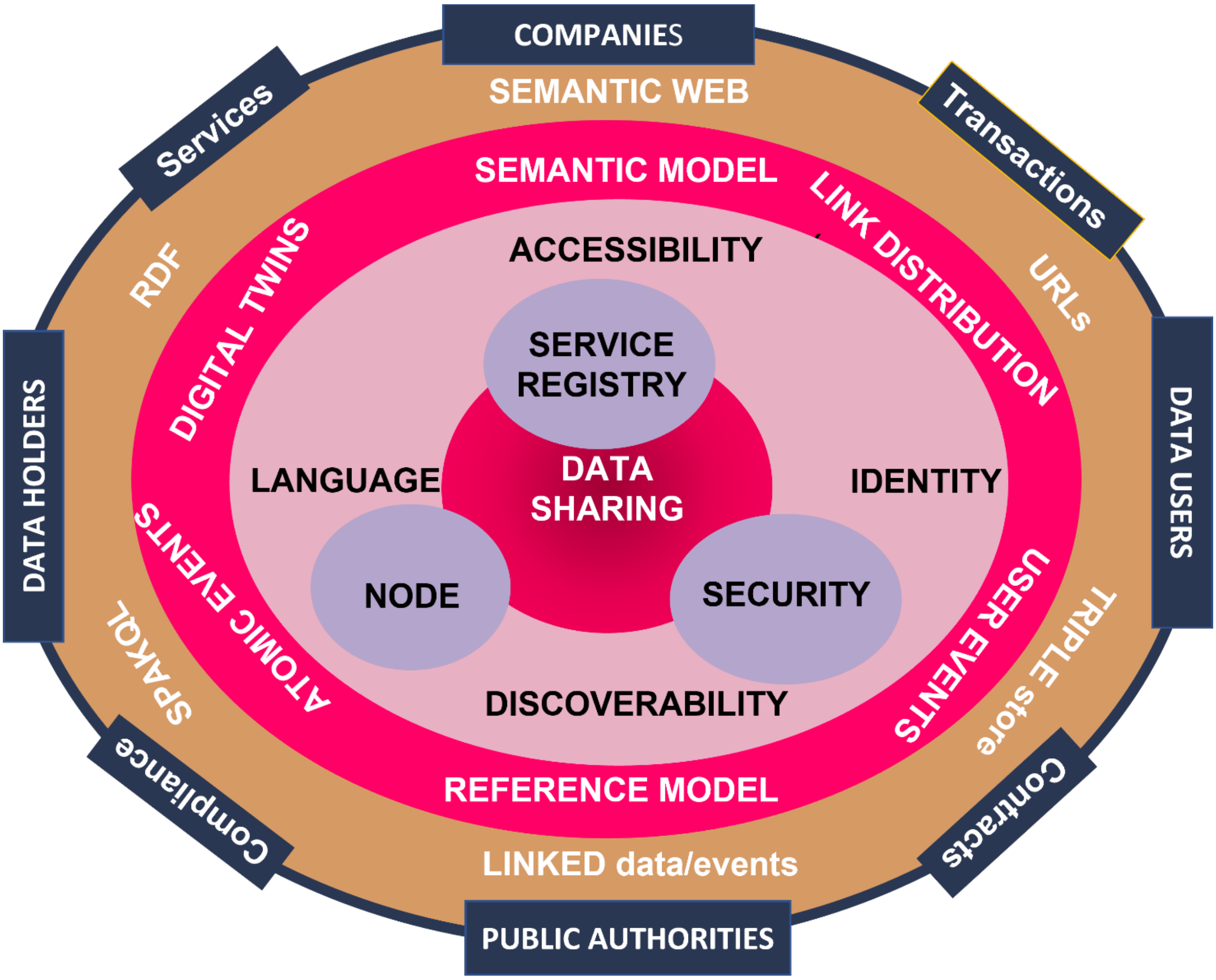


Figure 1 The framework of the FEDeRATED Reference data sharing Architecture

The Reference data sharing Architecture comprises a conceptual, functional, and technical layer.

The Conceptual layer

The conceptual layer covers the two outer circles, illustrating the need for:

- 24/7 digital connectivity between data users and data holders enabling an innumerable number of various interactions (Business cases) - Business services, Business contracts, Business transactions and fulfilment of Compliance and Authority procedures (Legislation and regulation).
- Harmonized data interoperability (Semantics/Language).

The heart of the Reference Architecture is semantic interoperability. Therefore, FEDeRATED pursues semantic innovation, supported by the adoption of semantic web technology for configuring the data sharing technology and supporting access control. The semantic web is about browsing through data via links. Each relevant data set has its unique link or URL (Uniform Resource Locator). This link makes a data set uniquely identifiable. These links enable an individual (or computer application) to browses through the data, without being aware of the server where the data is stored. When accessing the data, a computer application can interpret and process the data, since there its semantics is also specified and shared. Underlying is the idea of public accessible data, called 'linked open data'.

Applying semantic web concepts to supply and logistic chains requires additional functionality to meet requirements like semantics ('language') in multimodal logistics chains and data sovereignty. A semantic model for data sharing specifies all relevant concepts and their properties for data sharing in supply and logistic chains, called 'language'. The design principles 'Digital Twin', 'business transaction', 'event', and 'infrastructure' modularize the model and make it extendible to easily support new functionality. The semantic model requires maintenance and supports (future) innovations.

FEDeRATED pursues semantic innovation, supported by the adoption of semantic web technology for configuring the data sharing technology and supporting access control. The semantic web is about browsing through data via links. Each relevant data set has its unique link or URL (Uniform Resource Locator). This link makes a data set uniquely identifiable. These links enable an individual (or computer application) to browses through the data, without being aware of the server where the data is stored. When accessing the data, a computer application is able to interpret and process the data, since there its semantics is also specified and shared. Underlying is the idea of public accessible data, called 'linked open data'.

Applying semantic web concepts to supply and logistic chain operations requires additional functionality to meet requirements like semantics ('language') in multimodal logistics chains and data sovereignty. A semantic model for data sharing specifies all relevant concepts and their properties for data sharing in supply and logistic chain operations, called 'language'. The design principles 'Digital Twin', 'business transaction', 'event', and 'infrastructure' modularize the model and make it extendible to easily support new functionality. The semantic model thus requires maintenance and supports (future) innovations.

The Functional layer

The functional layer covers the inner circle of illustration 1 - it aims to enable technical interoperability. Logistics and freight transport are conducted in a complex network that can only be interoperable when participants – logistic operators and public authorities - have implemented agreed functionalities. The functionalities have a close correlation to the applicable semantics in connection to the various choreographies of the stakeholder participation on the grid. They encompass:

1. **Service Registry** – an IT tool to be used by all participants for the development and management of the configurations of the various methods[3] to share data. Each stakeholder, industry association, regulator, enterprise, and authority, can deploy its own Service Registry providing:
    - Metadata for discoverability through the structured sharing of user event and data set specifications amongst Service Registries. These user event and data set specifications

[3] *A Service Registry enables any user group that likes to share data through expressing the user events and data sets in the semantic model. These data sets can represent any business document, like 'electronic CMR' data sets or other specify data requirements in the context of a regulation*

can be transformed to well-known technology like **messaging**, so-called RESTfull **Application Programming Interfaces** (APIs).[4].

- An **index** enabling each enterprise to specify which data sets and links are shared as open – and linked data (data sovereignty) based on their business services, compliant with regulations. An Index stores linked data and enables a triple store to implement the complete semantic model. Each data user can formulate its queries to its local index based on the model and available links provided by one or more data holders.

2. **Security** - enables trusted and secured data sharing, implementing compliance and addressing economic sensitivity of data. A security perspective addresses cybersecurity and all relevant aspects for data sharing. Cybersecurity mechanisms need to be implemented by all participants in data sharing. The FEDeRATED Architecture focusses on data sharing, i.e. aspects like
   - **Identity** - An employee can use its **identification** provided by its employer.
   - **Authentication** of this identity - The Identity and Access Management (IAM) system of each organization deploying the FEDeRATED architecture has to be recognized and registered to allow authentication. **Authentication** of a data user by a data holder does not require the verification of certificates, nor does it require knowledge of the person acting on behalf of an organization.
   - **Authorization** for access control - It is up to each employer to organize **authorization** of its employees, using an access control mechanism of choice (e.g. role – or attribute based access control). Mutually recognized Identity Brokers need to provide an infrastructure of trusted IAM systems.
   - **Secured data sharing**. Security enables employees of an organization ('data user' according to the Data Governance Act) to access data of another organization ('data holder' according to the same act), trusting IT components based on the application of (eIDAS certified PKI-) **certificates** and applying end-to-end encryption and authentication. A data user is only able to access data of a data holder based on a link provided by that data holder ('**linked data**'). Linked data acts as an additional security mechanism. It enables a data holder to distinguish between open – (linked open data) versus controlled data access. Events are the links that are shared, they also represent the progress of logistics activities.

3. **Data sharing node.** This node acts as a type of gateway for an actor with all other actors for sharing data. Therefore, it has functionality to integrate with internal IT systems and behave according to agreed protocols with all other nodes. In this perspective, a node consists of three layers:
   - **Communication layer** – the ability to share data with (1) another node and (2) an IT system of an actor. The communication layer implements one or more options of the connectivity -, security -, and presentation protocol (see paragraph technical layer).
   - **Data layer** – the storage and validation of events and ability to formulate queries on these events (search). This layer implements the semantic model.

[4] *Any technology used to for data sharing need* ***communication protocols****. RESTfull APIs use for instance Internet protocols (http(s) also used for browsing), but also others can be applied. A variety of protocols like the Connecting European Facilities (CEF) funded FENIX connector protocols, CEF eSens Delivery, and the IDSA (International Data Space Association) connector protocols can be applied.*

- **Processing layer** – handling queries and distributing atomic events. The processing layer could be extended with value added functionality like detection of differences in data values of the same Digital Twin provided by different data holders.

The distribution of functionality to each of these layers, based on APIs, is visualized in figure 2. By providing APIs to IT systems of actors, the communication protocol and send/receive are determined. Whenever user events are received from an IT system, these are decomposed by the semantic adapter to atomic events that are stored by the data layer. They can also be offered directly to the distribution function, thus enabling sharing events with other nodes.

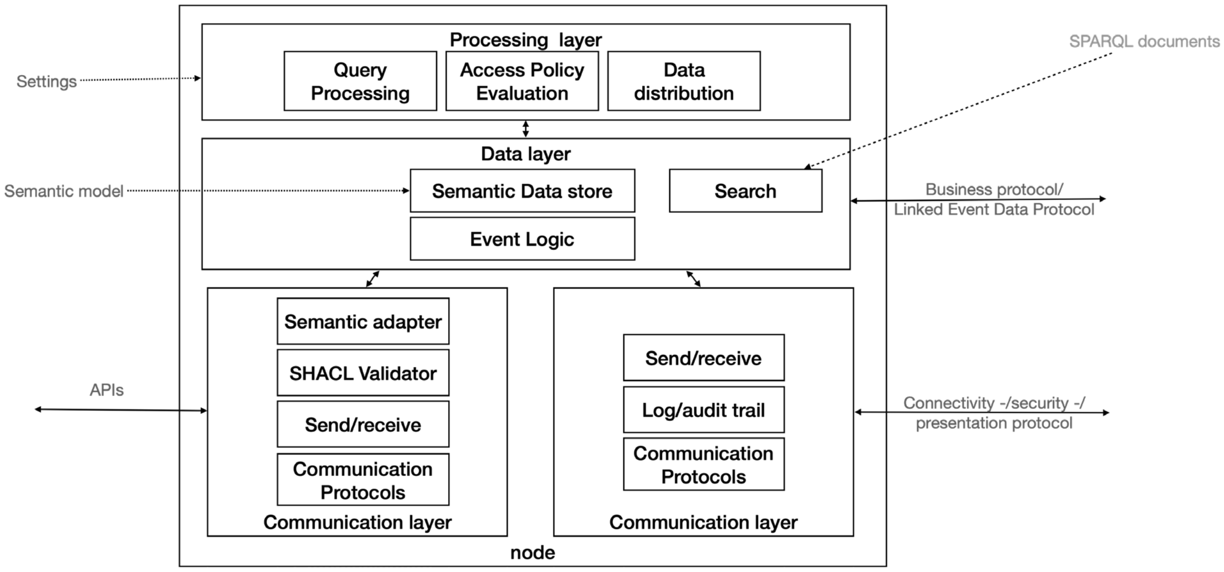


Illustration 2 The FEDeRATED node

Access control can be implemented in many ways like role and attribute based access control. The data that needs to be provided is expressed by the semantic model, for instance as a query with a validation of the response to the query. It is up to each data holder to implement these queries based on a technology of choice. The queries and required responses are provided in a machine-readible (semantic web) format that can be transformed to other formats.

The Technical layer

One of the basic principles of the FEDeRATED Reference data sharing Architecture is 'freedom of choice'. Every organization can have its own implementation of the architecture, creating a 'federated network of platforms'. This implies that the technical components of the architecture can be implemented by each organization (logistics enterprise, authority, platform service provider) enabling organizational interoperability.

The DTLF building block/principle – technology independent services - applies. However, the technology to be implement by a participating organization can only work when they adhere to an agreed set of interfaces. This results in a protocol stack, consisting of the following protocols:

- **Connectivity protocol(s)** – the technical capability for reliable data sharing between two implementations of the protocol stack.
- **Security protocol(s)** – the safe and secure sharing of data.
- **Presentation protocol(s)** – the syntax and technology (messaging, Application Programming Interfaces) used for actually sharing data.

- **Linked Event Data protocol (pull)** – sharing of links based on logistics events. This layer already uses a particular part of the semantic model. It is the interface between two instances of the Index component. [5]
- **Business protocol** – the functionality of each event in its context for sharing data between business processes. This refers to the process, it specifies a structured set of event types (business process choreography) and their minimal data requirements for providing business services. The business layer implements 'search' and pulls data; it may pose additional security requirements (like authentication of users and/or verifying their credentials). Furthermore, a business service discovery needs is required to create a level playing field.

For these interface protocols to work three issues must be further specified:

- A **minimal required subse**t of the protocols - at least contain the semantic model and options on the presentation -, security -, and connectivity layer. When specifying the Linked Event Data protocol, a function needs to be assigned to events, e.g. loading and discharge
- **Choices** to be made at various levels of the interfaces. For instance, a RESTfull Application Programming Interface (REST API) with JSON data is an example where choices have been made with respect to connectivity (http), security (https), and presentation (APIs with JSON).
- **Other general data sharing functionality**, additional to these protocol layers, to support for instance non-repudiation needs to be implemented.

The protocol stack supports multimodal transport in end-to-end supply chains for all cargo types. Individual enterprises will not always provide or require the complete functionality. They need to configure the protocol stack for their capabilities and goals[6]. This configuration will be based on the business process to conduct and the underlying choreography. The configuration will lead to a set of agreements enabling data sharing to materialize.

The Reference data sharing Architecture does not cover specific legal interoperability issues. It is up to every participating organization within the grid how to deal with the relevant legal frameworks, policies, and strategies.

[5] The consequence of implementing the linked event data layer (data pull) is that data remains at the source. It implies that any upstream organization that received a link to that data via any intermediate(s) is able to access this source data. Another implication is that particular data can change ownership as responsibilities in supply and logistics chains and ownership of goods are transferred. This type of functionality is (most probable) currently not supported in IT systems used by organizations. Deployment of this functionality can be via a 'node'.

[6] This is called 'plug and play'. An intermediate step towards plug and play, which is considered the final objective, is to offer predefined settings to these enterprises (and authorities). Examples of predefined settings are particular document data sets that can be retrieved via a pull (eB/L, eCMR, eAWB) and visibility events (ETA of a transport means, load/discharge events, etc.).

TABLE OF CONTENTS

Table of figures

1 INTRODUCTION

1.1 *Objective*

This Reference Architecture document should be considered as the FEDeRATED IT architectural knowledge base. It is geared to assist:

- The EU Digital Transport and Logistics Forum (DTLF) to develop a sustainable data sharing architecture for a network of platforms
- FEDeRATED LivingLabs project managers developing and executing federative data sharing business cases between data users and data holders.

The document identifies the components for data sharing in supply and logistic chains from a business and functional perspective and the interfaces between the various components specified from a business, functional, and technical perspective. The technical perspective of the components, which aims to enable different stakeholders to implement the architecture using a technology of choice, is outside scope of this document.

As digital technology evolves, this Reference data sharing Architecture document should be considered as work in progress. still lacks sufficient details on some parts where other parts are still empty. This will be finalized by the Final Master Plan. Furthermore, it will possibly also result in a separate protocol stack

The current version of this Reference Architecture document was agreed by the FEDeRATED IT Architecture Board on 27 June 2022. It aims to serve as an Annex 1 to the FEDeRATED Milestone 10 report, due 31 October 2022. This Milestone 10 report will sketch in non-technical terms the first draft of the FEDeRATED Master Plan in connection to the FEDeRATED LivingLabs. This document.

The knowledge gathered and elaborated in this document is based on 3 years of intense discussions and knowledge gathering within the FEDeRATED project, especially the FEDeRATED IT Architecture Board and Semantic Modelling Group. This was also done in collaboration with the DTLF, subgroup 2 and in connection to the EU CEF FENIX project.

1.2 *Background and relevant input*

FEDeRATED has developed a Vision (Milestone 1) and an Interim MasterPlan (Milestone 2) to elaborate how the so-called federated network of platforms as identified by the DTLF can materialize. This network approach covers three perspectives:

- Business perspective – language and process. In brief: the condition being that data that are understandable for third parties are eligible to be shared for multiple purposes)
- Functional perspective - discoverability/searchability, access, Identification, Authentication, and Authorization (IAA). In brief: browsing the network of platforms requires a trustworthy environment where every party is able to move freely, find a suitable partner and be found by others, make your preference known and shield away when you want to
- Technical perspective - basic functionality for data sharing (log, audit trail, security, etc.). In brief: the technology toolkit enabling you to browse the network in an accountable way.

Co-financed by the Connecting Europe Facility of the European Union

These perspectives are more detailed in this document and can be illustrated as follow.

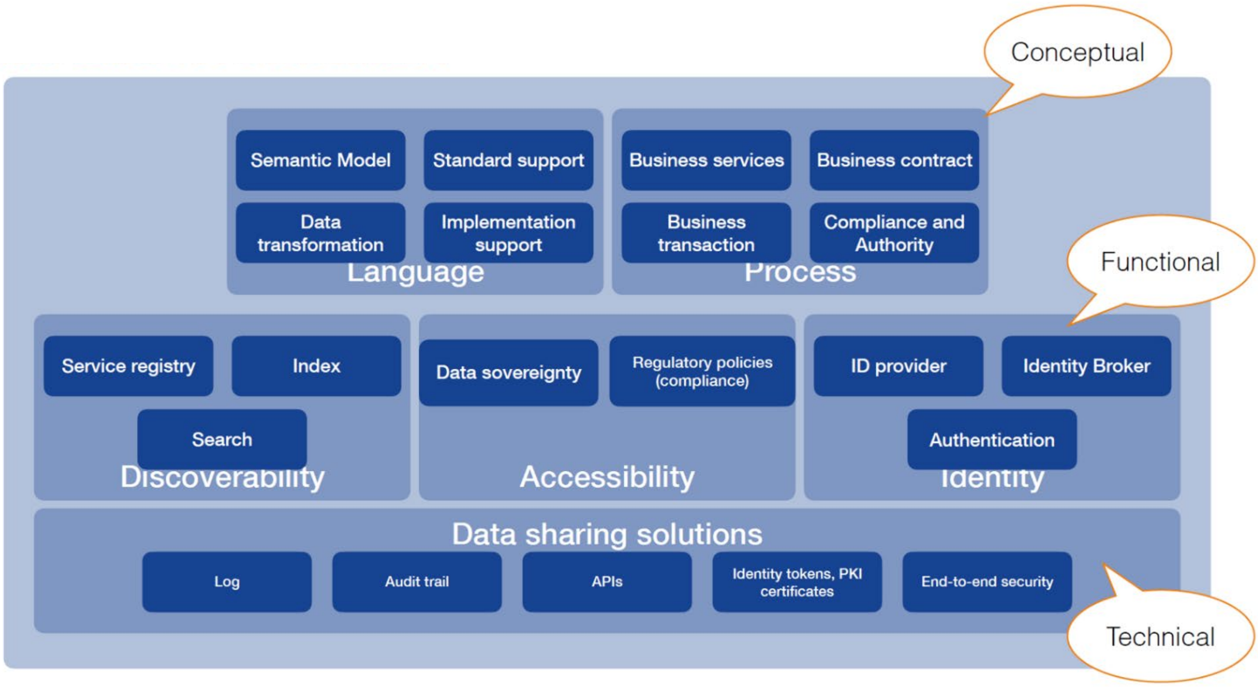


Illustration 3 The overarching federated network of platforms architecture

In this Reference Architecture document, the following has been taken in consideration:

- All relevant output produced by the various DTLF Subgroups is used.
- All business processes - Business-to-business (B2B), business-to-government (B2G), government-to-business (G2B), and government-to-government (G2G)
- All transport modes, cargo types, and relevant logistics activities in supply and logistic chains. This does not necessarily imply that FEDeRATED will develop all relevant aspects. For instance, part of the language can be developed by another stakeholder and linked to the FEDeRATED specification of the language.

Relevant international treaties, EU acts and regulations and national legislation effecting data sharing and (cyber-)security are applicable to the architecture presented, i.e. the Hague-Visby Rules, the EU Data Governance Act, and the Cyber-Security Act.

1.3 *From architecture to interfaces*

The FEDeRATED Architecture aims to enable the free flow of data between all relevant stakeholders in supply and logistic chains to support the physical flow of goods. The architecture is based on the core DTLF and FEDeRATED principles. These principles set the proposed framework – Reference Architecture - for the development of the advocated digital technology. The requirements are:

- All parties – organisations (IT systems/platforms) involved in supply and logistic chains should be able to participate and profit from it.
- Participating organisations are accountable for their actions. When interfacing, organisations should adhere to a protocol stack:
  - o Connectivity protocol

    - Security protocol
    - Presentation protocol
    - Linked Event data protocol
    - Business protocol
- The proposed Reference Architecture will be constantly updated (change management)
- A governance perspective will identify what aspects of the Reference Architecture should be developed and maintained on an EU/international scale and national scale, including the accompanying certification procedures.

The participating organisations are free to implement the Reference Architecture under the condition they comply with the above requirements.

1.4 *Intended audience*

The further development of this reference Architecture will require interaction and commitment of:

- Business analyst – to specify interfaces and interaction patterns in supply and logistic chains applying the 'language' and processing rules, and to analyze their impact on business processes.
- IT architects – to analyze the impact of the protocol stack on existing IT infrastructures and system of an organization.
- IT developers – to develop and deploy software supporting the relevant parts of the protocol stack for an organization.

1.5 *Structure of this document*

This document is structured as follows

- Chapter 2 – the core concept
- Chapter 3 – elements and perspectives
- Chapter 4 – language as the core of the solution
- Chapter 5 – management perspective, collaboration for a distributed solution
- Chapter 6 – safe and secure data sharing
- Chapter 7 – data sharing between the various stakeholders
- Chapter 8 – the proposed architecture in brief
-

2 THE CORE CONCEPT AND ITS APPLICATION

This section illustrates the objective that is to be reached by implementing the FEDeRATED architecture. A reference – serving as an illustration of the core concept - is made the ‘intelligent’ box[7] The example provides input to requirements to ‘language’ (see next sections).

2.1 The ‘intelligent’ box

The core concept is illustrated by a box delivered to a warehouse or depot. This box has several identifiers such as shipper and carrier numbers. These identifiers have little meaning to the warehouse operator. Instead, what if we gave the pallet a universal identifier: one that can be accessed simple, as a two-dimensional barcode on the box that links to additional information such as its content and the action to be performed by the warehouse operator: i.e. unpack and store the content in the warehouse. It just so happens that the W3C (World Wide Web consortium) has defined such a system: the URL (Uniform Resource Locator) or web address linking to data on a webserver, as shown in Figure 1.

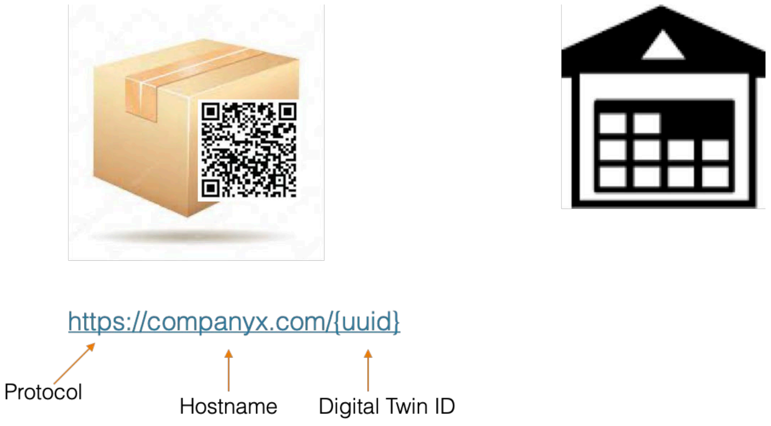


Figure 2-1 uniform identifiers (URIs) as links to data

The previous examples shows that the URL of the box consists of a hostname and a UUID, a Universal Unique Identifier. A UUID is a means to generate global unique identifiers, applied in many (modern) IT solutions.

The warehouse operator scans the barcode of the URL to retrieve and update information such as goods receipt at warehouse, storage location in the warehouse, or similar.

Another example concerns trucks involved in road accidents. Emergency services such as the police and fire brigade, as well as many other entities, need to be informed of the nature of the cargo onboard the truck. Is the cargo dangerous, is it livestock, or of any other nature that will influence the work of these emergency services? Where can these emergency services find data pertaining to the truck? The only available data at the time of an accident might be the license plate of the truck and trailer and this can be linked to a URI of the truck; this then enables a link to eCMR (electronic version of consignment information) data, for example, which will provide access to the necessary information.

In all examples it is all about applying the FAIR principles: Findable, Accessible, Interpretable, and Re-usable. Applying URLs as identifiers for accessing data, common security aspects addressing

[7] *is also used in the documentation produced by the DTLF (DTLF Interim report, dtlf.eu).*

for instance identities, and common language and interaction set for business process collaboration are required. Data remains at its source, only links are shared amongst the various entities. All examples have two roles in common as defined by the Data Governance Act, namely a data holder and data user. These issues will all be addressed by the architecture.

2.2 Data browsing – towards innovative applications

Think of all types of goods, assets, and locations having a unique link to additional data. For instance, a terminal can have unique link to its opening hours, geo-coordinates, gates and any other data that terminal operator wants to disclose. Quays, roads, all types of assets, and packaged products can have such a link. Only bulk cargo like sand or grain will not have a barcode with the link.

Browsing these links provides data of those physical objects and locations. Browsing will become stronger if also links between these objects and locations are available. We enter the area of linked event data where events have a meaning in supply and logistics chains: the ETA (Estimated Time of Arrival) of a truck at a terminal, a vessel ETA in a port of call or alongside quay, and a container loaded on a vessel. Events construct these links: they can have a link of a container and one of a vessel. Events need to include 'time': when are these links constructed, i.e. at which time is a container loaded.

Business documents will also have a unique identifier as a link to its electronic data. Physical objects like goods, containers, and transport means are 'linked' to these documents via (administrative) events.

Thus, having these events makes it possible to browse through all types of data. One can simply access a vessel - or container track or access all business documents in which a container has been linked.

2.3 Distributed data management

Data browsing of links and events gives opportunities but can also bring threats. Using a smart phone, everyone can read barcodes and directly link to a website with additional data and start browsing.

In supply and logistic chain, open access to data is an issue. Data transparency increases vulnerability and liability, resulting in theft, and thus an increase of logistics costs. Access to event data will reveal the content of a container. Data is also commercial sensitive; it shows trade relations, prices, conditions, etc. Even the 'hostname' in the previous example already reveals information. On the other hand, authorities also require data browsing to access the content of a container for instance for customs risk assessment or emergency handling in case the container is involved in a road accident.

Thus, controlled access to links and events combining links is required. It implies that any link on a physical object is just an identification without meaning. The identification can refer directly to source data, which would imply increase vulnerability to unauthorized data access. By combining a meaningless link with additional data that has been shared via events, like a host name of the one that has provided the link, to become meaningful. There can be a business rule like composing a (meaningful) URL is only feasible if one has received the meaningful - and meaningless identifier in advance.

There are cases where the meaningful identifier, i.e. a host name, is published as a web address on for instance a truck. It means that everyone (with the proper credentials) that is able to combine both identifiers can access the data, unless the public web address differs from a data address (i.e. the so-called endpoint for a data query). The architecture will provide rules for handling these.

In supply and logistic chains, parts of orders can be subcontracted. For instance, these meaningful – and meaningless identifiers of a customer are thus shared by a service provider to its subcontractor. That customer must be sure that only an authorized subcontractor requires access to that link.

2.4 Data semantics and – quality – Digital Twins

When browsing through data, the meaning of that data needs to be clear for processing by IT systems and solutions. Like said, each physical object and location will have a link to its data. A digital representation of those physical objects, locations, etc. is called 'Digital Twin'.

A Digital Twin representing a part of the real-world needs to be specified by its properties. What is the data representation of a Digital Twin representing equipment like a container, i.e. which data properties describe a container and which can be derived? Container number, container size and type, etc. are properties of a container; an indicator of an empty container can be derived from the fact there is not a link to goods or there is a difference between container tare weight and (verified) gross mass.

A Digital Twin also has restrictions as to its association (link) to another Digital Twin. For instance, a container can only be used to carry pallets and boxes that fit, implying it cannot be used to carry another container or goods that are too large to fit. Ferries can be used to transport containers, but only if they are on a trailer. These types of rules reflect the real-world and thus also need to be reflected by the digital world.

To be able to process the data, its quality needs to be specified. Data quality is amongst others about 'correctness' and 'completeness' of data: a minimum event data set required to perform logistics activities, data formats, and association rules between Digital Twins. In terms of data correctness, a container number has for instance a prescribed structure and there are code values for container size and type. These restrictions can be validated before sharing and/or at reception of the data. Any deviations from (a minimal set of) these restrictions have impact on decisions made with this data.

2.5 Data requirements of users – flexibility and extendibility

It will not be possible to have a single organization specifying these data requirements. There will be modality or sector specific data requirements, like for the chemical or automotive sector. Modalities and sectors can re-use common agreed semantics and extend it for their specific needs. These needs may already be known or will have to support any future applications.

Whenever (communities of) users formulate their specific data requirements, they can publish them and make them available to a larger community. They can become part of the common data requirements that can be used by all users. The latter requires a governance structure.

2.6 Data requirements of a single user

All users, like authorities and enterprises, differ from each other. It implies that each user will have different data requirements to support its business processes. However, many data requirements can also be common to user groups or are formulated by regulations. For instance, particular business document data sets like eCMR are common for a large group of users. Another example is a container track based on browsing the data of various sources.

Common data sharing requirements relate to initiating and completing business transactions, regulations, predictability of supply and logistic chains, and any data sets that reflect liability and responsibility (business document data sets in the current implementation). These requirements are formulated as follows:

- **Business transactions** – this includes digitization of all relevant individual process steps from finding and matching logistics services to their execution (visibility) and payment. There are various ways to implement these processes; best known are framework contracts and shipment-based booking and ordering.
- **Document data sets** – this is about digitization of existing business documents like electronic CMR, - Bill of Lading (eB/L), and – Airway Bill (eAWB). Data can be made available according to the architecture. The structure of these document data sets needs to be expressed by (modality specific) data requirements. Aspects like data integrity are of relevance for these types of data sets, since they reflect responsibilities and liabilities of service providers.
- **Visibility data (events)** – these represent the past, present, and foreseen progress of transport orders. Since there are also various ways events are perceived, implying we need some standardization that is adaptable to user requirements. They reflect predictability of supply and logistic chains.
- **Compliance to regulations** – data will have to be available to an authority, based on a (national implementation of a) EU Regulation. To optimize the implementation, the compliance data requirements must be expressed in the common data requirements, maybe with some extensions specific to a regulation. Please note that compliance can be based on digitization of document data sets, where an authority must be able to accept digitized document data (e.g. eFTI) or a logistics stakeholder has to provide data access digital.

Not all combinations realize interoperability of all business processes between any two stakeholders. For instance, if a user only implements sharing document data sets, it will not be fully interoperable with another one that implemented quotation and ordering as part of logistics services.

Any specific data requirements of a user must be expressed in the common ones, including any industry and compliance specific data requirements, and relate to data sharing capabilities of potential data holders. For instance, a user can never receive container data of a data holder that only transports bulk cargo.

Thus, data sharing capabilities and logistics services need to be known to fully make use of the capabilities provided by FEDeRATED.

2.7 Itinerary at the core of digitization

The first focus of this document is on transport (and transshipment) services, although other services can be included. Itineraries of transport means are at the core of all transport processes. Estimates and actuals of arrival, loading, discharging, etc. are at the core of all processes. They describe what is planned to take place and takes place. Itinerary data allows all relevant stakeholders to prepare and synchronize processes. It allows authorities to assess any risks that contributes to improvement of seamless flows. It enables safer handling of incidents, etc.

2.7.1 The concept 'itinerary'

An 'itinerary' of a transport means can be from the start of the lifecycle to its end or can be just a movement between two locations. Itinerary is defined by its user or modality. For instance, a flight is an itinerary from one airport to another, potentially via several intermediate stops, where one or a succession of airplanes pick up and drop off freight at any of the airports along the route towards is final destination. In this example, a flight may be scheduled at regular basis or be chartered. A voyage of a vessel on the other hand is from one direction to another upon which various ports are called, complemented by its return voyage where the same ports might be called.

Similar definitions can be given for other modalities like road ('trip') and rail ('path') transport. Basically, an itinerary of a transport means is defined as:

- Itinerary is a timed sequence of places of call. Synonyms for itinerary are voyage (vessel), trip (truck), and flight (air transport). In current IT applications these will have an identification, e.g. a trip identification or flight number. Such an identification is not required when applying semantic technology but might be required for a search by a user.
- Place of call of a transport means is a location where goods or any type of equipment are loaded and/or discharged (passengers can also be involved). A place of call can be a terminal in a port, an airport, an inland hub, a distribution center, a retail store, etc.
- Route. A 'route' is defined as the use of an infrastructure (road, rail, (inland) waterways, air) taken by a transport means between any two places of call in its itinerary.

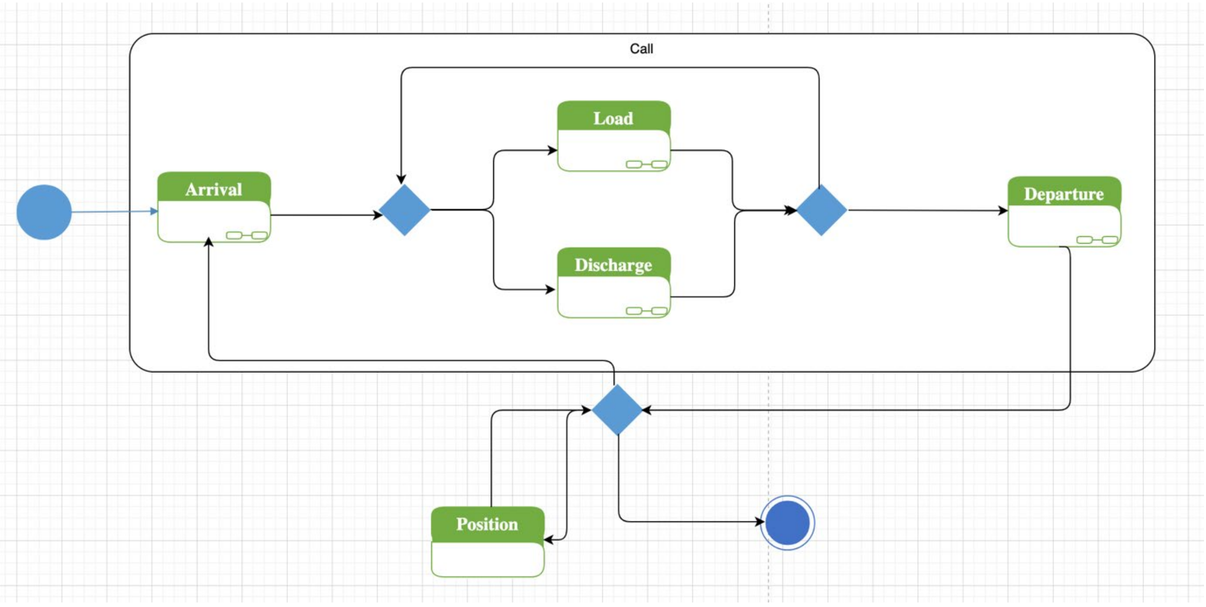


Figure 2-2: state diagram of an itinerary (trip)

Figure 2-2 shows that an itinerary starts with an arrival of a transport means at a call, followed by load and/or discharge activities in an order relevant to the transport means. After completion of load/discharge, the transport means departs to a next call, a relevant position or, when empty, to its home base (or some comparable location). Each state like arrival, load, discharge, departure, and position ('via-point') is represented by an **atomic event** (section 4). A load event, for instance, associates a goods to a truck at a specific time.

2.7.2 Data sharing relations

Supply and logistic chains is generally considered to be a complex system. It consists of many, autonomous operating organizations and, in future, autonomous assets like self-driving trucks and vessels. The behavior of such a complex system is determined by all participants. When they are willing and able to share the proper state data known to them, the (parts of the) system can be optimized. This requires common rules for data sharing.

Its basic operation can be visualized as, where 'itinerary' is at the core of the physical processes, where a physical operator can be a subcontractor or an employee of a service provider (like a truck driver).

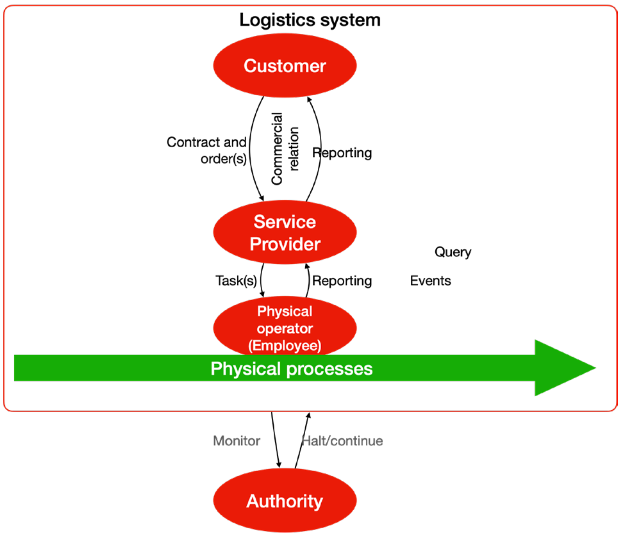


Figure 2-3: basic operation

In this simple approach, authorities monitor the progress of physical flows and can inspect (halt) or allow their continuation. The logistics stakeholders act and share data in the context of commercial relations. Commercially it is all about business service catalogues, timetables, bookings/framework contracts, orders, and reporting of the progress, also known as 'visibility'.

Bookings and orders are generated by a customer to its service provider. They can be updated by a customer, based on events received from other service providers, for instance to indicate a delay. These updates are only distributed within certain limits like time windows. In case a decision is required for reorganization of a chain, this is outside scope. It requires what is called a **control tower**.

The figure shows that a customer will receive a report from their service provider, where this report is generated by one of its employees (or an automated system). This could be a truck driver reporting on their load and discharge activities in its task description, where the task description represents the itinerary (trip) of a truck. For warehousing, the task description of a physical operator could be to unpack and store goods and report on where the goods are stored. The same approach is applicable for a terminal operator.

2.7.3 Common rules for distribution of data – state information

Where currently, distribution of data is quite a labor-intensive process of handling updates and providing different output to different customers, service providers, and authorities, much of the data can be distributed automatically once commercial relations and expected/planned itineraries are known. Common rules can be formulated, resulting in IT to support these rules.

These common rules are formulated as follows:

- **Commercial relations** – commercial relations determine which data will be shared amongst a customer and service provider[8]. A customer formulates its goal where upon a service provider makes makes a provides a fitting offer or plan. State information is shared

[8] The roles of customer and service provider are formulated in the FEDeRATED Interim Master Plan.

by data, which can be on physical aspects like goods and trucks, but also relate to financial aspects.

- **Compliance to regulations** – authorities have to ability to control goods flows based on regulations. These authorities should indicate which state information they require and when they require it.

State information can be specific to a commercial relation, but should always consists of the following main components with respect to the physical processes:

- **Physical objects** – which are the physical objects involved, e.g. goods, container, trucks and vessels. These will be modelled as Digital Twin.
- **Time and place** – what is the location of these objects in the present, past, or (predicted or required) future. This is a link between Digital Twins, Digital Twins and location, and Digital Twins and business transactions between customer and service provider.

State information will be shared by '**events**', section 4. The actual state only represents events linking physical objects to each other or to a location at a given time for a given (sub)system, e.g. a port area or a corridor. It will give for instance all barges with their cargo at a given time on the Rhine-Alp corridor. It might also give the traffic density of a road. All types of state data can be retrieved specifying a logistics system from this perspective.

3 ELEMENTS AND PERSPECTIVES

This chapter identifies the various perspectives to the architecture. These perspectives give guidance to development, maintenance, and deployment of the various components by the different stakeholders. The constitute the basis for creating a protocol stack.

3.1 FEDeRATED elements for data sharing

Developing of these perspectives is based on architectural approaches, like TOGAF and the elements that have been identified in FEDeRATED. These elements are shown in the following figure:

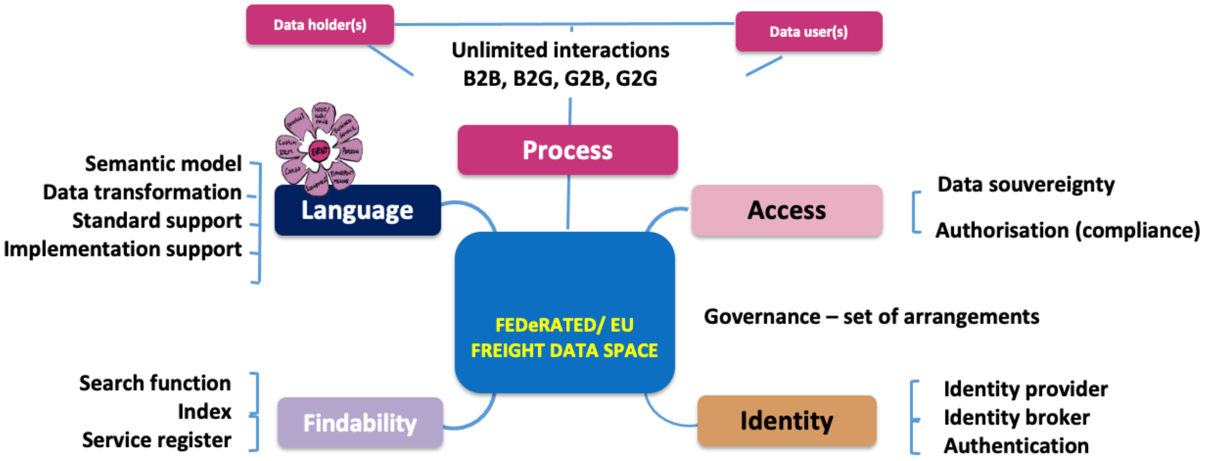


Figure 3-1 elements for data sharing

These elements have also been identified by DTLF II SG2 and are documented (slightly different and with different wording) in the intermediate report (to be published).

They can be given as:

1. **Language** – the semantics and interaction order (process choreography) for data processing by heterogeneous systems or platforms
2. **Findability** or **discoverability** – it is about being able to find service providers and data (the so-called service registry) and data that an organization needs for its tasks (index and search function). The latter is filled in with 'Linked Data': an organization receives a link to data as an indication of the data they may access.
3. **Access** – a company gives competent authorities access to data that they need in accordance with legislation and the company is willing to make available to others, as open data (data registry) or via links (index) that have been shared.
4. **Identification, Authentication and Authorization** (IAA) – the unique identification and authentication of a person and their authority granted by their employer (verification of authorization).

The elements ‘findability’, 'access' and 'IAA' together constitute ‘**data sovereignty**’, being a (main) principles of data sharing infrastructures in general as identified in the FEDeRATED Vision further elaborated by the FEDeRATED Interim Master Plan. A common language is needed to process data.

The FEDeRATED elements have also been identified by DTLF II SG2, as follows (see also next figure 3.2):

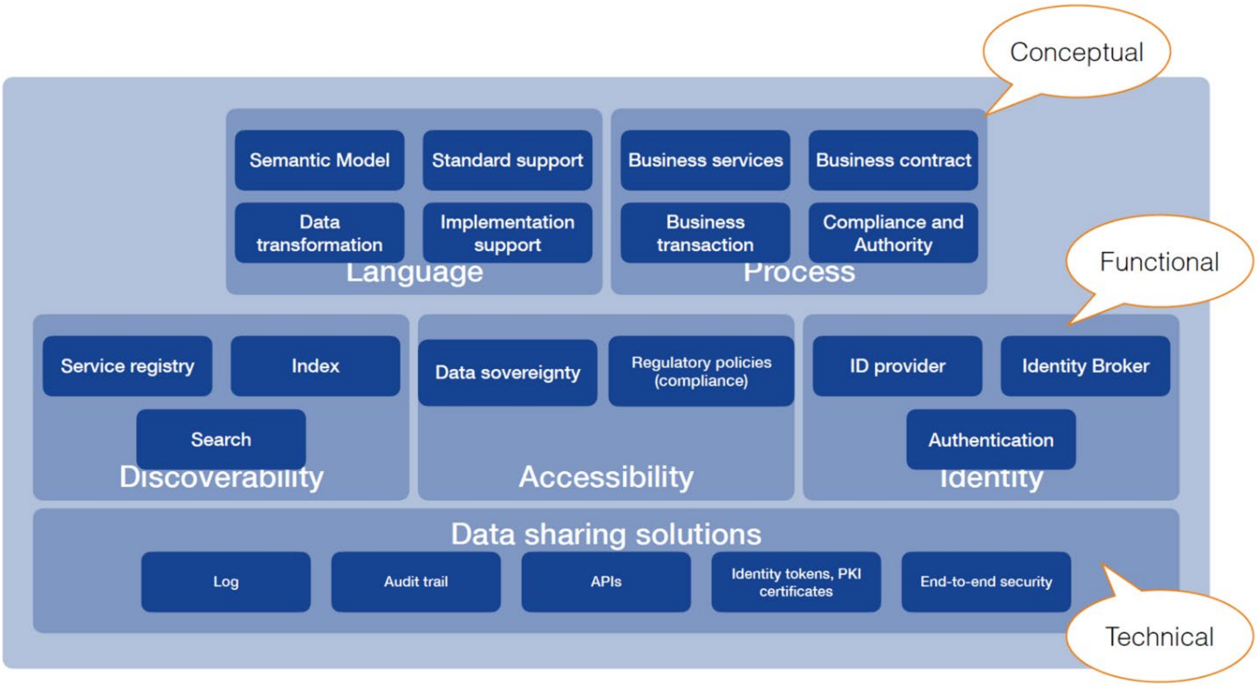


Figure 3-2: DTLF II SG2 building elements of the architecture

1. Conceptual level. These are conceptual building elements supporting semantic – and organizational interoperability. They are:

   - Language – a semantic model supporting data sharing in supply and logistic chain operations (multimodal) supported by tools and algorithms for its implementation by individual organization, supporting required standards.
   - Process – data sharing between business processes of collaborating organizations, business-to-business, business-to-administration, and administration-to-busines.

Functional components – the required components to realize data sharing in a technology independent way. These are grouped into discoverability of data and business services, data sovereignty for B2B and B2A based on access control compliant with regulations, and IAA mechanisms aligning with existing solutions like the eIDAS Regulation, iShare developed in the Netherlands, and Decentralized Identities (DIDs).

Data sharing solutions – these are all types of solutions (proprietary, COTS – Commercial Off The Shelve, and open source) and third party (platform) services. They must implement functionality, independent of any application area. The functionality basically supports non-repudiation, technical standards like REST APIs, PKI certificates, and end-to-end security mechanism to share commercial - or privacy sensitive data.

The conceptual level is independent of any implementation. It is used to configure the discoverability and accessibility components. Whenever an organization selects a data sharing solution, the identified functionality needs to be provided. If a solution does not provide it, an organization needs to implement it additional to that solution. The DTLF II SG2 figure shows some additional features of the architecture, especially the generic requirements of data sharing solutions. These will be addressed by the architecture in this document.

3.2 Stakeholder groups

The focus of FEDeRATED is on the following stakeholder groups:

- **Organizations** – these are authorities and enterprises that further digitize and share data. An individual enterprise can participate in multiple business cases, and potentially must implement data requirements in the context of various regulations. Enterprises and governing authorities are in fact 'users' in the context of the FEDeRATED architecture; they share data.

- **Regulators** – these formulate regulations, their data requirements, and its implementation in collaboration with for instance EU Member States.
- **Industry Associations** – these often develop amongst other guidelines for data sharing with and on behalf of their members. They are organized by for instance modality (e.g. road, rail, air, inland waterways, and sea) and/or cargo type (e.g. chemical industry, commodities, container transport by sea).
- **IT solution/platform providers** – these stakeholders deploy services to enable data sharing by supply and logistic chain organizations. They will have their own business – and governance model and deploy a particular data sharing paradigm. As of currently, most of these providers support messaging and APIs. New entrants might support Linked (semantic) Data. Integrators and platform service providers are examples.

Recognized standardization bodies like ISO, CEN CENELEC, OASIS, W3C, and UN CEFACT can also be involved as stakeholder groups. However, these bodies are currently not facilitating heterogeneity and complexity as described hereafter.

3.3 Heterogeneity and complexity

Language includes all modalities and types of goods for both business relations and compliance to regulations. There are several aspects of interest with respect to this model (in analogy with 'language'):

1. **Language evolves** – the model will always evolve with new data sharing needs. Companies, industry associations and legislators (at Member State and EC level) must have the means to further develop the model.
2. **Own 'dialects'** – it is not possible to support everyone from a central organization with solutions to participate in the infrastructure. Two approaches must be supported:
   - Every organization must be able to formulate which data it can make available and integrate with their internal systems (plug and play).
   - Organizations must be able to make agreements for their requirements themselves. This also depends on the scale; it is the intention of FEDeRATED to offer solutions for the European Union. Organizations must be given the means to specify their own data needs and data to be delivered. They can form communities.

   Note that some may call a community that implements its 'dialect' constitutes a data space. Via the common language, these data spaces are by nature interoperable. Thus, governance processes are required.
3. **Own 'language'** – there will always be organizations and associations that have their own language, for example their own standards. A translation with that language is needed, which is called 'matching'. An association having its own language is no problem if users of this language are only applying that one and not active in other industry areas.
4. **'Standard phrases'** – the model offers so much freedom that predefined settings are required, for example for electronic documents (eCMR, eB/L, etc.) and supply chain visibility (see before). Often these types of standard 'phrases' are specified by industry associations (think of DCSA – Digital Container Shipping Association – and IATA – International Air Transport Association) and regulatory bodies. For example, these standard phrases provide access to data in internal IT systems from a link that has been received. This is necessary for eFTI implementation, for example, but the same mechanism can also be used for EMSWe and for supply chain visibility.

These kinds of aspects of 'language' are related to governance – who has control over 'language' and how does it connect with other 'languages'. For this, resources, 'tools', must be developed. A governance should be involving all stakeholders.

3.4 IT perspectives established

Language is at the heart of the specific IT perspectives enabling data sharing. These perspectives must work together and constitute the functionalities of the achitecture. Combined, these functionalities must be linkable (interoperable):

1. **Service Registry (management perspective)** – this includes the different aspects o'language' and its governance as will be supported by the Service Registry. In the elaboration and application of this perspective, it will become clear that the model must be adjusted to comply with standard ‘sentences’.
2. **Security perspective** – this is about unique identification of people with credentials. It should allow employees of one organization to access data at another organization, where the latter can verify identity (authentication) and authorization of that person (veri’fiable credentials). This might be without revealing details of a person, like is name and function/role.

   In addition to identification, this perspective is also about security of data exchange and robustness of systems against cyber-attacks. For the security of data exchange between systems, public key infrastructures (PKI) have been developed; this is built on with eIDAS certified PKI certificates. Robustness against cyber-attacks must be implemented in accordance with existing ISO standards and the Cyber Security Act and is therefore not considered. Other aspects such as non-repudiation and data integrity fall under the data sharing perspective. These aspects are already documented in the DTLF II SG2 Interim Report and are adopted by FEDeRATED.
3. **Data sharing node** – the actual components for findability of service providers and data and the sharing of (linked) data is set up with so-called ‘nodes’. The main aspects of these (conceptual) nodes are to identify the data distribution algorithm, i.e.
   - who receives which links, and
   - how can data quality be assured (event logic, correctness/completeness of data, etc.).

   A node, which fully supports the language, must also be able to support one or multiple options of the presentation -, security -, and connectivity protocols (see introduction) like REST APIs using ‘https’. This also requires a so-called 'semantic adapter': transformations between various presentation protocols via the semantic model.

   The concept of a ‘node’ can be implemented by a stakeholder, a platform, or can be provided as a (cloud) solution. Part of the required functionality can also be implemented by existing IT systems of stakeholders.

IT architectural frameworks also identify a ‘**monitoring perspective**’. In fact, this perspective is implemented by authorities based on the principle of ‘goal binding’ (see the FEDeRATED Interim Master Plan, section 4). Each participating organization and platform service provider will have to offer a monitoring perspective for compliance.

Language is at the heart of these three perspectives. Management is about language; credentials can be formulated in terms of language (to which data does someone have authorized access) and the Data Sharing Infrastructure (see next figure) contains language to be able to share data and to support access control.

The interfaces between the various perspectives are shown in the next figure

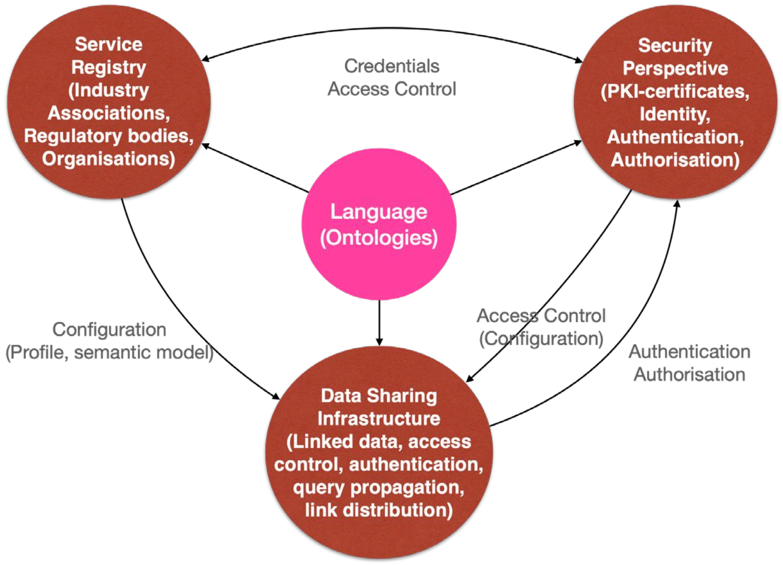

Service Registry (Industry Associations, Regulatory bodies, Organisations)
Credentials
Access Control
Security Perspective (PKI-certificates, Identity, Authentication, Authorisation)
Language (Ontologies)
Configuration
(Profile, semantic model)
Access Control
(Configuration)
Authentication
Authorisation
Data Sharing Infrastructure (Linked data, access control, authentication, query propagation, link distribution)

Figure 3-3 interfaces between the various perspectives

A process flow between these various perspectives is formulated as follows:

1. The semantic model is configured by all ‘nodes’ in the data sharing perspective that constitute the data sharing network.
2. Each organization in its role of user in the data sharing perspective formulates its profile. The process for an organization looks like (high level):
    a. Specify features that further specify capabilities or requirements of the organization. Examples: container sea transport by a shipping line and eFTI regulatory data requirements for a governing authority in road transport.
    b. Search for predefined configurations that are formulated by Industry Associations and/or Regulators. These predefined configurations are found by the features.
    c. Tailor the predefined configurations to ones needs, e.g. adapt a generic predefined eCMR for container transport by road.
    d. Publish the profile to the data sharing perspective in its function of service registry.

    Similarly, this process can be specified to support (open or community) data sets.
3. An individual organization assigns an identification and credentials to its employees. These credentials are expressed by the semantic model and/or predefined configurations. For instance, a particular inspection officer of an authority will receive credentials to access an appropriate eFTI data (sub)set. These credentials can be role – and/or attribute based. The Identity can be Authenticated by the data sharing perspective and the credentials can be verified, meaning they exist and are valid.

    There can be a case, where an employee needs to formulate its access rights on the fly, e.g. police wanting to inspect a particular transport means based on a court order. This employee has the credentials to formulate these access rights, make them available when accessing the data and provide the court order as a legal basis.

These processes will be detailed in this document, together with their structures and formats (semantics and syntax).

4 LANGUAGE – SEMANTIC MOIDEL

This section will contain an update version of section 5 of the Interim Master Plan, that will be provided at a later stage. It contains already a brief update to section 5 of the Interim Master Plan.

4.1 The use of semantic technology

4.1.1 Technology stack

To optimally meet requirements, semantic web technology has been selected:

- **Data requirements representation** – these are expressed by a semantic model represented as an ontology with various constraints. OWL (Ontology Web Language) SPARQL (SPARQL Protocol and RDF Query Language), SHACL (SHApe Constraint Language), and RuleML (Rule Markup Language) are some of the applicable open standards.
- **Data representation** – Resource Description Framework (RDF) or Java Script Object Notation – Linked Data (JSON-LD) are examples of a syntax that seamlessly integrates with its representation. Triple stores are used for storing data directly associated with their semantics.

4.1.2 Justification of applying semantic technology

The main reason to apply semantic technology is

1. *it best fits the core concept specified in section 2;*
2. it can be applied for linking various data sets to supply and logistic chain data;
3. it creates mappings between all types of internal data sets and (implementation of) open standards;
4. the technology can be used for data preparation in data analytics; and
5. semantic technology can be interface with legacy data models.

The Vision of FEDeRATED and the objective of the DTLF is to create an open and neutral data sharing infrastructure with a federation of platforms that provides a level playing field. This requires the specification of what is called ‘Technology Independent Services’ provided by the federation of platforms, platform interoperability, and the ability to share data between business stakeholders for business purposes without prior agreements. The latter is called ‘plug and play’: a capability where a user can register with the infrastructure and be able to share data with all others that are also registered to support its business services and – goals

In this context, a governance structure will be established in which various stakeholders can play a role (see before). These stakeholders should have machine-readable constructs that can be referred to, updated, etc. for sharing data. Semantic technology is selected since it is supported by open standards and open-source tools, but also by freeware and commercial (cloud) service providers. A so-called upper ontology can be constructed representing data requirements for multimodal logistics operations as a basis for lower ontologies specifying details. The upper ontology represents the main concepts that can be specialized by lower ontologies. Governance is still under development in the DTLF; the management perspective will support governance.

Another reason for applying semantic technology is in data preparation for data analytics. Data analytics considers three main aspects, namely data quality, technical – and business expertise. The business expertise is for applying data analytics in the context of business/logistics processes. Technical expertise is on the infrastructure and data management aspects, in relation to for instance data analytics applications.

The technical expertise refers amongst others to knowledge of data and its semantics. Studies have shown that data analytics involves data mining, data management, statistical analysis, and data presentation, before actual performing data analytics and interpreting the output of data analytics. The following figure gives an estimate of the time spent on various steps in the context of data analytics.

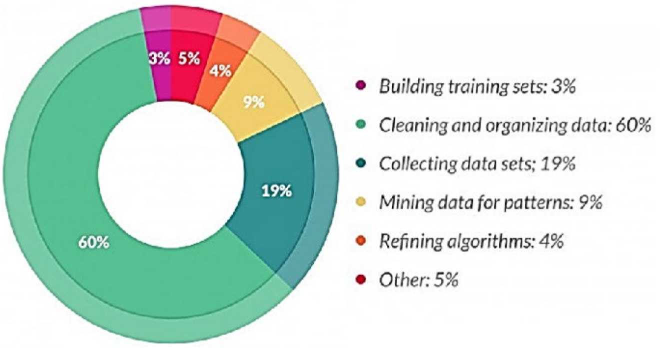


Figure 4-1: estimation of time spent on data analytics[9]

The previous is just an example, similar estimates are given by other sources. The objective of applying semantic technology and the architecture given in this document is to reduce the time of cleansing and organizing data and collecting data sets and thus have a better focus on the other activities (mining, refining algorithms, etc.). By applying a common semantic model that is applicable for any supply and logistic chain operators and not specific for customs, the technical expertise is also expected to increase.

In general, one could state that having a common language for multimodal logistics will contribute to innovation, since the potential market for applications increases. This is applicable to data analytics and all other types of applications that can be developed to support logistics service providers, their customers, and authorities.

At present, many data platforms use traditional data models and semantic technology can be used to interface with those models. This creates an opportunity for migration from traditional data models to semantic models and use both in parallel to provide both backward and forward compatibility.

4.2 Design choices for constructing the model

To be updated from the Interim Master Plan with additional information on for instance Digital twins, events, infrastructure, classifications, business transactions, organizations/actors.

4.3 Implementing business collaboration – business transaction choreography

This section needs to express business services, interaction types represented by events, and the choreography of interaction types. The outcome is the configuration of ‘event logic’ of a node (section 7). This should also be reflected in the Service Registry (section 5).

[9] Sarih, H. et al – Data preparation and preprocessing for broadcast systems monitoring in PHM framework, 6th Internetional Conference on Control, Decision, and Infomation Technologies (CoDIT’19), April 2019, Paris, France.

4.4 Specific issues

There are two that currently need to be addressed:

- Construction of supply and logistics chains based on the choreography
- Consignment versus shipment: this relates to the previous. The FEDeRATED proposal is not to mix process – (chain configuration) with data aspects, since the process aspects are dynamic and data represents a static situation.

5 THE SERVICE REGISTRY

The management perspective provides the required settings for the data sharing perspective. Since data sharing can be completely open to everyone, the number of degrees of freedom is probably too high for most organizations. They should have clear guidelines of what to implement to be able to share orders and visibility data and be compliant to regulations. Furthermore, it is not feasible to develop and manage all settings centrally. It would take too much time and effort, and thus be too costly.

This leads to two assumptions:

1. **Predefined settings** are available. Examples of these settings are structures like 'electronic CMR' (eCMR), 'electronic Bill of Lading' (eB/L), or 'eFTI data set'. These settings are developed by regulatory bodies and industry associations.
2. A distributed **tooling** infrastructure is available to support development, maintenance, and sharing of any predefined settings between stakeholders to enable rapid deployment.

Since this section is not about development of predefined settings, which should be done by industry associations and regulatory bodies, it is about specification of tools. These tools compose the 'Service Registry'. The Service Registry has to support the various value chains for providing predefined settings. Firstly, these value chains are presented and secondly the basic of the Service Registry.

The draft proposal for the Service Registry needs validation by Living Labs. The proposal is based on input like the MCP Service Registry (MCP – Maritime Cargo Platform[10]), the Semantic Treehouse[11], and standards/initiatives in the area of web services (e.g. USDL – Unified Service Description Language, a W3C standard[12]).

5.1 *Configuration Value Chains*

The objective is to manage, share, and deploy predefined settings. Enterprises need to comply for instance with data requirements of authorities, based on regulations. Enterprises also share data in a commercial relation, like for instance visibility events and links to business document data sets.

In this context, we distinguish between two potential configuration value chains that manage and share configurations that have to be deployed by an organization, namely:

- **Private configuration chains**. These are industry associations, communities (or data spaces), and logistics enterprises:
  - Industry associations provide settings to their members. Industry associations share these settings with other industry associations to re-use what has been developed.
  - Communities are centered around a hub like an airport, port, or inland port, supply chains, and corridors. They manage and share predefined settings for the community members. A Living Lab can be a temporary community at the start resulting in a more permanent one lateron.

[10] Developers.maritimeconnectivity.net/serviceregistry/index.html

[11] Tno.semantice-treehouse.nl

[12] Alistair Barros, Daniel Oberle (editors), *Handbook of Service Descriptions – USDL and its methods*, Springer, 2012

    - (large) Logistics enterprises specify their settings and data requirements for their service providers or customers.
- **Public configuration chains**. These are regulators and competent authorities that have to implement regulations.
    - Regulators can act at various levels, like EC and Member State, but also a a local level of for instance a municipality. In the context of a regulation, data requirements are formulated based on the semantic model. These data requirements are both the minimal requirements and the maximal data set that can be required.
    - Competent authorities govern a regulation, mostly at a MS - or regional level. They implement a data subset of the regulation that contains the minimal required data set for that regulation and relevant parts of the maximal data set. They cannot implement more than the maximum data set that has been agreed upon by the regulators.

Any enterprise in supply and logistic chain operations can compose its predefined settings by selecting those settings (1) that are relevant to support its business and (2) those that need to be implemented for compliance based on their goods flows. The first selection can be made via its membership of an association. The second selection needs to have identiable elements (metadata) based on for instance geography relevant to competent authorities in combination with properties of Digital Twins like dangerous cargo details.

The Service Registry should support both configuration value chains.

5.2 The Service Registry

The Service Registry is an IT tool that can be applied by various stakeholders like Regulatory Bodies, Industry Associations, enterprises, and authorities for development and management of configurations of the Data Sharing Perspective. An organization must eventually manage its configurations for the processing layer functionality, the SHACL Validator, and the semantic adapter of a node (section 7.5), but should also provide settings for those that implement messaging or Application Programming Interfaces (APIs).

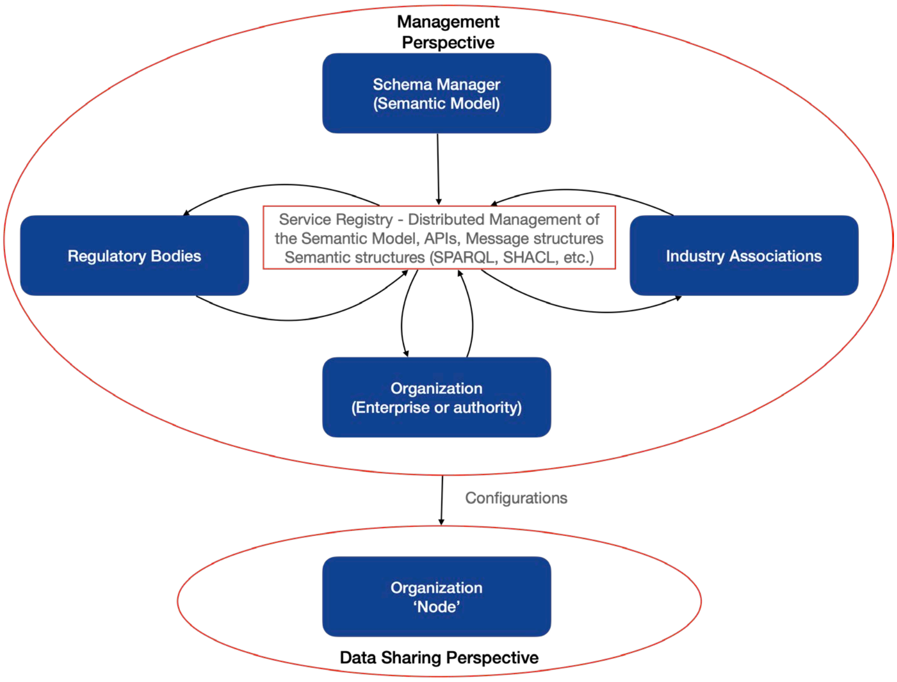

Management Perspective
Schema Manager (Semantic Model)
Regulatory Bodies
Service Registry - Distributed Management of the Semantic Model, APIs, Message structures Semantic structures (SPARQL, SHACL, etc.)
Industry Associations
Organization (Enterprise or authority)
Configurations
Organization 'Node'
Data Sharing Perspective

Figure 5-1: The Service Registry and its users

The ‘Service Registry’ can be applied for management of the Semantic Model, semantic structures like SPARQL queries representing a document data set and SHACL to validate the output, and APIs and message structures. The semantic structures, APIs and message structures are all generated. Depending on the deployment of the Data Sharing Perspective, the APIs and message structures must be stored separately since those require version management. Semantic structures deployed by a ‘node’ can change over time.

The content of the Service Registry should be ‘discoverable’: a data user, a customer, or an organization configuring its node should be able to find the appropriate data sets. Industry Associations should be able to re-use their predefined settings, etc. Thus, search criteria must be defined. This is what is called ‘metadata’. The semantic model itself is also part of those search criteria. If for instance a data user searches for an organization that can provide ‘traffic data’, it should give a particular set of instances of the infrastructure like ‘roads’ in a ‘region’. ‘Traffic data’ as such can also be defined as a SPARQL on the model, for instance by an association of infrastructure managers in road transport.

Where the previous figure shows that the Management Perspective produces the configurations for the Data Sharing Perspective, this will be a particular organization developing and maintaining those for its node, API – or message-based implementation.

5.3 Basic concepts of the Service Registry

This section specifies the basic concepts of the Service Registry, resulting in metadata. It starts from the perspective where a user of the Service Registry is able to construct collections of related ‘views’. Additionally, metadata for users must be included.

5.3.1 Collections and views

The following definitions are given:

- **Collection** – (a set of one or more) views with a logical meaning to a user. Thus, each user can specify its own collections. A collection can consist of a single view.
  A user (community) defines a collection. A collection represents for instance the set of interactions and their logic in the context of a business service, relevant interactions in the context of a regulation, or a paid or open information service (e.g. ‘weather service’).
- **View** – a selection of the Semantic Model that is relevant to a user. A view represents an interaction (see for instance USDL).
- **View type** - a particular entry to the semantic model for creating a view. The types of views as based on the semantic model are (currently):
  - Event – user defined events that are expressed in atomic events.
  - Infrastructure – views relevant for the infrastructure, for instance opening hours of a hub. This view type will (probably) require subtypes, since infrastructures are represented by a variety of concepts (road, rail, terminal, inland waterways, etc.).
  - Business document – user defined document data sets like eCMR, eB/L.
  - Business services – logistics and other types of business services relevant to users.
  - Digital Twin - user defined views on Digital Twins. This view type will (probably) also require subtypes, since the number of subtypes of Digital Twin will increase.

Views can be complex. For instance, a view would be a ‘container track’ from the perspective of Digital Twin – Container.

5.3.2 Detailing ‘collection’

A collection can be created, copied or imported from another collection, published, and updated.

Co-financed by the Connecting Europe
Facility of the European Union

Deletion (or deactivating) of a collection is only feasible when no view in that collection is applied in the Data Sharing Perspective.

To be able to copy a collection, it must be discoverable. This can be done by including metadata for a collection:

- Name of the collection – a free text description
- UUID of the collection – unique identification of a collection
- Collection type – the main concept of the semantic model for the views in the collection. The view types are covered, but also particular concepts and data properties with selected values can be given. For example, the collection may contain all events for road transport or those for vessels entering a port.
  The following collection types are distinguished:
  - Business services – a collection type of business service specifies all relevant interactions and their sequence of a particular business service (e.g. transport, load, discharge, transship, store). The business service is given.
  - Regulation – all interactions and their sequencing for compliance by enterprises to a particular regulation. The regulation is given (see the optional field ‘regulation’).
  - Supporting services – any data that is made available by a service, including a specific interaction sequencing if applicable. This can be any type of data.
- Applicability restriction – any restrictions indicating where, when and for whom the collection is applicable. An example is that events for vessels entering the Rotterdam port are given. Another example of a restriction can be a selection of Digital Twins, e.g. ‘container’ transport by ‘sea’, where the latter refers to ‘vessels’.
- Logic (dynamic behaviour) – this represents any sequence restrictions (‘choreography’) imposed on the interactions specified as views. Sequence restrictions also provide a specification of the sender of an interaction.
- Regulation (optional) – name, description, and URL of a regulation for which a collection is applicable.
- Start validity date – the date/time at which the collection is applicable
- Publication date – the date/time at which the collection is made public. This date/time is preferably before the start validity data
- End validity date – the date/time at which the complete collection cannot be applied.
- Governing body – UUID of user that has created the collection.

Collection type defines the search criteria for similar collections and their views.

For regulations, the collection type indicates the objects that are subject to a regulation. An applicability restriction can be at EU level, Member State level, or authority in a MS with its region. As such a particular collection in a Service Registry serves as what is mentioned a Policy Enforcement Point for event distribution (section 7.5.2). The concepts, properties, and values for collection type need to be accessible as RDF data with a node, where the node uses them to search for applicable regulations. A node can be preconfigured with applicable collection types for event distribution.

5.3.3 Detailing ‘view’

Management of a view in a collection is as follows:

- Create, update, and retrieve – a user can create a new view, retrieve it, and change it according to requirements. At creation time, metadata is included for each view.

- Publish – a user makes a view or a collection open. Any view that is published, is published by an open standard. The latter makes it feasible to share views between different Service Registries (see deployment options).
- Delete – a user can delete a view from a collection, only if it is not referred to by another other user.
- Link – a user can link a view of another user and include it in its collection. This is via the UUID of a view. In case a complete collection is copied, all views of that collection are included.

In case a user has created a logic as part of a collection, all interactions defined by this logic need to be specified by a view.

Each view has the following metadata:

- Name of the view – a free text description
- UUID of the view – unique identification of a view
- UUID of the base view (optional) – the unique identification of a view that is the basis of this view. The base view contains a superset of concepts, data properties, and classifications for this view. This view cannot add anything to the base view, without consent of the governing body of that base view.
- Governing body of the base view (optional) – UUID of the governing body that has developed the base view.
- View type – the main concept of the semantic model for constructing a view (see before).
- Applicability restriction – any restrictions indicating where, when and for whom the view is applicable. In case restrictions are given for a collection, these can be refined for each view in a collection. An example is that an event for vessels taking a pilot onboard in the Port of Rotterdam is in a particular area at sea.
- Start validity date – the date/time at which the view is applicable, e.g. for data retrieval
- Publication date – the date/time at which the view is made public. This date/time is preferably before the start validity data. In case all views of a collection are published, the collection has the publication date of the final view that has been published.
- End validity date – the date/time at which the complete view cannot be applied.
- Governing body – UUID of user that has created the view.
- Re-usability restrictions – any permission that is required to re-use a particular view. This mechanism is used to create a community, see further.

A view results in a configuration of a node or another type of deployment environment in the Data Sharing Perspective. Thus, a view must have a technical binding that is available at publication date. Such a technical binding consists of three aspects, namely:

- **Standard support** – to support open standards, existing ones can be selected to share data. A data carrier representing an open standard requires data transformation from and to the semantic model. Standard support requires a mapping supported by the semantic adapter of a node.
  A data carrier is specified by:
    - Data structure – an identifier of a data structure (e.g. eCMR, eB/L, OneRecord).
    - Controlling agency – the body that has submitted the data structure (e.g. UN CEFACT, DCSA, IATA).
    - Version – the version of the data structure.
    - Technical binding (presentation protocol) – the syntaxis that can be applied for the data carrier (a list of at least one).

    - Namespace – any relevant namespace that provides a structured file of the data carrier. It can refer for instance to an XSD (in case of XML as technical representation)..
- **Security - and Connectivity Protocol** – a specific protocol that supports data sharing with a technical binding. This is applicable for APIs. These can have a web service or REST-based protocol.
- **Technical binding** – a view can have several technical bindings, namely:
    - SPARQL - a SPARQL query between nodes
    - SHACL - a SHACL document for the SHACL validation
    - Internal format - a simplified RDF scheme that serves as an interface of the Semantic Adapter to process messages or API data.
    - (YARR)RML – rule markup language is a machine-readable structure for transforming data; YARRML is providing flexibility and ease of use.
    - JSON(-LD) – Java Script Object Notation for sharing data and a variant for representing triples (JSON – Linked Data).
    - XSD – XML Schema Definition.

  Each technical binding will have a version. In case of a selection of an API Protocol, a technical binding like JSON(-LD) or XSD will be applicable. (YARR)RML, JSON(-LD), and XSD are examples API bindings.

A technical binding like SPARQL and an API Protocol have an **endpoint**. This endpoint is specified by the user implementing the binding or protocol. This is called the instance in the MCP Service Registry.

In case a view provides access to data, accessibility policies are applicable. These are formulated by authorities, for instance by an API and/or SPARQL query of a view, in which case a competent authority has compulsory access, or by a data holder itself. Accessibility can be based on a common classification, namely:

- **Open data** – data is accessible by any data user. No authentication or access control is implemented.
- **Community data** – any data user that can be authorized as a member of a community (or has a particular role) has access to the data. This can refer for instance to a collection of a governing body that can only be accessed by members of that governing body. Membership rules should be implemented to apply the views in the collection.
- **Permissioned data** – a data holder decides on permission to a data user upon request. This can also be in a commercial relation where a link has been shared prior to access to data by a data user.
- **Commercial data** – data can be accessed according commercial conditions like payment, licensing, etc.

5.4 Users of a Service Registry

A user manages collections and their views and make them available to other users (potentially under conditions). There are two types of users, namely those that share data between resources and those that provide predefined settings like regulating bodies and industry associations.

The following specialization structure for users is applicable:

- **Organization** – the supertype functioning as legal entity. It has the following general properties
    - Name of the organization

    - URI (Uniform Resource Indicator) where more data of an organization (resource) can be found
    - Brief description of the organization
- **Authority/Enterprise** - subtypes authority and enterprise of organization. These have the following properties:
    - **Enterprise profile** – the main concepts of the semantic model supported by an enterprise. These are relevant business services that can be provided (like 'transport' and 'storage'), types of cargo (containers, goods, etc.), types of products (e.g. chemicals, high value electronics), dangerous goods, and modalities (road, rail, sea, etc.) that are supported by an enterprise. These can be used by an enterprise as a user to find collections like those of business services.
    - **Enterprise profile restriction** – any restrictions to the applicability of a profile, for instance a geographical coverage (corridor(s), country(-ies), etc.). Externalisation (outsourcing) could also be an indicator, but mainly in terms of customers to discover a particular service provider. Restriction are formulated as rules on the semantic model.
    - **Authority regulation(s)** – the regulations that fall under the responsibility of the organization
    - **Authority restriction(s)** – any restrictions like geographical area in which an authority is operational.
    - **Contact details** – both an enterprise and authority have contact details.
- **Resource** – any resource of an organization that can act as data holder and/or -user. This can be anything ranging from an IT system/platform, a Digital Twin, and a business unit. It will have the properties as 'organization' and a name of the resource. A resource has a technical protocol for data sharing with other resources:
    - Name – name of the protocol.
    - Version – version of the protocol applied.
    - Security mechanism – the type of encryption over the link, including the certficates

The URL of a resource is the endpoint for APIs and/or SPARQL queries. Whenever an enterprise or authority have implemented a node, it is the endpoint of that node. Any internal endpoint of an organization connecting to that node is not made public.

5.5 Deployment options of the Service Registry

There are basically two deployment options of the Service Registry that can be combined, namely:

- Single user – a single user implements its own Service Registry on a server and manages its own content. It can find and access the content of all other Service Registries.
- Multi-user – multiple users share one Service Registry on a server and manage their content. Like for a single user, it can find and access the content of all other Service Registries.

Whether these options are required is not yet known. One can envisage that EU legislators apply a multi-user solution with their Member States, where even different DGs of the EC share the same solution. The following application could be envisaged to support data sharing in the context of legislation:

- An EU legislator manages a collection of view(s) to support data sharing in the context of a regulation
- A relevant MS manages its own collection of view(s) for the same regulation, where each of the MS views is based on the EU views.

- An authority implementing a regulation in an MS manages its own collection with view(s), where the MS views are taken as baseline views. The views of the authority are published for compliance by enterprises.

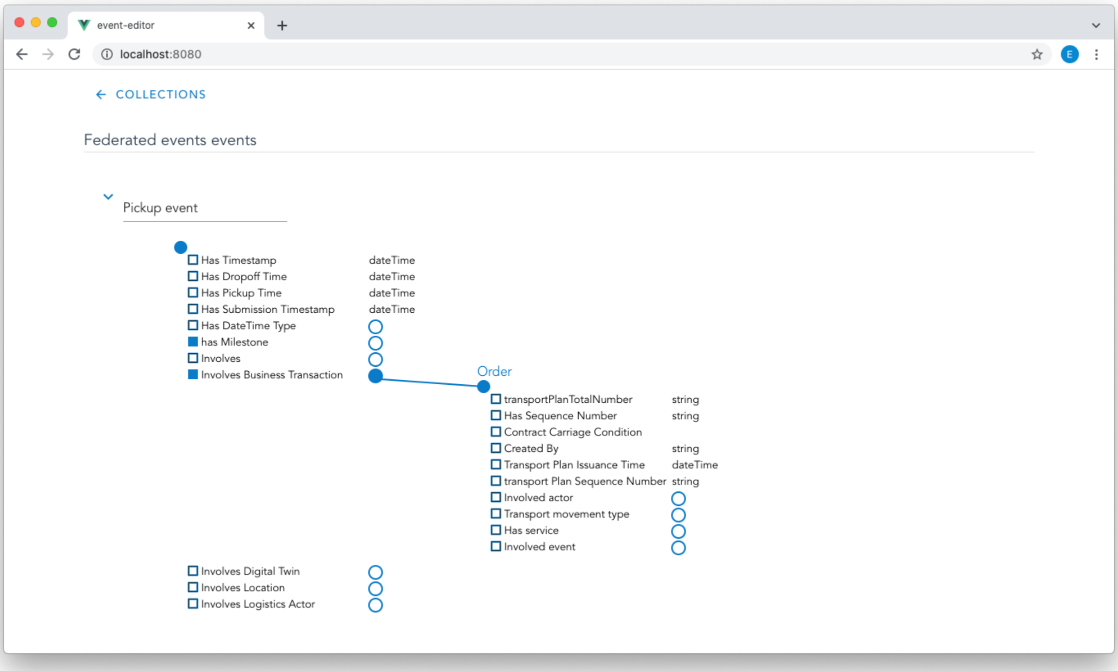
Similar constructions can be made for SMEs in a country, where one central body manages and makes available relevant collections and views to their members. They can assist their members in formulating their 'resources', which can also be done by platforms or other types integration providers.

5.6 *Final notes*

The content of this section will be further elaborated by applying it to a Living Lab. Prototypes of the Service Registry are under development but have not yet been validated with users. These prototypes also must be consolidated into one. When this one prototype has reached a level of stability and completeness for validation, potential users of the Service Registry need to be involved.

The first step is to evaluate in for instance the architecture – and semantics group which editing interface would be the optimal.

The following figures show a screen view of a user event editor prototype.



Federated events events

Pickup event

●
□ Has Timestamp dateTime
□ Has Dropoff Time dateTime
□ Has Pickup Time dateTime
□ Has Submission Timestamp dateTime
□ Has DateTime Type ○
■ has Milestone ○
□ Involves ○
■ Involves Business Transaction ●

Order
●
□ transportPlanTotalNumber string
□ Has Sequence Number string
□ Contract Carriage Condition
□ Created By string
□ Transport Plan Issuance Time dateTime
□ transport Plan Sequence Number string
□ Involved actor ○
□ Transport movement type ○
□ Has service ○
□ Involved event ○

□ Involves Digital Twin ○
□ Involves Location ○
□ Involves Logistics Actor ○

Figure 5-2: user interface of a prototype user event editor

Some screen views of another prototype are shown hereafter. The first view is that of the start screen showing multiple so-called 'projects'. Access control is implemented for these projects.

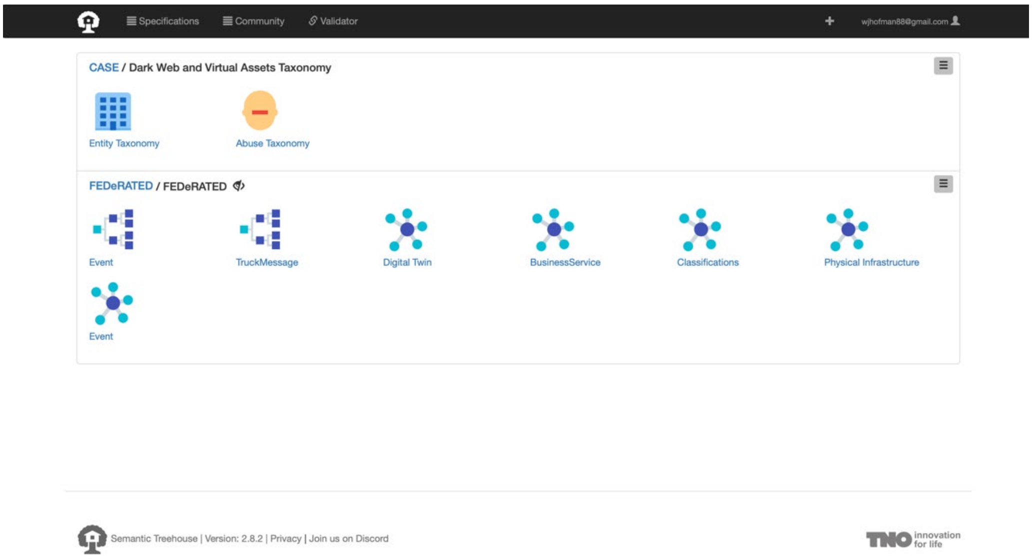
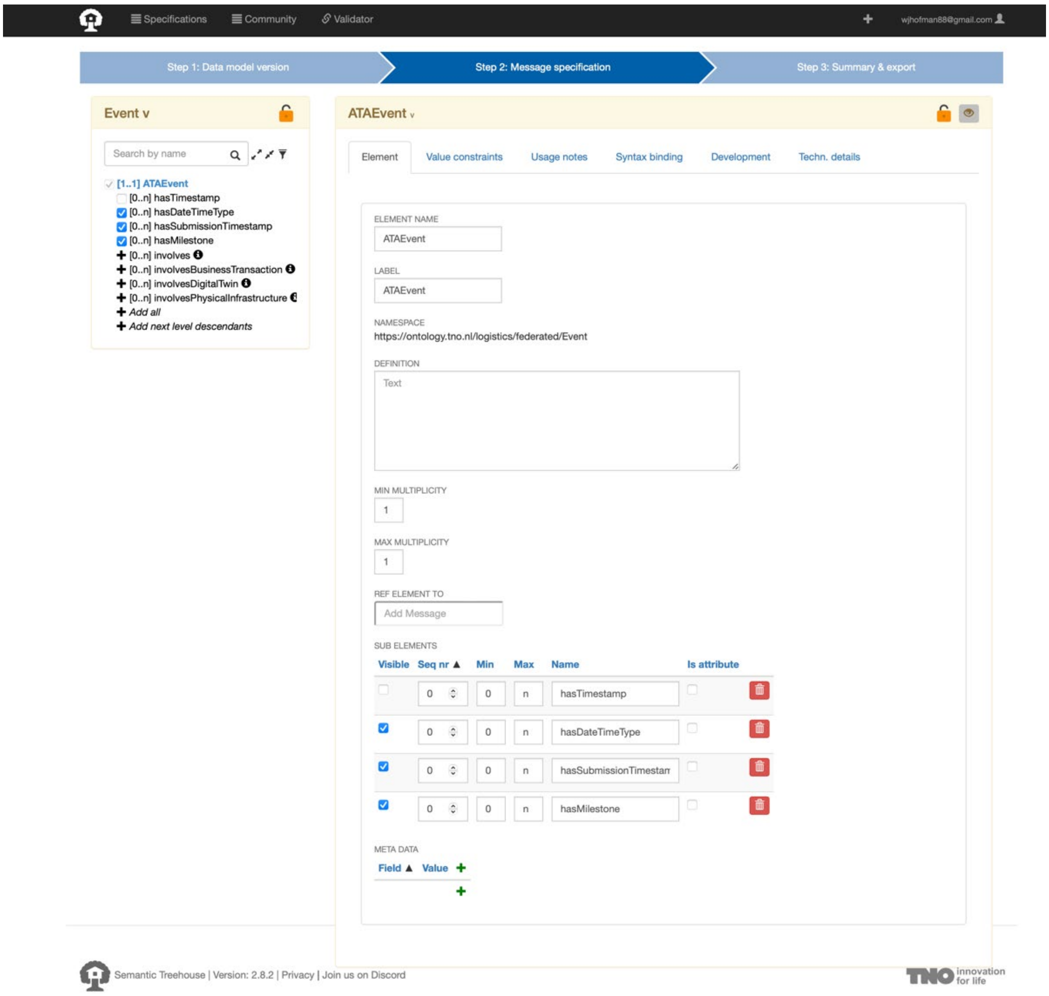


Figure 5-3: user interface of a multi-user application prototype



- ✓ [1..1] ATAEvent
  - □ [0..n] hasTimestamp
  - ☑ [0..n] hasDateTimeType
  - ☑ [0..n] hasSubmissionTimestamp
  - ☑ [0..n] hasMilestone
  - + [0..n] involves
  - + [0..n] involvesBusinessTransaction
  - + [0..n] involvesDigitalTwin
  - + [0..n] involvesPhysicalInfrastructure
  - + *Add all*
  - + *Add next level descendants*

ELEMENT NAME
ATAEvent

LABEL
ATAEvent

NAMESPACE
https://ontology.tno.nl/logistics/federated/Event

DEFINITION
Text

MIN MULTIPLICITY
1

MAX MULTIPLICITY
1

REF ELEMENT TO
Add Message

SUB ELEMENTS

Visible	Seq nr ▲	Min	Max	Name	Is attribute
□	0	0	n	hasTimestamp	□
☑	0	0	n	hasDateTimeType	□
☑	0	0	n	hasSubmissionTimestam	□
☑	0	0	n	hasMilestone	□

META DATA

Field ▲	Value	+
		+

Figure 5-4: user interface of the prototype for construction of so-called message views

The second screen shows how constraints can be formulated for each element in a view, a syntax binding can be generated, etc.

Deployment is also still to be explored. For instance, publication can be on a permissionless blockchain (‘open data’) or a Service Registry might be found via DNS. Since a Service Registry contains all relevant details for data sharing, governing rules have to be defined for allowing Service Registries to the federated network of platforms.

6 SECURITY PERSPECTIVE

Security needs to address two aspects, namely cyber-security of IT systems and components and secure and reliable data sharing and – access. Cyber-security is addressed by the Cyber-security Act and is outside scope. Users of the federated network of platforms need to implement the necessary measures.

As the Interim Report of DTLF SG2 (see dtlf.eu, SG2, Interim Report) indicates, the other aspect of security consists of secure system-to-system data sharing and Identification, Authentication, and Authorisation (IAA). Access control to data in case of a data pull is based on an authenticated identity, where a person has authorization to access particular data. In case of this data access, that person is an employee of a data user accessing data of a data holder.

6.1 Identification, Authentication and Authorization (IAA)

This section gives a conceptual model for IAA. The requirements are mapped to existing solutions. One of the requirements to a solution is that the identity of an employee of a data user accessing data of a data holder needs to be hidden to the data holder. The assumption must be that any stakeholder has sufficient security measures implemented in its own organization.

First, a conceptual model of IAA is given as currently under development based on SSI (Self Sovereign Identities) with DIDs (Decentralized IDentities) is discussed. Secondly, the model is mapped to existing solutions and the state of the art is assessed, resulting in a solution that can currently be deployed.

6.1.1 Conceptual model IAA

For IAA, the following requirements are applied:

- Single Identity - an employee can use the same identification to access data (and IT systems) of its organization and any other organization (based on access control).
- Credentials - each organization provides the relevant authorizations to its employees. These are the credentials (or claims) of those employees. Organizations can issue and revoke credentials; credentials can also have a validity period. The credentials represent the authorization of an employee, for instance to access eFTI data for a particular purpose (‘fit for purpose’).
- Token – an employee can generate and apply a credential that can be verified by a data user without disclosing the identity of that employee. The employee acts on behalf of its organization, using the credentials assigned to him by that organization. A token is generated that can be used one or more times or is valid for a period (lifetime of the token). The token must have sufficient data that allows a verifier to validate that a credential exists.

There are four roles in this scheme, namely :

1. a person (employee) or thing (robot, asset, etc.),
2. issuer of credentials (their employer, owner/exploiter),
3. a verifier (an IT system being accessed), and
4. a register storing and evidence that credentials have been issued.

Verification (5) is done in the data perspective (see later). ‘Things’ also require IAA, although another legal framework for liability and responsibility might be applicable. There are ‘things’ that are owned by a platform/service provider, in which case the provider organizes identity of those things, e.g. sensors.

This model is shown by the following figure (it is the model implemented by the European Blockchain Services Infrastructure – EBSI). The 'token' shown in the figure can be an OAUTH2.0 token.

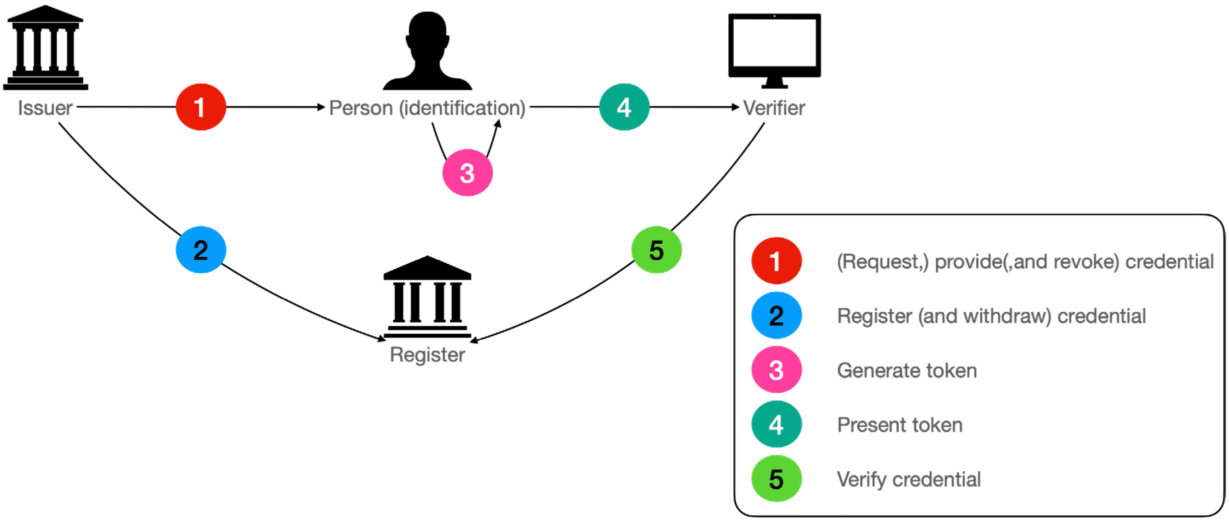


Figure 6-1: roles and mechanisms for applying tokens and verifiable credentials

When generating a token (step 3), two-factor authentication can be implemented. A person provides his username/password combination and must enter a code received via for instance mobile phone.

In this model, an issuer has registered itself by a DID document as a trusted organization. How this would work for supply and logistic chains requires further elaboration.

6.1.2 Mapping the model to existing systems and frameworks

In the case of the eFTI Regulation (and other regulations, acts, and conventions that implement a data pull with an index), an employee of an authority requires access to data of an eFTI platform. The eFTI platform needs to be able to authenticate the identification and validate the authorization given to that employee by its organization.

Currently, organizations have their Identity and Access Management (IAM) Registry(-ies). These store identifications and credentials of employees of that organization. These credentials are verified by internal IT systems when a user tries to access those systems. Mostly, verification is based on a username/password mechanism as indicated (potentially supported by a two-factor authentication procedure).

The current assumption is that each organization will have its security policies for providing credentials. These policies are trusted. It implies that a person can only receive a token if that person has the proper credentials. These credentials will not be verified by a data holder.

In case an employee of an organization has credentials for different roles and depending on the role a different data set is required, a data holder needs to identify in which role data access is required, the exact role must be validated. An example is where an authority implementing multiple regulations, each with its access policies. OAUTH2.0 cannot support this functionality. There are three solutions:

- Change OAUTH2.0 – an extra parameter is added to include a (reference to) a credential that can be verified.
- Apply DID implementations – the credentials and the purpose are shared by a token. The credentials can be verified.

- Different credentials for different regulations/data sets – employees receive multiple credentials that are stored in their IAM Registry.

The last solution is the simplest but might require changes in security policies of an organization. In case an OAUTH2.0 token is applied, the token should indicate which grant to execute. This is part of the response-type of OAUTH2.0. These should be standardized to reflect for instance the regulation for which access is granted.

To apply a IAM Registry of an organization, it should:

- Implement OAUTH2.0
- Refer to the required response type
- Be trusted

The latter aspect can be addressed by a (federation of) Identity Broker(s).

In the longer term, persons might have to implement a DID (Decentralized Identity) solution like SSI (Self Sovereign Identity). Experiments will have to be performed, since it may reduce complexity of the infrastructure. An Identity Broker is not required.

6.1.3 Current state of IAA in the EU – brief overview

The EU develops an EU wallet by which a European citizen can store various credentials (e.g. driver license, passport, identity card, diploma,) issued by an authority, linked to its eID. Information of its eID is also stored on the blockchain. The blockchain solution supporting the 'register' role is developed by EBSI. It can also be used to store trusted IAM Registry APIs and thus serve as Identity Broker.

There are currently two Identity Brokers[13] for persons to access data of another organization based on Identity Brokers, namely:

- eIDAS Regulation – it supports the access to online government services by citizens and enterprises with an eID (electronic IDentity). In some EU Member States (MSs), it also supports the interaction between civilians and enterprises, but not for all MS in the EU. An eIDAS broker contains several certified Identity Providers that can be used in a MS; Identity Brokers of different MSs are not necessarily interconnected for cross-border identification or citizens and enterprises (see 'Implementation of the eIDAS nodes: state of play', August 2020). In the case of eIDAS, the identification of a person is always shared with a data holder, which does not meet the requirements (see the introduction of this section).
- iSHARE – it supports employees of one enterprise to access data of another enterprise acting as data holder. iSHARE registers Identity Providers can be used. It is the intention of iSHARE to install a network of Identity Brokers in various countries. For interaction, iSHARE applies eIDAS certified PKI-certificates for link encryption with https. Authorized data access by users is based in identity tokens, utilizing OAUTH2.0. These tokens hide the user that requires access, but the validity of a token can be authenticated. [14]

[13] Probably there are more mechanisms implemented by Member States, where these mechanisms may differ per MS. These other mechanisms are not considered at this moment, but might be relevant.

[14] Actually, iSHARE complies does not comply with is regulations or legislation covering B2G transactons. A legal basis, e.g. application or eFTI, for a iSHARE broker is required. If this is the case, each MS could apply its eIDAS broker or implement an iSHARE broker. The IAM Registries recognized by these brokers should all be applied for B2B, B2G, and G2B data access by persons.

To fully comply with the Reference data sharing Architecture, all Identity Brokers should be federated in order to support a network of interactions. In practical terms this would enable an officer of an authority in one MS accessing an IT system like an eFTI platform that has registered its IAM Registry in another MS with another Identity Broker.

6.1.4 Conclusion and proposal for IAA

Research is required to explore the potential DIDs, EU Wallets, etc. in the context of a data pull by employees of different organizations.

The DIDs are currently implemented via trusted organizations providing a trusted identity to organisations. An EORI number (Economic Operator Identifier) is an example of such a trusted identity. Such a trusted identity can be applied to integrate the IAM Registry with a federated network of Identity Brokers.

The proposal is to establish a (federation of) Identity Brokers that can be applied by employees in the public and private domain. These might be existing Identity Brokers like eIDAS brokers, this requires further elaboration. However, the identity of employees should not be shared, which is not (yet) supported by eIDAS. It is supported by iSHARE, but data sharing via iSHARE is governed by private law.

Technically, internal IAM Registries need to support OAUTH2.0 and be registered with an Identity Broker that adheres to the previous. Security policies of authorities need further consideration in case an employee of an authority needs access to different data sets based, where each data set supports a regulation.

6.2 End-to-end application security

End-to-end application security is about access to data by an authorized employee of a data holder via its application to data managed by an application of a data user. It is on IAA, but it may also require end-to-end encryption and authentication between the two applications that share the data. The end-to-end security is independent of any intermediate functionality like a node.

End-to-end encryption seems currently not feasible. It requires key management procedures like that of public and secret (private) keys of users. SSI could provide this type of security. Otherwise, the mechanism addressed in the previous section and that presented in the next section are to be applied.

6.3 Link security

Link security is about establishing secure data between system components. In the context of a federated network of platforms, it is about secure links between a node of a data holder and that of a data user.

There are different aspects to link security, namely:

- the trust of a node sending/receiving data.
- data integrity of the data that is shared between both nodes.

There are different mechanisms to provide data integrity and trust. Data integrity can be addressed for instance by immutable, transparent availability of a hash of the data that is shared, for instance via a permissionless blockchain network. Asymetric encryption is another solution. Another way of creating trust is to register each participating node with a trusted registration authority. This requires extra functionality (and governance) which is not recommended.

Combinations of encryption and authentication of trusted nodes and data integrity is supported by protocols like Transport Link Security (TLS), using for instance eIDAS certified PKI certificates. This currently seems the most safe way to implement link security between two nodes of the network. Https can also provide link security, but does not involve trust in a client accessing a server. Therefore, https is considered more vulnerable.

6.4 Non-repudiation

Non-repudiation is the ability to proof that data has been send to with or received from a node. Both a sender and recipient need to be able to provide such proof, where the proof is immutable and identical. Non-repudiation is applicable to all data that is shared: linked event data and query/response data and especially:

1. relevant in case of conflicts caused by for instance accidents or incidents take place or activities are not performed in time.
2. Required as proof of compliance to regulation, e.g. particular data sets are shared with or made available to relevant competent authorities.

An immutable log and audit trail are a means for implementing non-repudiation. A log contains all data that is shared and an audit trail timestamps and initiating or receiving system components that triggered the sharing. An audit trail of a sender must relate to IAA: it may contain the identification of an employee that send an event or triggered a query. Immutability of a log and audit trail can be achieved by storing a hash on a permissionless blockchain. Such a blockchain is a type of notary function that can provide an immutable proof of the truth in case of conflicts. Another type of immutability is achieved by encryption of a log and audit trail, where decryption is properly organized.

An immutable log and audit trail can also be provided by a third-party notary function, called a 'Clearing house' by IDSA. It is upto organisations to set up functionality for non-repudiation and to clearly specify when it is required. Many IT applications already have this type of functionality, implemented by open-source software components.

6.5 Access control

Access control assumes that IAA is implemented. It implies that an authenticated employee of a data user can be given authorized access to data of a data holder. There are different ways to implement access control in IT applications like Role Based Access Contral (RBAC) or Attribute Based Access Control (ABAC).

A data user requires access to data of a data holder based on two assumption:

- Compliance: there is a regulation by which an employee of a competent authority requires access to a particular data set, based on an identification.
- Commercial: there exists a commercial relationship between a data holder and data user, by which one of them acts as customer and the other as service provider.

There are extensions to these two assumptions (grey-scales):

- Seamless goods flow: an enterprise provides access to additional data to an employee of a competent authority to improve risk assessment and improve seamless goods flow.
- Situational awareness: data is accessible (or can be analyzed applying Privacy Enhanced Technologies (PET)) to other organizations for optimization of (scarce) resource utilization. This is of interest in for instance traffic optimization and reduction of waiting times at hubs. Most often, a third party (algorithm) is used as a trusted service.

- Open data: data is available to anyone, IAA is not required. Open data can be implemented by specific APIs or predefined SPARQL queries.

Access control for compliance, seamless goods flow, and open data is expressed by a SPARQL query formulated by a data user. For instance, a compentent authority formulates its data requirements on the semantic model, publishes it as a SPARQL query, and refers to the SPARQL query that needs to be executed when requesting data. This SPARQL query can be the implementation of an eFTI data subset.

A SPARQL query for compliance and seamless goods flow ill always have an identification (for instance a UUID) that is provided by a data holder to a data user via an event. A container number and consignment reference number are examples of such identifications. As said, the competent authority has formulated and published the query and thus the required access is known. This is a type of ABAC.

A SPARQL query for open data can be more generic and not require a particular identification. In this case, the query is formulated by the data holder.

In a commercial relation, identifications to data that is accessible by a data user are shared by a data holder with that user in the context of a business transaction for a business service, i.e. a transport order. Thus, when a data user request for instance access to container details, the UUID (and container number) of that data is shared within a transport order for that container. There needs to be an agreement on which data will be accessible, i.e. the queries on particular Digital Twins or other types of objects need to be predefined in commercial relations. This can be on the level of a business service, for instance by formulating that a query on a Digital Twin like a container in a transport transaction always results in container size and type, tare weight, etc., namely all data relevant for executing a transport service of a container.

Any SPARQL query formulated on the semantic model can be transformed into another format, like XACML (XML Access Control Markup Language) or a JSON REST APIs. Organizations need to indicate the formats that have to be supported.

6.6 Summary

With respect to security, the following minimal functionality must be implemented:

- All participants must implement the necessary security measures to meet the EC Cyber Security Act.
- Transport link security between a node of a data holder and a node of a data user.
- Local access control based on IAA implemented by a IAM Registry of an organization.
- One or more trusted third parties that provides an identification to an organization (permission to participate) as a basis to trust their internal IAM Registry
- Non-repudiation must be supported at least for compliance and should be supported in commercial relations.
- Access control is based on identifiable SPARQL queries that are transformed into other formats required by participants.

At the time a trusted network of multiple Identity Brokers that can be applied by authorities (public) and enterprises (private), can be installed, authorization tokens must be applied.

7 DATA SHARING – ROLES, STANDARDS, FUNCTIONALITIES AND NODE

This section presents a data sharing deployment solution specifies the functionality of what is called a node: the behavior that any two participants of the FEDeRATED should implement, independent of a solution and technology that has been chosen. The concept is that of a data 'pull' based on available 'links', supported by access control and Identity, Authentication, and verification of Authorization (see the security perspective).

A federated network of platforms is a set of interoperable nodes, where each node behaves according to agreed protocols, implements the complete semantic model, data distribution, index, search (query), and service registry, and (conceptually) interfaces with IT back-end systems of a stakeholder.. A **node** is introduced for representing this implementation:

- A node provides the required **functionality** and **behavior** of an actor in its roles (data holder/-user)
- A node has an interface with internal IT system(s) of an actor, via for instance an **adapter**. An adapter can be complex in case multiple IT systems of an actor need to integrate with a node.
- By **registering** a node, that node is part of the network of all existing nodes. It enables an actor to share events (booking -, order -, and visibility events) with all other nodes.

First, the data sharing roles are introduced. Secondly, the underlying concepts for implementing the semantic model are discussed, leading into a decomposition of the concept of a 'node'. These nodes form a network.

7.1 Data sharing roles

Data sharing processes distinguish two roles as specified by the Data Governance Act, namely a data holder and – user:

- A data holder is a legal person or data subject who, in accordance with applicable Union or national law, has the right to grant access to or to share certain personal or non-personal data under its control. (source: Data Governance Act). A data holder is the same as a data provider in the context of data sharing.
- A data user is a natural or legal person who has lawful access to certain personal or non-personal data and is authorized to use that data for commercial or non-commercial purposes (source: Data Governance Act).

Each data user can act as data holder for other data users. Thus, a chain of data holders and -users can be created. In this context, a platform also acts as data holder. In its role of data steward and/or – custodian, a platform acts as data user to its users. Thus, a chain of roles is created.

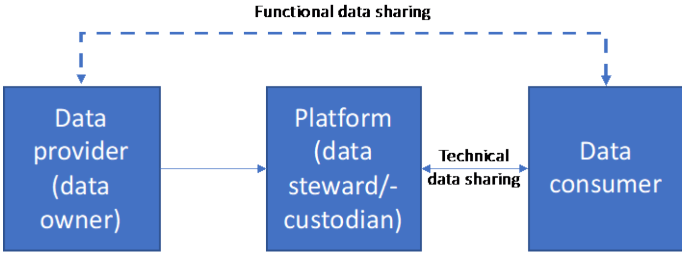

Functional data sharing
Data provider (data owner)
Platform (data steward/-custodian)
Technical data sharing
Data consumer

Figure 7-1: the different roles in perspective

7.2 Open standards for data representation and – sharing

7.2.1 Semantics and triple stores

The technology stack (see chapter 4 'language') is deployed by a triple store (next figure). It supports the various standards of the stack. It can import the complete semantic model and link RDF data to the proper concept and properties. Each instance of a Digital Twin, like a truck, can have a unique identification thus allowing to store data of that Digital Twin only once. This will be further elaborated. A triple store also supports SPARQL as a query language. Depending on the triple store, it can also have additional functionality like its own graphical query language (GraphQL of GraphDB). Triple stores come as open source, freeware, and as cloud services.

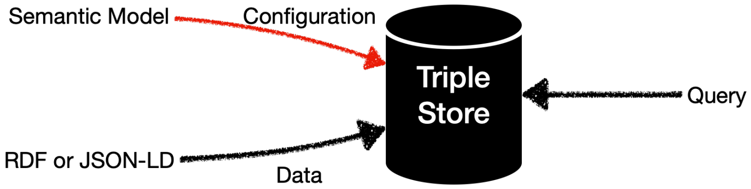


Figure 7-2: basic implementation

Applying the technology stack, brings additional functionality, namely the **unique (global) link** of any concept, where a concept can be anything, a box, a truck, a container, a document, or anything relevant to supply and logistic chai operations (see the concept). The link is used in IT systems governing physical flows; they are additional to any already known identification like for instance a container number or a license plate of a truck. These links can be printed as (two-dimensional) barcodes (also known as QR-codes) on physical objects like shown in the example.

The unique global link consists of two elements, namely a 'base URL' (URL -Universal Resource Locator or web-address, which is a unique identifier) and a software generated identifier, a Universal Unique Identifier (UUID).

Links provide (authorized) access to data of the concept. Sharing links implies a **data pull**, as required for instance to support of a targeting officer (see before). These links can be printed as QR-codes on physical assets. A combination of URL and UUID may reveal commercial sensitive information since a URL may refer for instance to a producer of high value products. Thus, part of the link is stored and shared between IT systems (the URL), where that part can be linked to an identifier used in the physical world. The identifier should preferably not contain any classification that would reveal information on the content or nature of goods.

Besides physical assets and boxes, also organizations, persons, locations, and infrastructural elements can have a unique link that can be used by IT systems. These unique links can refer to additional identifications like a port number for a terminal, geo-coordinates of the location (or area) where a pilot may board a vessel, or a chamber of commerce identifier of a legal entity.

The previous figure shows one triple store suggesting a central platform. The objective is to distribute this triple store to all stakeholders involved. The following figure shows the basic pattern that can be repeated. It consists of three triple stores that share links and can process federated queries.

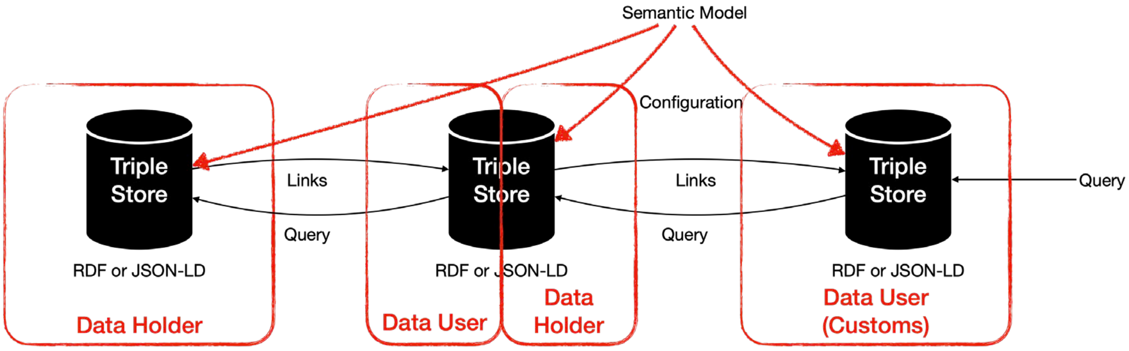


Figure 7-3 multiple linked triple stores

These triple stores all implement the same semantic model. The data is stored as RDF or JSON-LD. The data holder and – user roles are also mapped to this pattern, where a customs administration acts as data user. It shows that customs can pose a query to a known data holder, where this data holder can federate its query as data user to another data holder. Thus, customs can retrieve upstream data via browsing through links.

The previous figure shows that links are distributed from a data holder to a data user. The figure also shows a chain of data holders and -users, which can in fact be a network. An enterprise or platform acting as data holder can interface with multiple customs administrations and a customs administration with multiple data holders. There must be agreements on link distribution. Link distribution is the basis for query federation, where downstream query federation depends on implementation by all relevant stakeholders or the first known data holder, e.g. a platform like TradeLens, has the role of data user to all these downstream data holders. The following figure shows such a network, that includes query federation.

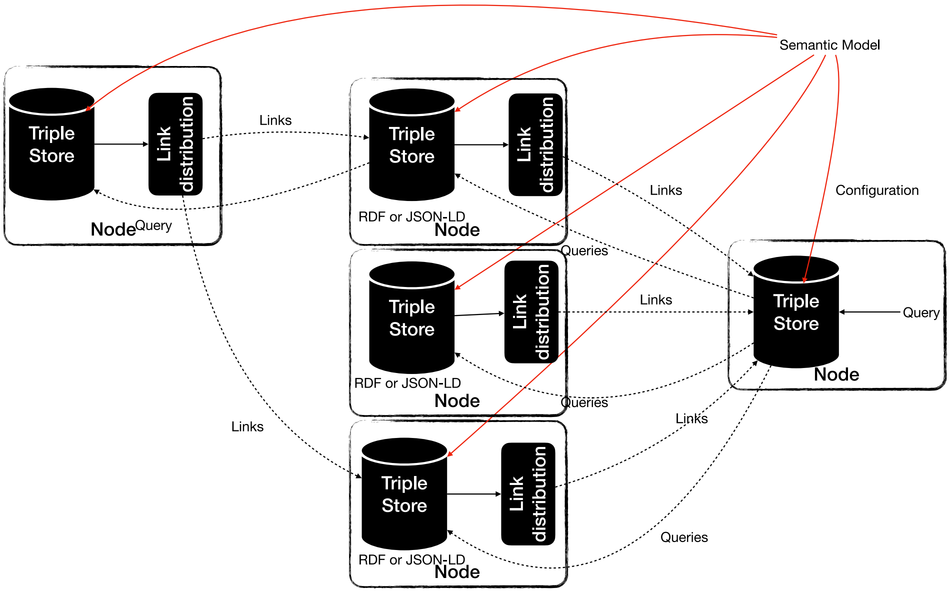


Figure 7-4: multiple linked triple stores

The dotted arrows show to which triple stores a link and queries can be distributed (the queries and links can of course also be distributed between the three triple stores at the right side of the figure). Each of these triple stores is implemented by an organization in its role of customer, service provider, authority, or even physical operator. They all have the same semantic model. Together, they constitute a network where all triple stores have each other's addresses. Queries are only distributed to a triple store if links have been received from that triple store.

The combination of a triple store and link distribution functionality is called a 'node'.

7.2.2 Data distribution – physical processes and event distribution

Where the previous figure shows one triple store, this store is in fact distributed like visualized in the following figure. This requires data distribution mechanisms.

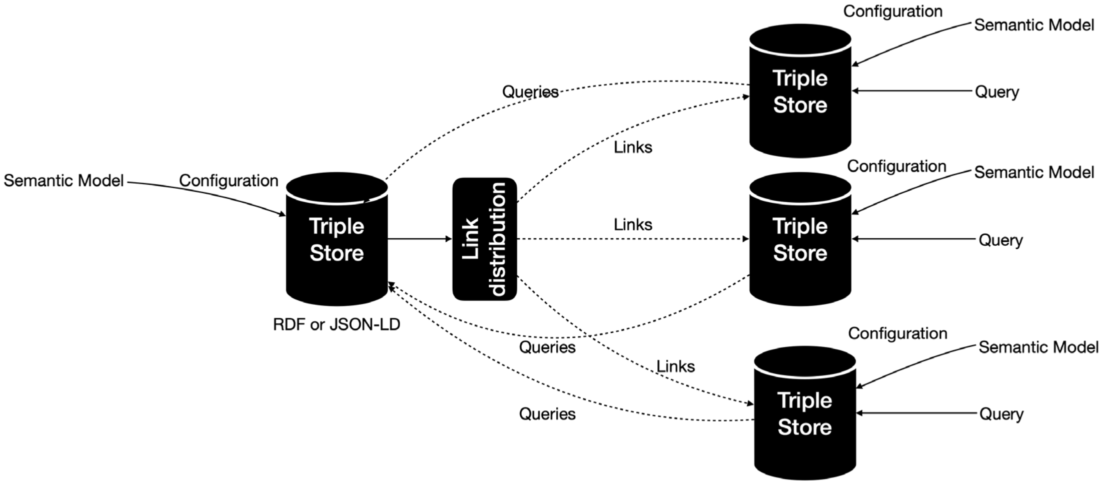


Figure 7-5 multiple linked triple stores

The dotted arrows show to which triple stores a link and queries can be distributed (the queries and links can of course also be distributed between the three triple stores at the right side of the figure). Each of these triple stores is implemented by an organization in its role of customer, service provider, authority, or even physical operator. They all have the same semantic model. Together, they constitute a network where all triple stores have each other's addresses. Queries are only distributed to a triple store if links have been received from that triple store.

Sharing links requires agreements on **link distribution**: what are conditions for sharing links. These will be elaborated later on.

7.2.3 Implementation aspects

Visualizing IT systems by a triple store, as done in the previous pages, is a simplified representation. Users will already have IT systems that have to be integrated with or behave as if they are a triple store.

We distinguish three types of implementations that will co-exist, namely:

- **Web of supply - and logistics data** – the semantic technology is fully implemented, and each stakeholder integrates with the semantic technology. Basically, two APIs exist, namely those for sharing links and those for accessing the data (via a SPARQL query, the so-called SPARQL Endpoint). By sharing links, the SPARQL query makes all potential data sources accessible. This type of implementation requires RDF plugins to existing IT systems, data transformation, and generation of unique links to optimal utilize the semantic technology.
- **(Platform) Application Programming Interfaces (APIs)** – there are still many stakeholders that implement APIs. Since we foresee that user (communities) must be able to specify their SPARQL query, each query will result in a separate API. Instead of a

generic SPARQL Endpoint, there will be an API with an endpoint for each query. The number of queries will increase by the number of users (or communities) that formulate their queries and require an API implementation. For interoperability purposes, these APIs should be generated from the semantic model(s).
FEDeRATED foresees that existing platforms or IT systems of users supporting APIs can integrate via these APIs with a semantic environment. It requires data transformation and link generation.

- **Standards (synonym: EDI – Electronic Data Interchange)** – many users deploy already standards like UN CEFACT, WCO, IMO, and EC CDM, all with their specific implementation guides. For migration purposes, a transformation between data represented by each of these implementation guides and the semantic model needs to be supported. The transformation shall be twofold, where the transformation from semantic (RDF) data to a standard implies a loss of functionality.

These different implementations require functionality. This functionality will be explained later in this document. These three types of implementations also refer to options that have to be supported by the presentation protocol, and potentially the security – and connectivity protocol.

7.2.4 An example – mapping stakeholder roles and triple stores

This example shows how a customs administration at a country of entry can access data by receiving links from stakeholders in the country of exit and - loading. It considers as example various actors that conceptually implement what we call a node with a triple store.

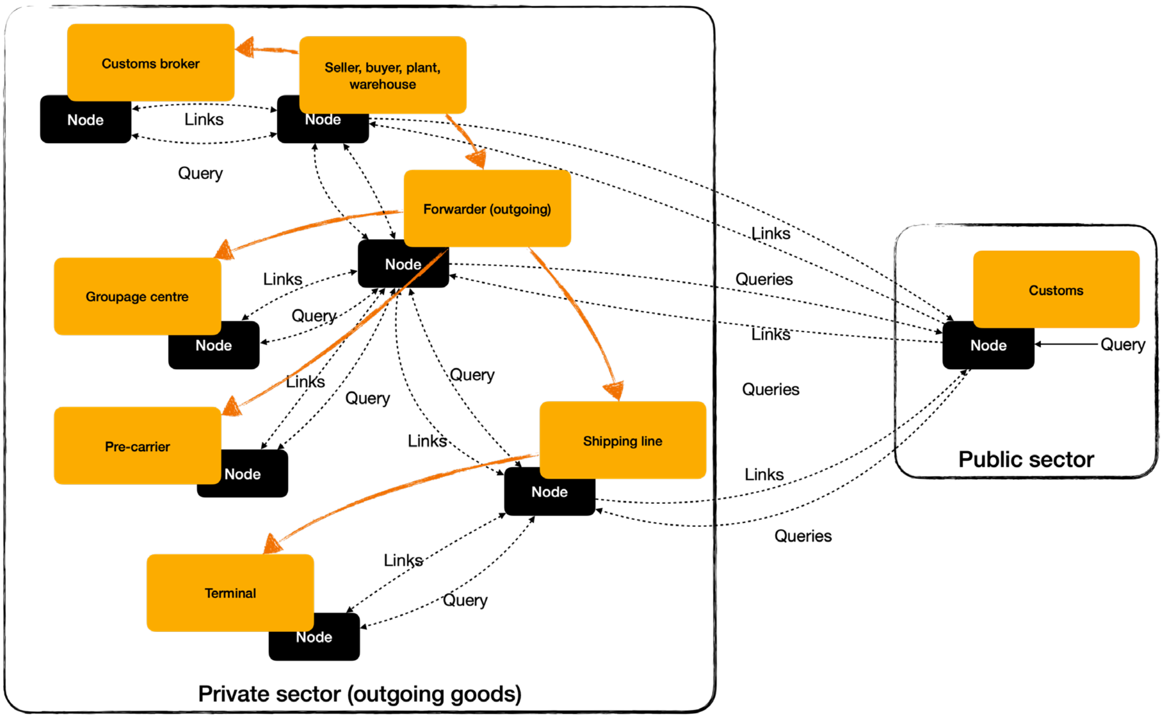


Figure 7-6: linked triple stores and stakeholders in supply and logistic chains (outgoing)

Figure 7-6 shows the following roles (concepts and definitions applied here are given by the semantic model):

- **Seller/buyer or plant/distribution centre**. This is the origin of the products that are moved from PLA (place of acceptance) to PLD (place of delivery). It can be based on purchasing (seller/buyer relation), Vendor Managed Inventory (from distribution centre to PLD), or stock replenishment (from a manufacturing plant to a distribution warehouse). These can provide product details for export and import declarations and the packing list.

- **Forwarder** (outgoing) – a forwarder manages groupage (outgoing), stripping (incoming), and transportation between a port and the hinterland (especially for LCL (Less than Container Load) transport) and is thus able to provide a stuffing list and status of the transport legs. A forwarder does not know the actual content of the goods, although this forwarder may have more information on the goods than a shipping line based on the goods description provided by its customer. A forwarder will receive a Bill of Lading (B/L) of a shipping line and share it with the forwarder for incoming cargo.
- **Shipping line** – a shipping line can only provide a load-/discharge list, complemented by transhipment lists, and pre- and on-carriage transport legs for FCL (Full Container Load). A shipping line never knows the actual content of a container, only a description of the container content (STC – Said To Contain), based on international conventions like The Hague-Visby – or Rotterdam Rules. A shipping line is aware of the country of origin and – destination of a container and can provide these types of links to a customs administration, thus providing details of container – and vessel track between ports or PLA/PLD.

  A shipping line will share a B/L with its customer and will inform a so-called notify upon arrival of the cargo in the port of discharge or will transport the cargo, i.e. container, to a place of delivery. The place of delivery can be a stripping centre or the final destination (e.g. distribution centre) of the content of a container.

Other stakeholders active in these types of chains do not directly provide data to a customs administration at entry at a country of exit. They act on behalf of others that contractually can provide links. The following stakeholder roles can be considered:

- **Pre- and on-carriers** – they can provide details of the transport legs between a port and the hinterland, but they are not aware of the country of destination or – origin of a container or the goods they transport. They will never provide links to a customs administration. They are either contracted by a shipping line or a forwarder. It can
- **Groupage – and stripping centres** – they can provide a stuffing list. They are contracted by forwarders and are not aware of destination of a container (groupage centre) or goods (stripping centre).
- **Customs brokers (export (and import))** – these act on behalf of a consignor, consignee, or their agent (forwarder). A customs broker at export has product details for the goods that are to be transported from PLA to PLD. These details are not shared with a forwarder in case the forwarder is their customer. The HS-code of the export declaration is based on the product description in line with export statistics and restrictions of the exporting country. For import, the objective is to have an HS-code that fits VAT purposes. For import and export declarations, a customs broker does not require the link of products to goods, given by the package list.

7.3 Data sharing options

This section introduced a triple store and link distribution as the core concepts of the architecture. These core concepts will have to be embedded in practice, where still existing solutions exist like messaging, APIs, and standards. These will be introduced here. They require additional functionality.

7.3.1 Data sharing paradigms

There are basically three ways of data sharing, namely:

- **Messaging (e.g. EDI)** – data is sent by a data holder to a data user. A data user must be reachable by a data holder to share a message. In most cases, these messages support business process integration of a data holder and – user.
- **Application Programming Interfaces** (APIs) – (sets of) APIs are used to access (GET) or share data (PUT/POST). In case of data access, a data user can retrieve data when it is required. A data holder must be reachable. Otherwise, the mechanism is like with messaging.
- **Linked Data** – links to data are shared amongst stakeholders. These links can be evaluated to retrieve the data.

Linked Data can also be implemented by APIs, although these could also be semantic APIs (so-called SPARQL endpoints). The functionality is in the endpoint, which allows to share and have controlled access to data. Linked data is selected by FEDeRATED (and DTLF) as a basis for data sharing.

7.3.2 Functional – and semantic data sets

In practice, Linked Data should be combined with messaging and other types of APIs. This can be done by analyzing the difference between a Linked (semantic) – and a messaging/API data set:

- **Functional data set** – each functionality has its own structure. These are the predefined settings of the management perspective. It is the messaging/API data set. For instance, there are separate 'transport order' and 'eCMR' data structure. They can be presented in any given syntax (EDI, JSON, XML, RDF). It is always clear what data should be provided to whom, based on functionality that has been agreed upon.
- **Semantic data set** – a data set is formulated as a (SPARQL) query on the semantic model. The query formulates the required output. This output can have options (e.g. there should always be 'cargo', but these can be 'containers' or 'goods') and specific constraints (e.g. if containers then attributes like container number must have a value according to a predefined pattern and container size and type must have a value from a classification). Again, the output can be any syntax, preferably JSON-LD or RDF.

A functional – and semantic data set are not disjoint concepts. A functional data set can be formulated as:

- A subset of a **semantic data set** fort the required function, e.g. expressed in the relevant part of the semantic model as RDF or JSON-LD. It means that data sets and queries are formulated as sort of standard for generic functionality. These are for instance the preconfigured settings (section 5).
- **Standard** or **proprietary** format, implying the semantics of a functional data set is expressed in another model than the FEDeRATED semantic model. The proprietary mdoel can be mapped to the semantic model. The standard or proprietary format can be used as data carrier. There will always be (an implementation guide of) a standard or proprietary format.

7.4 Operation and functionality

This section explains the way the architecture will be operating. It identifies the various functional components to explore how these can be combined into the concept 'node' introduced before. Details of the functionality are described at a later stage; this section shows the behavior.

It is split into two parts:

- **Data distribution** – distribution of links (event data) to all relevant nodes. This will currently be limited to events for supply chain visibility and links to additional data. The other phases of the choreography will be included at a later stage.
- **Data retrieval** – retrieval of data by a link that has been shared.

The installation and (de-)activation of a node is part of a technical implementation and will be described at a later stage. It is based on data sharing patterns that will be described first.

7.4.1 Data sharing patterns

Data distribution and – retrieval must consider four patterns:

1. **Data chain** - a chain of data holder/-user. An example is where a shipper provides a link to data to its LSP and the LSP outsources part of the transport a carrier. The latter carrier must have access to relevant data of the shipper (data retrieval) based on distribution of links.
2. (Logistics) **chain management** – one actor manages and coordinates activities of two or more others. An example is where an LSP decomposes and bundles orders of shippers for transport and distribution. Each service provider of the LSP performs its part of a logistics chain, e.g. a carrier to a hub and from a hub to the destination by another carrier (next figure).
3. **Data handover** – a data holder hands over data to another actor that becomes a data holder. An example is the case where a shipping line hands over the goods to some actor known as 'notify' in a port of discharge. Relevant data should also be handed over.
4. **Distribution** – links to data are distributed by a data holder to multiple data users. This is the case where multiple authorities in different countries or regions require relevant (links to) data of for instance a truck passing that country or region, including load and discharge events (trip or itinerary data).

Data handover seems to contradict data sovereignty and data at the source. However, a business service may be completed, any data that was at the source will be archived (this includes or may be only the linked event data that has been shared), and some other stakeholder will continue a logistics chains with relevant data.

Data handover reflects current processes which have a long history. Data handover can be avoided by implementing the data sharing solution. Instead of data that is handed for instance by a shipping line to a forwarder for incoming cargo, a link to data is handed over. This can also be handed over by the forwarder for outgoing cargo to that of incoming. It requires shipping lines to share any relevant transshipment events between an outgoing – and incoming port where in the latter case the cargo is available to a forwarder.

The patterns are shown by the following figure 7-7.

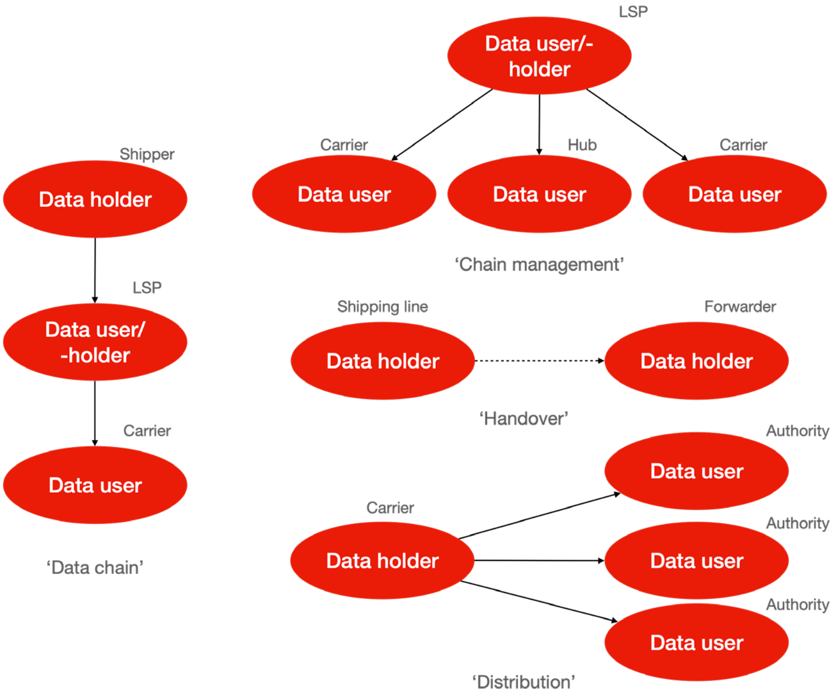


Figure 7-7: data sharing patterns

The arrows in the figure illustrate the distribution of links. The logistics roles are shown besides the data roles as an example. The handover process involves in practice more actors to assure that a shipping line hands over the goods to a proper notify and thus also the relevant (access to) data.

Combining the first three patterns makes clear that not all data will be handed over by one data holder to another. For instance, a shipping line coordinating pre-carriage, loading, and discharging goods, will not handover these details to a forwarder at import, but only the data it has received from its customer. This customer, which could be a forwarder at exit, could handover the data to the forwarder at entry as a proof for having data. A waybill supports this kind of functionality. The handover between a shipping line is not only based on this (waybill) data set, but there is also a payment aspect. This example is shown in figure 7-8.

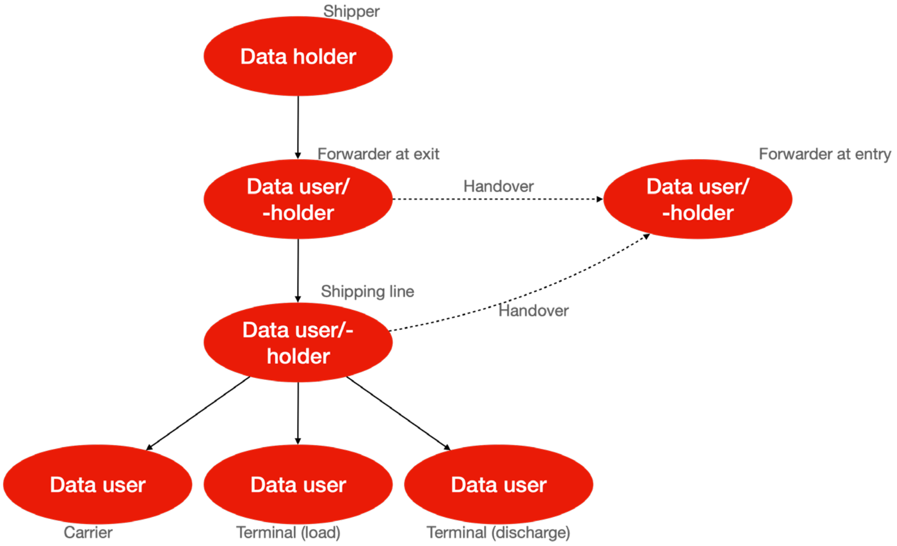


Figure 7-8: combining patterns – an example

Just a brief note on figure 7-8. Currently, the chain controlled by a shipping line can be decomposed via transshipment of goods during transport. Any changes or delays are not reported to a forwarder at exit, which means that a forwarder at entry might not be informed of these changes. This aspect might be solved by a data distribution environment, including handover. Shipping lines might also provide tracking information, but that needs to be available to (authenticated and authorized) data users. In air transport, delays can be accessed publicly.

7.4.2 Data storage – linking events

It is assumed that each actor has its own internal processes, organization, and supporting IT to make decisions. These decisions will result in the data sharing patterns.

For instance, bundling of shipments of different shippers into one larger shipment for transport by an LSP is an internal decision. It results in a chain where the individual shipments are bundled at a location for their main transport to a destination by a carrier. This could be air transport where bundling is at the airport or container transport via sea where bundling is at a groupage center. Another example of such a decision is trip or itinerary planning and chain composition.

Events, containing the links, are shared between any two actors, independent of the patterns these actors participate in. Those events refer to Digital Twins. Each event always has one data holder and one data user. Thus, each event is unique in the relation between two actors.

These unique events can refer to the same Digital Twin, e.g. the same container. Whenever a data user receives an event for a Digital Twin, it can share as data holder other events with other data users of the same Digital Twin.

From a Digital Twin perspective, all actors involved and the events they have shared can be retrieved. These events linking a Digital Twin to bilateral relations of data holder/-user are stored by each actor that has both roles for that Digital Twin. A network (or chain) of events for each Digital Twin is the basis for data distribution and – retrieval. This network/chain reflects the physical flows of Digital Twins based on outsourcing relations by individual actors as indicated before.

The implication is that each event has a unique UUID in a relation between a data holder and – user. Access to a Digital Twin is via evaluating its link in this bilateral relation. Let us take the example of the data chain pattern of a shipper, LSP and carrier. The shipper stores data of the 'goods'. An event with the UUID of the goods is shared with the LSP, that shares another event with the same goods UUID with the carrier. The carrier is only able to retrieve the 'goods' data via its LSP, it does not have direct access to the goods UUID since the shipper does not know the carrier.

Another example is that of 'chain management' where an LSP coordinates transport of goods by different carriers via a hub. The carriers are only aware of the location of the hub where they must deliver the goods or pick them up. The carrier dropping of the goods informs the LSP of its estimated arrival time, which is communicated by the LSP to the hub and the carrier that requires to pickup the goods. Any delays of the first carrier might cause cancellation of the second carrier and replanning by the LSP. A hub could also provide time windows to the LSP that coordinates these with the carriers. Note that these processes are currently implemented by direct contact between carriers and hubs, which might cause sub optimal synchronization of modalities/carriers and thus higher costs for an LSP and eventually its customers.

Data distribution constructs these chains of events and data retrieval uses them to access data of Digital Twins, locations, etc. An exception to these rules is access by an authority to a link, where this link represents a shipment file like an eCMR or eAWB. The link might be provided by a carrier to an authority, where the link stems from the customer of that carrier. Evaluation of all links shared via linked event data may take too much time (although it is expected to be faster than current practice). This requires further elaboration.

Note that events are associations between Digital Twins, locations, and business transactions. As such, events can also compose the data of an order.

7.4.3 Data distribution

Data distribution is about distributing ‘events’, where these events are user specified. The flow of events start by ‘generate user event’ from an existing IT system of a data holder (see following figure. The figure shows the two roles, data holder and – user, and an actor taking both roles for distributing new events based on received events.

The following steps are performed by a data holder:

- Validate – a user event is validated on its structure and content.
- Transform – a user event is transformed into ‘atomic events’.
- Store – atomic events are stored in the index.
- Distribute – the atomic events generated from a user event are distributed to a (or more) data user(s). Distribution of atomic events is based on rules that support the patterns shown earlier.
- Log – those atomic events that are shared are logged per combination of data holder – data user. Thus, if atomic events are distributed to multiple data users, multiple entries are constructed. Any internal event, i.e. events that are not shared between a data holder to a data user, don’t need to be logged.

Steps to be performed by a data user are:

- Receive – atomic events of data holders are received.
- Log – upon reception of atomic events, these are logged per combination of data holder – data user.
- Store – the atomic events are stored in the index. At reception and storage, atomic events may trigger the distribution of new events.
- Transform – atomic events are transformed into user events. These might be identical to the ones submitted by a data holder but can also be specific to a data user.
- Import – import of user events into the IT system of a data user.

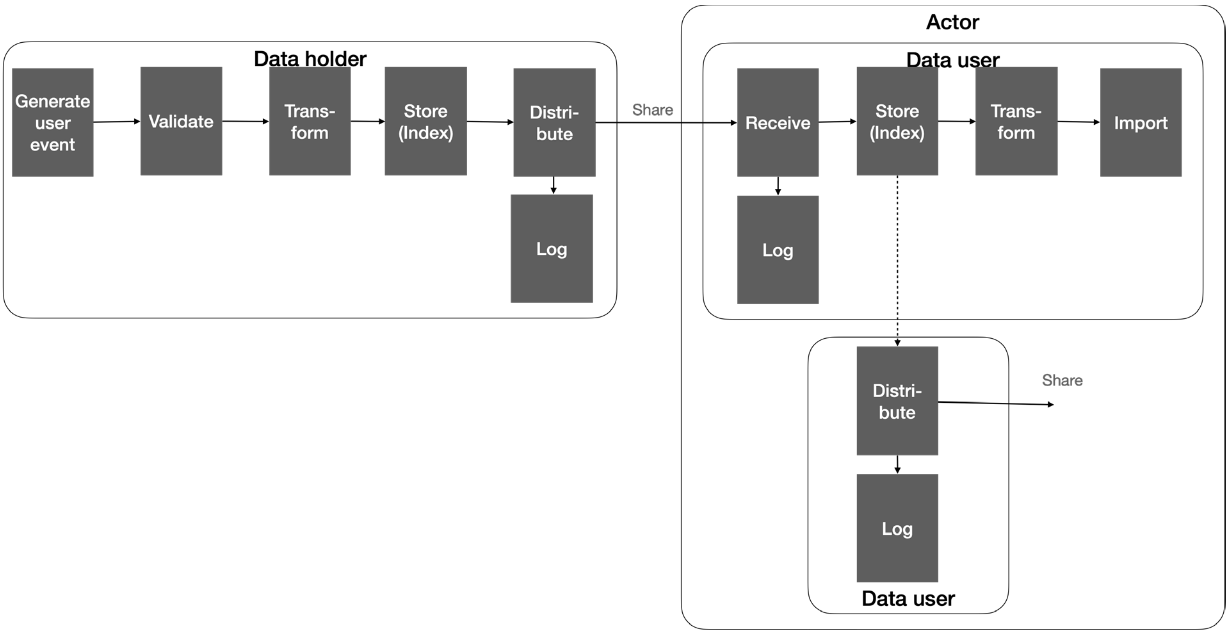


Figure 7-9: sharing events

There can be variations. For instance, reception of atomic events may involve the validation of those events. Another variation is where user events submitted by a data holder could directly be passed to a data user. The latter requires however a particular distribution algorithm for those user events and makes the solution specific to those data holders and – users that implement these user events (and thus does not contribute interoperability and general applicability of the solution).

A third variation is in the role of the index. As will be shown in the next section, the index can be used to query and access all received atomic events by a data user. The index is also required by a data holder for distribution of events as will be shown later. An illustration of the latter is where a loading event is only distributed to a customer and not to others. The index has registered that an order with a customer exists and event data (with expected dates) has been shared.

7.4.4 Data retrieval

In case of data retrieval, a validation against whether a link can be evaluated can be implemented. Each event has a unique identifier with identifiers of one or more Digital Twins, locations, and/or business transactions (document data sets). Such an identifier of a Digital Twin, location or business transaction can be a UUID that in combination with a URL can be used to access data of a data holder. A UUID can also be present on physical assets and goods for instance in a sensor or as barcode, besides other identifications like ‘container number’ or ‘license plate’. Not everyone that is able to read a barcode representing a UUID on a box may get access to the data stored by a data holder of that box.

To prevent that any unauthorized person can access data by reading for instance a barcode on a package, this barcode must only consists the IT identification on a Digital Twin should only consist of the UUID and not the combination of URL and UUID. A node that stores the UUID and is only accessible by employees with the proper credentials combines the UUID with base URL of the organization that provides the UUID. It serves as an extra security mechanism: the identifiers on Digital Twins don’t have any meaning.

The data retrieval flow should be able to forward an incoming query to access data to its original source. A data holder that receives a query uses its index to retrieve the data holder of whom he has received a link. Such a mechanism supports the 'data chain' pattern implementing the principle of data at the source.

The flow is depicted in a query – and response flow. The query flow might initiate a new query flow, based on data stored in the Index. This is called 'query propagation'. Instead of querying an IT back office system of a data holder, the query might only be forwarded to the (original) data source. It supports for instance the pattern of data chain, where the data of goods is stored by a shipper and requires access of a carrier via its LSP.

The assumption is that a query is generated by some actor, where it can be a person or software that generates the query. The query results into distribution of this query to all actors involved. The query can be anything like retrieval of container data represented by a UUID as Digital Twin. Each query requires search criteria that are valid to a user, like a container number in the previous example.

The following steps are shown:

- **Query** – processing an incoming query on the local store (index).
- **Send Query** – the query is send to the identified data holder, via the index.
- **Receive query** – a data holder receives a query of a data user.
- **Authenticate** – the identity of the data user is authenticated. Authentication is required only for employees of a data user sharing the query. In case the query is submitted by an IT system, link security is applied. The assumption is that a data user also implements link security for interfacing between its internal IT systems and the index.
- **Access** – the access policies to the query are retrieved. These formulate for instance the required response of the query.
- **Query propagation** – the access policies may imply that data is directly accessed from an internal IT system of a data holder and/or additional information needs to be retrieved form another data holder. Both or one option can be the case, depending on the data in the index: a query to an internal IT system and/or to another data holder.
- **Respond** – this should store that a response is given based on the query propagation. The respond may result in an error if all required responses are not retrieved in time. An option could also be to partially provide responses with an indication of the missing elements.
- **Validate/transform** – these are identical to data distribution. Before submitting the final response to the original query, that response will be validated.
- **Send-/receive response** – sharing the responses between a data holder and data user.
- **Import** – a function that enables the import of the response data set into internal IT systems of a data user. This is a type of transform.

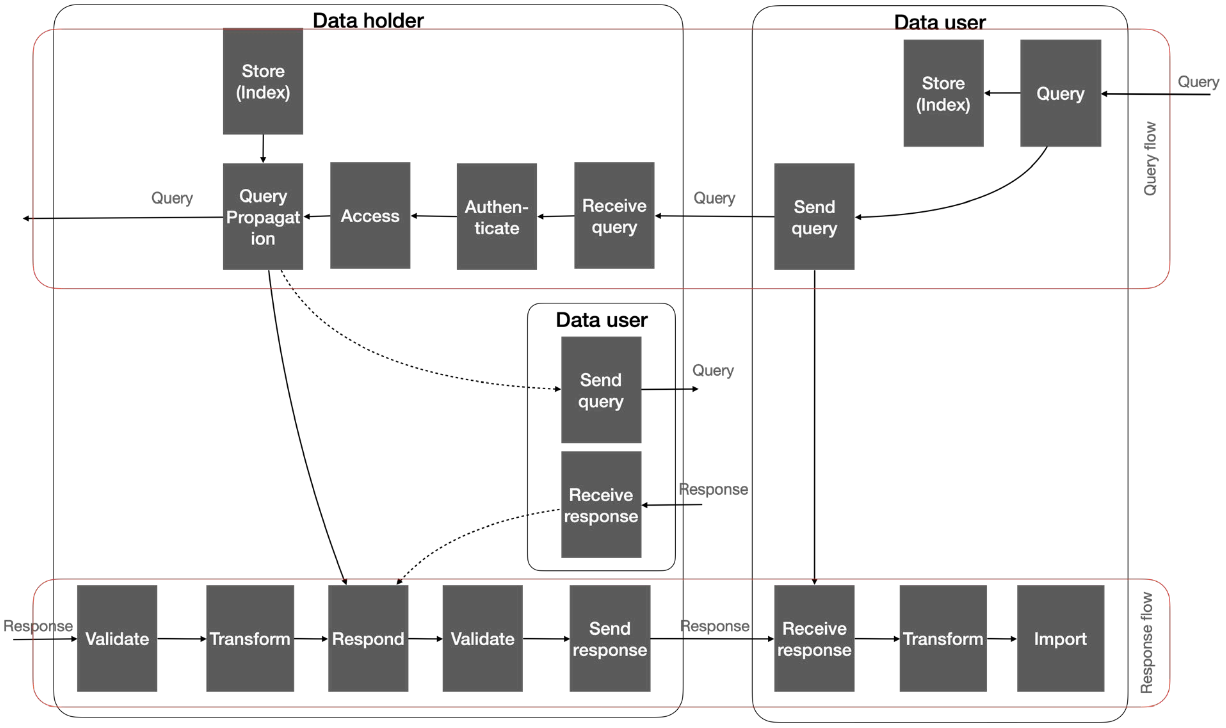


Figure 7-10: data retrieval

Figure 7-10 distinguishes a query and a response flow. A query flow is initiated by a data user on its index, top right of the figure. This can also be a query generated by 'query propagation' to a data holder, based on receiving a query. This recurring pattern of query flows is initiated by the query propagation function.

The previous figure does not show the 'log' function. However, every 'send' and 'receive' of queries has to be logged.

7.5 Node

7.5.1 Network of nodes

The previous section has identified functionality that has to be realized by components. The functionality must be implemented by each actor participating in a network. A **node** is introduced for representing this implementation:

- A node provides the required **functionality** and **behavior** of an actor in its roles (data holder/-user)
- A node has an interface with internal IT system(s) of an actor, via for instance an **adapter**. An adapter can be complex in case multiple IT systems of an actor need to integrate with a node.
- By **registering** a node, that node is part of the network of all existing nodes. It enables an actor to share events (booking -, order -, and visibility events) with all other nodes.

A federated network of platforms is a set of interoperable nodes, where each node behaves according to agreed protocols, implements the complete semantic model, data distribution, index, search (query), and service registry, and (conceptually) interfaces with IT back-end systems of a stakeholder.

The following figure visualizes the network of nodes. It shows that each node is registered to the network at its Registration Service. These Registration Services are interoperable, thus enabling a node registered by one Registration Service to share data with another node that is registered another Registration Service. The Registration Service is comparable to a DNS (Domain Name Service) server.

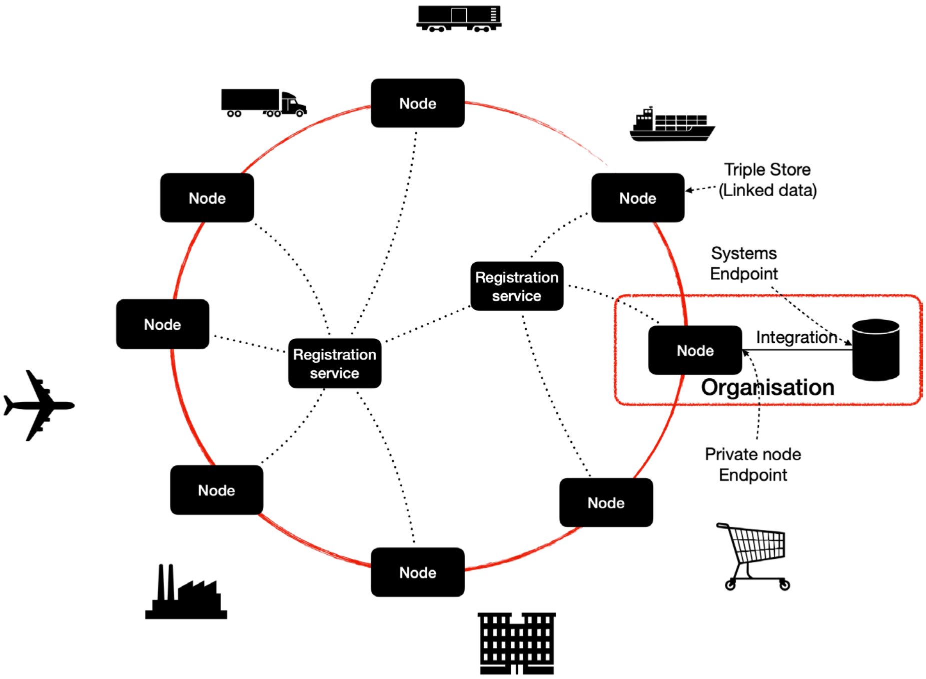


Figure 7-11: network of nodes

Figure 7-11 also shows that a node has a private endpoint. That endpoint is only accessible by an organization. They can query data stored by the node, based on IAA. In this case, access control is not implemented: all employees of the organization that implements a node that have the proper credentials have access data in the node. A node can also support more than one organization, in which case it must implement access control and IAA mechanisms.

An IT back-office system of an organization not only provides data to a node, but also can be accessed to retrieve data via its System Endpoint via its node.

7.5.2 Node functionality

As indicated before, any two nodes only share atomic events or queries. The functionality of a node can be grouped according to data – and processing aspects:

- **Audit trail** – everything that is shared between a data user and – holder is logged and an audit trail with timestamps and actions is constructed for implementing non-repudiation.
- **Send/receive** – the sharing of events, queries, and responses between any two nodes. Events will be stored at reception in a semantic store. Queries need to be handled according to the agreed access policies. Responses are the results of those access policies.
- **SHACL validator** – all input to a node will be validated. This can be input from an actor or another node. Validation is based on SHACL that is generated by the management perspective.
- **Semantic adapter** – all input by an actor to a node is transformed into atomic events and queries on those events. Transformation of atomic events and queries to formats required

by an actor as output is formulated as a query on the index of that actor. The semantic adapter is configured by the management perspective.

- **Semantic data store (index)** – a triple store containing all events that are shared with other nodes. The semantic data store can also serve as a log, depending on the implementation. If for instance events are only stored if they are shared, it can function as log. If events are however not shared and thus not logged, resending should be initiated by a data holder again, which would not be preferred. Simplest way is to have an indicator in the store that is set to 'shared' as soon as it has been shared.
- **Access Policy Evaluation** – this is like a XACML Policy Decision Point (XACML – eXtensible Access Control Markup Language). This so-called PDP needs to access two types of authorizations.
  - Commercial relation – a general rule is that access to data is provided in the context of events and UUIDs shared in a commercial relation.
  - Regulatory compliance – whenever a competent authority requires access to data, the access policy will be retrieved. This authority access policy is stored by so-called Policy Authorization Points (PAP) that are part of the management perspective. Access will only be provided if events with UUIDs are shared with the appropriate authority(-ies).
- **Event logic** – validating the allowed sequence of events, e.g. a discharge event is preceded by an arrival event, the arrival event by a departure event, and the departure event by a load event (figure Figure 2-2). Other event logic refers to the business process choreography (section 4.3).
- **Event distribution** – event distribution is based on 'atomic events'. This makes all nodes able to operate, independent of any requirements of an actor like specific user events. There are basically three rules for event distribution that need to be refined:
  - Data Holder events – a data holder can indicate to which data user(s) an event is shared. These are specifically booking – and (updates of) order events.
  - Data user events – these events are shared in the context a commercial relation represented by the existence of an order event. These are supply chain visibility events produced by a data user; they can lead to updates of order events by a data holder (see the pattern 'chain management').
  - Regulatory distribution – certain authorities must receive events, based on the regulation(s) they implement. These rules are stored by a so-called Policy Enforcement Point (PEP) that is part of the management perspective.
- **Query processing** – a query is on one or more links (UUIDs) of for instance Digital Twins, locations, and business transaction data sets. A query must be propagated to another node if a link has an associated event with a data holder, where the actor that received the query is the data user. If the latter is not the case, the query is forwarded to the internal IT system of the actor receiving the query.

  All queries that have been forwarded to other nodes and internal IT systems should provide a timely response. Query processing manages and combines responses. The type of response that is required, is formulated by the Policy Authorization Point of the management perspective.

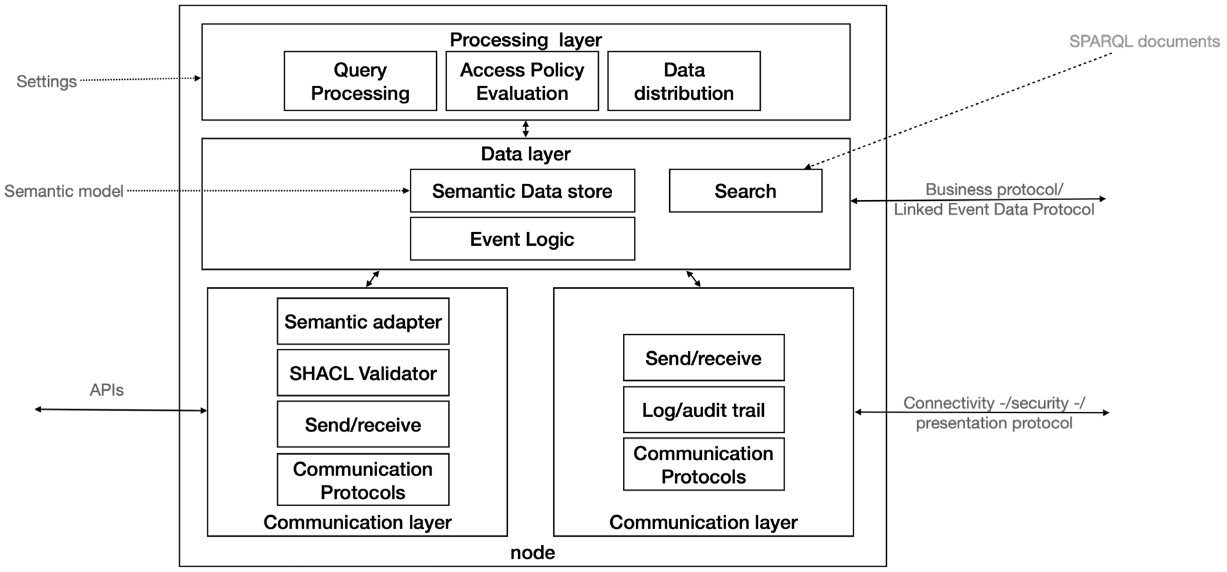
7.5.3 Node decomposition

A node acts as a type of gateway for an actor with all other actors. Therefore, it has functionality to integrate with internal IT systems and behave according to agreed protocols with all other nodes. This defines the distribution of the functionality as shown hereafter.

In this perspective, a node consists of three layers:

- **Communication layer** – the ability to share data with (1) another node and (2) an IT system of an actor. In the following figure, the latter is based on APIs. The communication layer implements one or more options of the connectivity -, security -, and presentation protocol.
- **Data layer** – the storage and validation of events and ability to formulate queries on these events (search). This layer implements the semantic model.
- **Processing layer** – handling queries and distributing atomic events. The processing layer could be extended with value added functionality like detection of differences in data values of the same Digital Twin provided by different data holders.

The distribution of functionality to each of these layers is visualized in the next figure. By providing APIs to IT systems of actors, the communication protocol and send/receive are determined. Whenever user events are received from an IT system, these are decomposed by the semantic adapter to atomic events that are stored by the data layer. They can also be offered directly to the distribution function, thus enabling sharing events with other nodes.



1. *Figure 7-12: layered functionality of a node*

Authentication is not shown and will not be discussed further. The assumption is that OAUTH2.0 is applied in case an employee interacts with a node. Since the nodes support event distribution and query processing, link security is considered relevant. An IT back office system of an actor will only be accessed via a node and not directly by a user.

7.6 Data distribution

Section 2.7 already indicated that sharing state information from the context of physical processes, or more specific ‘itinerary’, is the basis for data distribution. This is on the one hand in the context of data sharing patterns (section 7.4.1) and on the other hand driven by compliance to regulations. The latter is specified by ‘collection’ (section 5.3.2) representing details relevant to distribution of data and ‘authority restrictions’ (section 5.4) as part of the authority that needs to receive particular events for movements.

Both mechanisms need to be detailed. Whereas these mechanisms can be based on real-time retrieval of collection and authority restrictions from the Service Registry, they can also be supported by existing protocols

7.6.1 Data distribution mechanism in the context of regulations (B2G)

To be elaborated in an updated version

7.6.2 Link distribution in the context of supply chain visibility (B2B)

To be elaborated in an updated version

7.6.3 Protocols

In the various deployment environments, links will have to be distributed according to agreed protocols. These data distribution protocols can be deployed with various technologies, for various objectives, and in various environments.

An example is link distribution for implementing the eFTI regulation. So-called metadata like an itinerary and load/discharge activities needs to be distributed to the competent authorities, but is also shared in a B2B context. A rule for B2B distribution is the existence of an order as a transport contract. Links are distributed between a customer and service provider with for instance a publish/subscribe mechanism. The latter is configured dynamically, based on the existence of an order. The same data might be shared with a national authority that distributes the data to other Member States and makes the appropriate data available to its own competent authorities. In the latter case, the data distribution relates to access control and goal binding.

In dependent to its deployment, the link distribution protocols must be unambiguously published and can be implemented for example as follows:

- **Distribution** – there is an open (open: is publicly available for everyone), agreed protocol for distribution of links that can be implemented by each data holder. The protocol is published as executable - or a configuration of software, for instance a smart contract on a blockchain.
- **Publish/subscribe** – a data user may implement the distribution protocol as a subscription to links and thus generate linked data, unless a particular data user requires a (specific) subscription not supported by a distribution protocol. In the latter case, a data user should indicate this to a data holder.
- **Access control** – in case a Member State would like to hide complexity of governing authorities and regulations from business, access control to links can be implemented based on the principle of goal binding.

These are just examples of how a link distribution protocol can be deployed. Tooling might be required to transform a link distribution protocol into a deployment solution.

7.7 Access Policy Evaluation

To be elaborated in an updated version

7.8 Query Processing

To be elaborated in an updated version

8 THE PROPOSED ARCHITECTURE IN BRIEF

8.1 General

The concept is a federated network of platforms to enable data sharing in the logistic chain while providing interoperability harmonization between individual platforms. This concept allow for:

- smooth interaction between and among the different logistic chain operators and public administrations involved;
- enterprises to optimise the use of supply chains;
- dynamic planning to enable various ways of collaboration and optimize capacity utilization;
- recognizing existing (partial) systems;
- streamlining multimodal transport;
- decreasing or removing costs derived from lack of interoperability.

8.2 The Architecture

Based on these requirements the goal of the proposed Reference Architecture is to facilitate supply and logistic chain interoperability by providing the capabilities – within a infrastructure provision (data sharing grid) - to any data holder and data user the opportunity to do business with another i.e., engaging and fulfilling any kind of contract, execute transactions, providing (Value Added) service and complying with legal obligations.

The Architecture complies with the DTLF Building Blocks – Plug&Play, Federation, Technology Independent Services and Trust, Safe and Secure – the FEDeRATED Core Operating Framework and 37 Leading Principles. A core requirement is data at source.

The functional requirements of the Reference Architecture refer to the need for:

- a “common” language
- discoverability of data and their holders and users
- security for all participants
- access to all participants

The proposed Reference Architecture results in the need for data users and data holders to technically:

- Utilizing an IAA infrastructure.
- Applying a Service Registry (component) - allowing for a distributed development of data sharing according to common concepts and
- Applying a Node (node) – enabling open-source data sharing, potentially with predefined interfaces to support stakeholder roles. The Node:
  - always interfaces with Service Registry and can thus always be configured and
  - allows each stakeholder to share data independent of any existing platform while implementing an open source.

The Service Registry and the Node (components) will behave according to defined interfaces and protocols, ensuring safe and secure data sharing in the context of various EU legal settings. Semantics constitutes an integral part of these components.

Co-financed by the Connecting Europe
Facility of the European Union

8.3 Considerations towards deployment

The proposed reference Architecture comprises of some innovative features 9the new), aim to solve some existing interoperability bottlenecks. Deployment can only succeed when legacy (the existing) is recognized and can be dealt with rhrough stakeholder commitment.

The new versus the existing:

- The ideal deployment of the architecture is based on semantic technology. However, many existing systems and solutions are based on other technology like Application Programming Interfaces or Blockchain Technology (BCT). They interface via existing commercial and community platforms with APIs and messaging.
- The objective is to create a set of Technology Independent Services (TIS) enabling 'plug and play' by all stakeholders. When utilizing semantic technology for data sharing, TIS are completely configurable utilizing the Service Registry. This Service Registry can be applied also for other technology, but it will lead to a large amount of for instance APIs. The same is applicable to plug and play: organizations can implement so-called RDF plugins (or maybe duplicate their database to a triple store for external data sharing) or they need to implement (a potential large amount of) APIs.
- Whenever a platform supports TIS APIs, it is not required to have them interoperable. A user can call TIS APIs of any platform since they should all be identical. The latter will probably not always be the case since a platform will only implement a part of the TIS APIs relevant for its service. If, however, all platforms implement the same TIS APIs (or a minimal agreed set) according to IAA interfaces, there is no need for platform interoperability if these platforms agree on some type roaming agreement where one platform user can utilize another platform to share data with a user of that other platform.

Various steps have to be undertaken, not the least establishing collaboration with potential users, to complete this Reference Architecture to fly. Some necessary steps to be undertaken are:

- Development of prototypes.
- Pilot testing of the solutions (i.e., LivingLabs).
- More instructions and examples illustrating how the two components can be applied in practice.
- Development of a Governance perspective to provide more details on:
  - Service and operational governance (participation requirements, B2B/B2A service provision, marketplace development network future development)
  - Technical governance and security (Semantics, API development, System maintenance, Contingency Plan, Cybersecurity)
  - Data governance and digital identity (data ownership / privacy, digital identity and Service Registry, quality and traceability)

8.4 Utilizes the infrastructure provision (grid) - Value added services

The data sharing infrastructure provision is a basis for developing value-added service (VAS). Any VAS is outside scope of the infrastructure. It rather uses the capabilities of the infrastructure. It can run as a service on the infrastructure or can be implemented by one of the users of the data sharing infrastructure provision. A VAS is defined any (third party provided) service that utilizes links and access to data provided by those links to generate new data for one or more users of the infrastructure. ETA prediction, dynamic routing, risk assessment, and maintenance prediction are examples of a VAS. A VAS can be based on data analytics.

Conceptually, the infrastructure consists of nodes that are interoperable. A node supports the index, containing Linked Event Data that are received from or shared by a user with another user of the infrastructure, and a query for additional data by a data user to a data holder, based on links contained by the index. Each user (or user group, in which case a platform implements (party of) the node functionality) has its own node and each node contains different data. A node supports all required functionality for safe and secure data sharing, including functionality supporting non-repudiation (log and audit trail) and data integrity.

The capabilities provided by an infrastructure deploying node functionality are as described before: common language, data sovereignty, discoverability (Service Registry, search, index), data quality, event distribution, and query federation. These are capabilities that can be deployed by a Value Added Service (VAS).

A VAS can be provided by a third party that does not have a function in logistics operations but provides optimization data for logistics stakeholders and/or authorities (users of the infrastructure). A third party may provide a VAS as a service to users of the infrastructure and/or as a facility used by one of the users for its own optimization.

A VAS always provides a function to one (or more) user(s) of the infrastructure, where this user needs to provide the necessary links to the data (**event distribution**, which can be configure by the user to activate a VAS). A VAS will run a **search** on its node to collect all relevant data for calculating its output. Potentially, it requires **query federation** to assess the proper data set. The quality of the output data of a VAS depends on the **completeness** and **correctness** of its input data, which can be validated by a node (**data quality validation**).

A VAS can be provided to various users, based on their role, by applying the **semantic model** for developing its interfaces. It may be transport mode specific by using particular digital twins (e.g. vessel, truck), infrastructure (road, inland waterways, rail), and relevant features of the infrastructure (locks, traffic density, bridges crossing other modalities, water depths, weather conditions). These are all data holders. For optimization of a VAS, a copy of the infrastructure and its features might have to be stored at the VAS. These are all design decisions of a VAS. For instance,

- **ETA prediction** for arrival of a transport means at an intended location can be based on the location and speed of that transport means at a given time (an event with position and speed needs to be available), and the infrastructure composition (the available stretches of an infrastructure must be available). An improved ETA prediction can also make use traffic density, weather conditions, and estimated fluctuation of traffic density between the position and the location for which the ETA needs to be calculated.
- **Corridor management** is another example of a (more complex) VAS where agreed optimization rules provide an optimal speed for a transport means to reach its intended location in time (an ETA for this intended location is given). It requires details of more than one transport means, their position, their speed, and their intended destination. Corridor

management is applied by inland waterways[15] and can be applied where infrastructures for different modalities cross each other (e.g. water with rail or road).

In fact, corridor management is part of what is called **situational awareness**. The latter implies that individual stakeholders optimize their goals with awareness and considering the goals of relevant other stakeholders. It means that they need to share their intention, e.g. their itinerary details, and need to agree on decision rules. Each participating stakeholder can implement these decision rules with their own algorithms or use a service.

[15] See for instance https://www.masterplandiwa.eu

9 Adoption and deployment phases

The adoption and deployment of the Reference Architecture (genuinely translated into FEDeRATED technical specifications) can be identified in 5 phases. They are briefly specified as follows:

1. **Language** – this phase is about applying the semantic model for interoperability. Each individual pair of stakeholders or a Living Lab may decide on the interactions, their proposed sequencing, their implementation, etc., but they all stem from the semantic model. Deployment can be by messaging, (REST) APIs with JSON(-LD) or XML data, and a semantic endpoint.

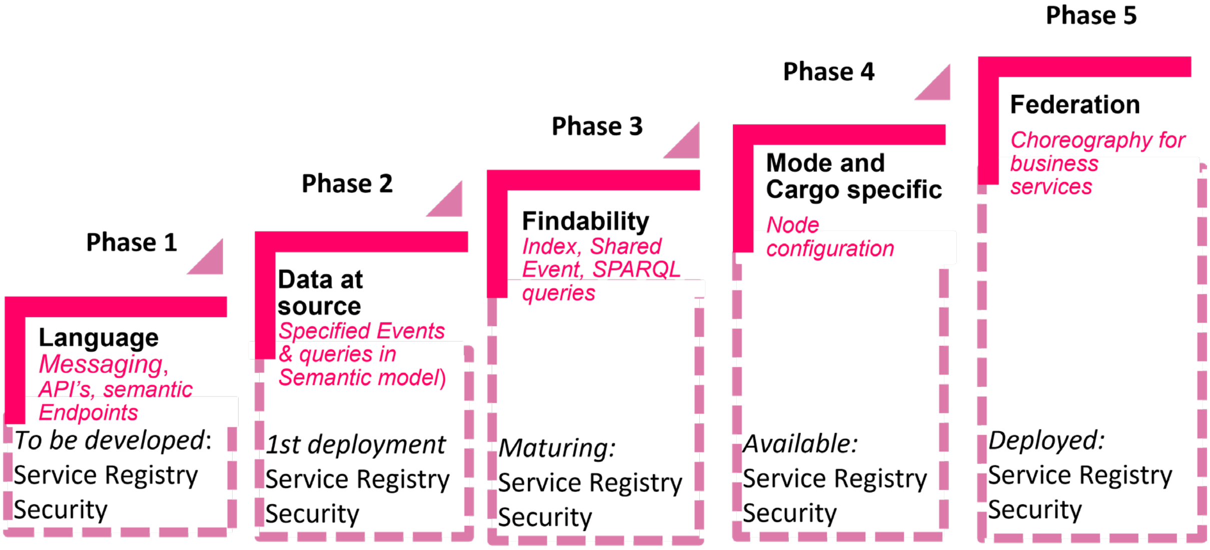
2. **Data at the source** – this phase is about specifying events and queries with the semantic model. They can be deployed according to the 'language' and the 'findability' phase.

3. **Findability** – this phase is about implementing the data pull mechanism. Each participant implements an Index, shares events, and implements SPARQL queries. Indexes share RDF data and can locally interface with existing IT systems of a stakeholder via for instance (REST) APIs.

4. **Mode and or cargo specific**. This phase is a node that is configured for a user group, community, or data space. Road transport implementing eFTI and eCMR is an example of such a data space, configured for particular functionality like (road) visibility compliant with (eFTI) regulations. Another example would be a node specific to transport of (bulk) commodities via sea.

5. **Federation** – this phase is full-fledged deployment of the business choreography for business services like transport, load and discharge, and storage. These are the Technology Independent Services. Each organization deploys its business services via the Service Registry and implements (relevant parts of the) semantic model and the business process choreography to support its business services. Thus, plug and play is implemented.

The phases are illustrated hereunder:



9.1 Language

This phase distinguishes between specification and deployment:

- **Specification** - the semantic model is applied for specification of data sharing between enterprises (B2B) and enterprises and authorities (B2A and A2B).
- **Deployment** – a data sharing technology fitting with capabilities of participants, for instance (REST) APIs with JSON(-LD) or XML data and messaging (XML, JSON). Participants may consider implementation semantic technology. Deployment not necessarily is about data at the source.

This phase requires deployment of PKI-certificates, preferably using TLS (Transport Link Security) for server-server authentication. Peer-to-peer connectivity can be implemented using a connectivity protocol of choice. Another option is the implementation of a platform, where each participant integrates with that platform.

This phase fits the current ways of data sharing, aligns the semantics of all interfaces for data sharing between participants, utilizes existing investments in interoperability, and can apply standard technology for peer-to-peer data sharing (**strengths**). Thus, it provides an **opportunity** for rapid adoption of the semantic model since existing technology can be re-used.

However, there are several **weaknesses** that lead to higher costs (TCO – Total Cost of Ownership for data sharing) like:

- These interfaces can differ per Living Lab or use case, which prevents interoperability between different use cases. Gateways can be developed
- Implementation by (REST) APIs requires version management and thus includes additional costs
- There will (potentially) be many (REST) APIs, each LL and use case can develop its own APIs.
- When applying the data pull principle for deployment, it can only be applied bilateral, i.e. between pairs of participants. To fully support a chain, it requires federation of a query to a data source, which requires additional functionality to existing IT systems of participant.

The TCO is higher in comparison with the other phases for adoption and deployment.

The solution also has **threats** like:

- Especially when a platform is used for integration, this platform may aim for scaling, i.e. connecting a large number of users, and thus increase (platform) competition.
- Each LL or use case can use existing tools that import the semantic model and enables them to construct their own model supporting their data sharing requirements. This will potentially lead to different versions of the semantic model, since participants will include extensions and make changes.
- The participants of a LL or use case lack knowledge of semantic modelling in combination with a potential lack of logistics. The learning curve for applying the semantic model may be too steep.

To address the first threat, participants may decide to use existing technology for peer-to-peer data sharing. The Dutch eGov Logistics applies for instance Corda technology, that also offers non-repudiation and link security. Connectors supporting peer-to-peer data sharing according to the International Data Space Association architecture can also be considered, but the architecture requires some central components for a Service Registry (called 'data broker' in IDSA) and. Non-repudiation (called 'clearing house' in IDSA).

To address the last two threats, a first version of the Service Registry can be developed as a tool that implements governance procedures for the semantic model, hides complexity of semantic technology, and supports the generation of (swagger) REST APIs for a LL or use case, where each participant can include its endpoint. The tool should be easy to use and support business analysts in formulating data sharing requirements.

To reduce the number of APIs and variants in specifications, Industry Associations and Regulatory bodies may specify interactions with the semantic model, where these can be re-used in LLs and use cases. This approach is specified in the current version of the FEDeRATED architecture. One could

9.2 Data at source

The extension of the previous phase is that events and queries for a data pull are specified and deployed with (REST) APIs. The same technology as with the previous phase is applied, expect the rules for specification differ from that phase.

This solution has the same **strengths** as the previous one. The **opportunity** of the previous solution can be a **weakness** since it requires mechanisms for sharing event data, which may not yet be supported by existing IT systems. The main **opportunity**, however, is the support of supply chain visibility by this solution, utilizing existing IT systems and solutions.

The **weaknesses** and **threats** of this solution are identical to those of the previous phase.

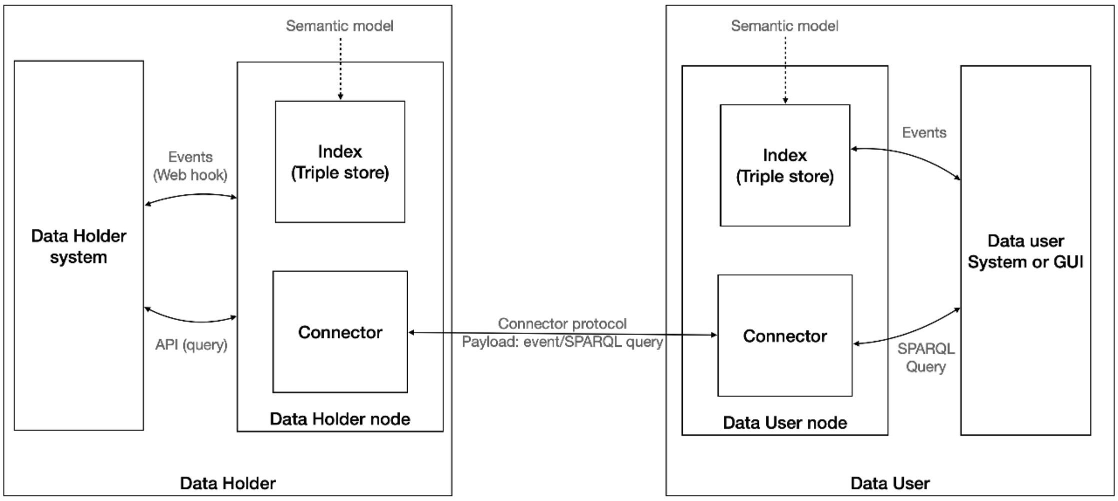
In this phase, the Service Registry requires specific functionality to specify 'events'. Those are called 'user-events': they combine more than one atomic event of the semantic model into an event that has meaning to participants in a use case.

Like in the previous phase, Industry Associations and Regulatory bodies may specify relevant 'events' and 'queries'.

9.3 Findability – implementing the semantic model and Index

Findability implements the data pull principle with an Index for each participant. This is shown as 'node' in the next figure (see also the FEDeRATED architecture). Participants of a LL or use case apply the semantic model for specifying 'events' and 'SPARQL queries'. Between any two implementations of an index, RDF data and SPARQL queries are shared with semantic technology. The Index fully implements the semantic model by means of a triple store, thus enabling data holders and – users to formulate their queries on the index.

The following figure depicts the implementation:



The implementation requires a semantic adapter to support integration with IT systems of a data user and – holder. The interface with an IT system of a data holder or – user is specified by the semantic model. SHACL validation is required to ensure data quality; the SHACL is specified by the semantic model. An additional function is required to map UUIDs (Universal Unique Identifiers) of concepts to identifiers used by data holder and – user IT systems.

The previous figure basically shows a peer-to-peer solution, but a platform may also be used to support the functionality. Data sharing between indexes of multiple organizations maintained by the platform is via the triple store.

The same type of security is required like for the first phase. However, onboarding of many stakeholders in case of a peer-to-peer implementation may require additional security features like OAUTH2.0 or similar mechanisms for verifying credentials of users.

The **strengths** of this solution are:

- Configurable. It is completely configurable to a LL or use case that decides to apply the data pull principle. Event structures can be specific, SPARQL queries can be supported.
- Local interface. Each participant can have a local interface with its node, which requires limited amendments to internal IT systems.
- Standard technology. Use of standard technology (open source and/or freeware) for a node.
- Single endpoint. Each participant has one endpoint for all queries, implemented by its node.

These strengths reduce the TCO, although they come with **weaknesses** like:

- Complexity of local interface. Local REST APIs must be aligned with SPARQL queries by a data holder. This decreases flexibility of the solution. This may imply that the number of REST APIs for local interfaces may increase by the number of SPARQL queries to be supported.
- New technology. New technology needs to be implemented in the domain of a participant.

- Lack of interoperability. The interoperability between LLs and use cases is not guaranteed since each LL/use case can specify its events and queries.
- Local interface – semantic adapter. The semantic adapter may require additional functionality for implementing a local interface with an IT system. Data structures of internal interfaces may not be identical to the structure provided by the semantic adapter, code values may differ, etc.
- Query federation. The query federation mechanism needs to be implemented for optimal application of the 'data at the source' principle.

The **opportunities** are:

- Error reduction. There is no retyping of additional processing of data by a data user, which reduces potential errors.
- Data sovereignty. The data source, i.e. the one that stores the original data, always decides on data access by a data user; only data users that have received an event can access the data.

The **threats** are identical to those of the previous phase. They have to do with knowledge of semantic technology, specifying events, and supporting (complex) ad hoc queries formulated by data users. These need to be supported by internal IT interfaces that might require additional complexity. This issue needs further attention.

9.4 Mode and/or cargo specific

This phase is a node that is configured for a user group, community, or data space. Road transport implementing eFTI and eCMR is an example of such a data space, configured for particular functionality like (road) visibility compliant with (eFTI) regulations. Another example would be a node specific to transport of (bulk) commodities via sea.

The **strengths** of this solution are (in addition to these of the previous fase):

- User requirements. The solution meets particular user requirements and is tailored to these needs.
- Recognizable. The solution is recognizable by all stakeholders in the community or data space.

These strengths reduce the TCO, although they come with the same **weaknesses** as in the previous phase, with the addition:

- Single modality and/or cargo type. There is a lack of interoperability with users in other communities/data spaces.

The **opportunities** are on top of these of the previous phase:

- Network effect. The node can be applied by all users of the community and thus potentially enable data sharing for all stakeholders in road transport.
- Data sovereignty. The data source, i.e. the one that stores the original data, always decides on data access by a data user; only data users that have received an event can access the data.

The **threats** are identical to those of the previous phase, with the exception that knowledge of the semantic model is hidden to users. However, they still have to be able to support complex, ad-hoc queries.

9.5 Federation – Technology Independent Services and Plus & Play

Federation is the implementation of the business process choreography by events and queries supporting business transactions for business services. The business process choreography may differ per business service type (e.g. one might have a different choreography for 'transport' and 'storage'). These specify the Technology Independent Services (TIS) that are the local interface between a node (see Findability) and an IT system of a stakeholder.

Since a node has predefined interfaces, these can locally be integrated via for instance REST APIs with an IT system of a participant. The relevant part of these predefined interfaces differs per participant. It depends on the business services of that participant. For instance, a carrier providing container transport services will not be able to transport solid bulk like sand or grain. The local configuration of the interface is called 'plug and play'.

To support plug and play, the Service Registry contains per business service type the minimal data requirements of all identified interactions in the choreography. By selecting its business service type and specializing it to its business, a participant defines its capabilities (and requirements). These business service types and their data requirements are published, enabling any potential customer to discover a service provider. This needs further specification in a next version of the FEDeRATED Architecture.

There are different ways to specify and deploy a choreography, namely:

- Predefined. There is a (set of) predefined choreography(-ies) that is implemented by a node and can be configured locally by a participant.
- Flexible. New choreographies can be developed and configured in a node for its deployment.

Both are feasible by using the 'node' for sharing events and queries since the interactions and their data requirements of a choreography will be mapped the mechanism implemented by the previous phase. In addition, a node needs to have event logic. Event logic is developed to support a particular choreography.

The implementation of this functionality requires full-fledged security (IA, non-repudiation, and link security) for rapid on-boarding.

The **strength** of this solution is that it meets all requirements formulated by the Digital Transport and Logistics Forum. It creates an open and neutral data sharing infrastructure for supply and logistics, available to all stakeholders. It is the freight part of the Mobility Data Space.

The **weakness** is its complexity. In case a participant wants to implement the functionality, it requires knowledge, must develop new functionality, and implement new technology. This weakness can be addressed by providing downloadable software that can be implemented, integrated via REST APIs with IT systems, and deployed via for instance Docker container or Kubernetes. Another weakness is development of new procedures for on-boarding and potentially conformance testing of solutions. Development of new solutions by new entrants or existing integrators requires also clear, concise, and complete specifications.

The expectation is that federation will provide completely new **opportunities** for development of new applications, will contribute to sustainability, and create a market for innovation.

Threats are basically in intermediation of existing solutions (i.e. platforms and community systems) with their existing business models. There is no requirement for using more than one platform or community system if the TIS are available and deployed by those solutions. One can simply call the

TIS locally and have business transactions with all others involved. Another potential thread is disintermediation of the role of Industry Associations. Standardization of TIS and plug and play does not require any mode specific solutions.

Regulating bodies: standards and the phases

Of course, data requirements in the context of a regulation can also be developed and deployed along these phases. For instance, eFTI data requirements can be specified by using the semantic model and implementing it with APIs (phase 1 – language), but the data pull mechanism with events and queries can also be applied.

In such a case, those queries will not change frequently; changes will depend on changes in regulation. Thus, these queries can be implemented by logistics service providers and their customers by means of an access policy. A query of an authority must have the proper control information to select an access policy.

It is recommended that authorities specify their data requirements in terms of data that is shared by enterprises (B2B). This implies that mechanisms used for data at the source in B2B are also used for B2A, i.e. the events need to be distributed to the proper authorities. Authorities 'piggy back' on B2B data.

Many regulations are currently supported by specializing UN CEFACT or WCO models and formulating interaction specifications by (hierarchic) subset of these model. These interaction structures are provided in human readable formats, for instance paper, spreadsheet, and html. If one fully requires reaching federation, data requirements of these regulations must be expressed in terms of the semantic model applying data at the source. Such separate adoption and deployment phases need to be explored by regulating bodies. If regulatory bodies do not change, gateways need to be developed for interfacing to these standards.

More than one semantic model

It may be the case that a use case, LivingLab (LL), Industry Association, or Regulatory body develops its own semantic model using semantic technology. There are two options:

1. Linking – the ontology supports functionality that is not yet part of the FEDeRATED semantic model. It can become part of the FEDeRATED semantic model via linking. This is done by the semantic model developed by the European Railway Association (ERA).
2. Matching – both ontologies are matched with each other, so data can be transformed from one ontology to another and vice versa. Since ontologies will have different design principles, the FEDeRATED ontology is for instance based on 'Digital Twins', 'events', 'business transactions' and 'infrastructure', there is most probably not a complete alignment. One ontology might be aligned and matched with a part of the (FEDeRATED) other ontology. This might be required for One Record that is specific to airway bill data.

Expressing LivingLabs (LLs) in terms of the adoption and migration phases

Preferably, each LL to one of the adoption phases. It will also illustrate where the LLs are in terms of scaling and on-boarding new users, which is expressed by a need to implement a Service Registry and (common solution for) IA. Also, the aspect of rapid on-boarding should be given.

FEDeRATED
NETWORK OF PLATFORMS

Co-financed by the Connecting Europe
Facility of the European Union