# Multimodal Visibility Service

**Specification**

**March 2024**

*www.federatedplatforms.eu*

FEDeRATED
NETWORK OF PLATFORMS

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

# Table of Contents

# Summary

The objective of FEDeRATED is to realize and validate capabilities described in the architecture. These support the FEDeRATED Vision and and the Master Plan. One of activities identified by the Master Plan and the architecture is that of 'service development'. This is about 'Technology Independent Services' as identified by the Digital Transport and Logistics Forum, Sub Group 2 (DTLF SG2). This document provides an example of such a TIS called a Multimodal Visibility Service applying the capabilities of the architecture.

By applying the capabilities, event data structures and event sequencing are given in this TIS specification of the Multimodal Visibility Service. The specification address four aspects of visibility, namely:

- Customer visibility – a customer is informed of the progress of physical activities.
- Compliance to regulations – this is about informing competent authorities of cargo movements relevant for their competence domain and providing them the capability (link) to access data. This is an example of the eFTI (electronic Freight Transport Information) regulation.
- Event federation – a value added service to generate new events at reception of events, for instance to inform a customer or a competent authority.
- Physical handling – this is the basis for generating events. It is about truck drivers, terminal operators, warehouse operators, etc. to generate events according to instructions they have received from their back office. Via event federation, these events are the basis to inform relevant stakeholders.

This is a first draft of a Multimodal Visibility Service used to develop the capabilities. Not all required functionality is yet implemented by these capabilities as indicated by the architecture. However, by implementing this service with the Minimal Viable Product (MVP) described in the architecture document already value can be added.

As said, this is a draft specification. It requires further validation in practice by pilots in Living Labs. The MVP configured for the most relevant aspects of this service, i.e. the events that can be shared, is available.

# 1 Introduction

## *1.1 Objective*

To specify a Multimodal Visibility Service that can be implemented with various technological solutions. Two potential implementations are the support of openAPIs (API – Application Programming Interface) and semantic technology as support by a 'node'. The annex will specify how the service is supported by a node.

## *1.2 Background*

The Multimodal Visibility Service is an example of a 'service'. This concept is introduced by the architecture. The architecture also specifies the core concepts that must be instantiated by a service, like:

- **State** – this specifies the data set that has been shared by any two organizations participating in a service. State data structures, i.e. the data that can part of a state, is a tree structure (named graph) on the semantic model.
- **State transition** – moving from one state to another triggered by an event that is shared. The architecture provides a template for a state transition.
- **Event** – a data set with basically links to additional data and user defined identifiers (like container number) shared between two organizations. An event has a tree structure on the semantic model. Transaction – and visibility events are identified. This specification is about visibility events.
- **Start** – and **end state** – each service must have a start – and end state and a so-called path from a start – to the end state. The path is specified by state transitions.

These concepts (and properties given by the architecture) are an ontological implementation of Business Process Modelling notation (BPMn2.0) choreographies. This document therefore shows choreographies.

A **service** like a visibility service is always used in a context, namely 'transport'. This is called a 'business activity'. A business activity also has a data structure as shown in the architecture. This data structure is centred around transaction events; these reflect order type of data sets, includes planning and visibility. A transaction data set for transport specifies the start state of a multimodal visibility service.

The implementation of the Multimodal Visibility Service is by its **Minimal Viable Product** (MVP) as specified by the architecture document. Semantic Treehouse, the first version of the Service Registry, is used for constructing event data structure and the start state. These are configurations for the node: RML (Rule Markup Language) for the data transformation and SHACL (SHApe Constraint Language) for data validation.

Although state **data structures** and business activity data structures can be developed by Semantic Treehouse, this is not yet done, since the node does not yet support event logic.

The node supports the so-called **eventAPIs** as specified by the architecture. Part of these eventAPIs is the synchronization of the start state. This reflects a publish/subscribe mechanism, whereby for instance a customer can subscribe to events.

The eventAPIs can be integrated with IT legacy systems. If an IT system is not yet able to process or generated events, i.e. it is not event-based, additional software must be developed to filter events out of other data sets. If event data is not yet available in IT legacy systems, a GUI (Graphical User Interface) of a node can be used.

Whereas a Multimodal Visibility Service is an example of a service (or **Technology Independent Service**

(TIS) as defined by the Digital Transport and Logistics Forum, Sub Group 2 (DTLF SG2)), the architecture also introduces '**service customization**': each organization implements those parts of a service that fit its business activity. This is the profile of an organization.

An example of customization of the Multimodal Visibility Service is for container transport by sea. The service is customized by selecting only 'container' as cargo and 'sea' as transport modality. Other combinations are also feasible, like transport of 'normal' -, reefer –, and dangerous cargo containers by sea, road, and inland waterways. This is up to each individual organization to select. Mechanisms for matching these profiles are given by the architecture.

A profile provides in fact less options than the business activity of a service. A profile is thus associated with a business activity data structure. It thus also has a data structure, which refers to the 'start state' of the visibility service and results in SHACLs for data validation. It can be specified with Semantic Treehouse.

Since the MVP of a node does not yet support profile matching, this must be done at design time. A simple extension of the MVP would be to include these so-called 'service agreements' (see architecture).

### *1.3 Value added services – event federation*

The architecture introduces so-called value-added services (VAS) that use events and (links to) data shared between organizations. One of these VAS is event federation: the ability to (semi-)automatically generate new events upon reception of events. Event federation is for instance a service provider informing a customer of progress and a competent authority of cargo movements in its area, based on for instance a load event provided by a truck driver.

Event federation utilizes the protocols identified by the architecture. It would help adoption of the solution since many manual tasks are automated. An operator (or a sensor) can generate an event (load, unload, border passing, etc.) that is automatically distributed to those that need the information. This is either because they are subscribed to it (via a shared start state) or there is a legal obligation e.g. entering, moving in, or leaving the area of a competent authority. The latter is for example eFTI (where eFTI is voluntary, but must be implemented in case an enterprise uses eCMRs or similar data sets).

### *1.4 Extensions to the service – upper – and lower ontologies*

The multimodal visibility service specified in this document is restricted by its functionality as specified by the semantic model. It can be extended with the following aspects at a later stage, either extending the semantic model or creating additional specializations (these are called 'lower ontologies'):

- Modality specifics – each modality will have its own way of operation, leading to potentially additional interface specifications. This additional specification may have to be supported by additional APIs or extensions to existing APIs for a modality.
- Cargo specifics – each cargo type will have additional requirements as to stakeholders involved and thus data sharing aspects. Additional APIs will have to be generated to support these requirements.
- Dangerous cargo – dangerous cargo will have its own data requirements, potentially also different per modality (e.g. (deep)sea, road, and inland waterways have the same classification, air and rail have different ones).
- Localization – each location (e.g. sea-, air- and inland port) may have its own data requirements that differ. For instance, seaports have port authorities and processes with piloting and tugging. Furthermore, localization can also be on the country level with different authorities governing

specific regulations.

- Business activity – the current focus is on ‘transport’. A multimodal visibility service for transport is supported. Since synchronization with other business activities like transshipment and storage/production is required, these business activities can be included later.

The previous implies, for instance, that dangerous cargo, bulk (dry or liquid like chemicals), and reefer transport are out of scope and can be included later. This will affect data sets that can be retrieved and operations that are required.

These specifics can gradually be developed as so-called ‘lower ontologies’ providing specializations for these topics, for instance to cover visibility of dangerous goods.

Any extensions may also require the review of event distribution rules, resulting in change and/or new rules.

### *1.5 Structure of this document*

This document elaborates the setup and interfaces of the common LL:

- Section 2 – overview of the functionality
- Section 3 - specification of multimodal visibility service in a bilateral collaboration of customer and service provider
- Section 4 - physical operation
- Section 5 - chain management
- Section 6 - compliance with regulations

This document provides the functional requirements for events and states; their representation by the FEDeRATED ontology is given separately. This representation can be used to generate openAPI code, SHACL (SHApe Constraint Language) for data validation and RML (Rule Markup Language) code for transformation between JSON structures and RDF.

# 2 Visibility

This section further analyzes visibility. It addresses the various aspects of visibility and shows the distribution of visibility functionality to different roles. This latter is as example; the objective is to show how generic functionality can be developed that can be applied by stakeholders.

This section specifies the functionality and shows its potential distribution to stakeholders. Implementation of the functionality is up to each organization. Part of the functionality can be implemented by existing systems and solutions; other parts can be implemented as separate functionality like with a node (see annex) and/or a platform. When this will add clarification, such implementation aspects can be mentioned in the text as examples.

## *2.1 Visibility requirements*

Visibility is a service provider making the progress of a physical activity performed for a customer available to that customer. This progress is performed by reporting on physical actions like ‘load’ and ‘unload’ or ‘stuff’ and ‘strip’. A customer specifies its visibility requirements via an order.

Service providers can outsource activities, split a customer order into multiple ones, or combine multiple customer orders. This is about construction of logistics chains and the relation between consignment(s) and shipment(s). A customer may want to receive detailed updates of the progress, i.e. for each leg in a chain per order.

Having multiple transport legs in chains and (logistics) networks requires synchronization of the legs. A next leg like a hub needs to be informed of arrival and departure of transport means for optimization of its capacity. Thus, ‘arrival’ and ‘departure’ are also required as part of visibility, where an enterprise coordinating these legs forwards these events between the legs, e.g. an ETA event of a truck is forwarded by a forwarder to the hub that has to unload or accept cargo from that truck.

Supervising authorities and infrastructure managers also want to have access to data of cargo and transport means in their domain. This can be a national rail infrastructure provider, a customs authority, or a municipality. Each of them requires access to this data from a regulatory perspective, e.g. safety and security. Since this may result in a large amount of data that is not always used, supervising authorities and infrastructure can use visibility as a means to access data. This is the data pull approach of FEDeRATED.

Thus, there are three perspectives to visibility:

- **Customer – service provider relation**: a service provider informing a customer of transport progress. Upon customer request, these updates may consist of updates on intermediate transport legs of a chain if a transport activity is decomposed.
- **Physical operation**: an operator reporting the progress of performing actual transport based on its planning. This planning of an itinerary can be updated during its execution.
- **Compliance to regulation(s)**: informing a competent authority of a transport activity and providing access to data required according to the regulation.

An example of the relation between these three perspectives is an estimated arrival at a location of a truck. Such an estimate gives an indication of an unload operation and a load on another transport means. This estimate arrival event generates an event to the home base of a carrier, which can inform its customer. The latter can be an agent that organizes transport on behalf of a customer and informs its customer.

Furthermore, an agent will have to synchronize transport legs, meaning that a next leg carrier can be informed of an expected load operation.

## *2.2 Visibility in its context*

These three perspectives can be supported by components, where not all organizations will implement all components (see further). The next figure shows these generic components that can be implement by various stakeholders in supply and logistics chains. 'Chain management' is introduced for leg synchronization in chains. As the figure shows, an ordering and planning component are included since orders will trigger planning and chain management includes chain composition. Furthermore, query handling is shown. It supports the principle of 'data at the source' and validates the authorization for accessing data.

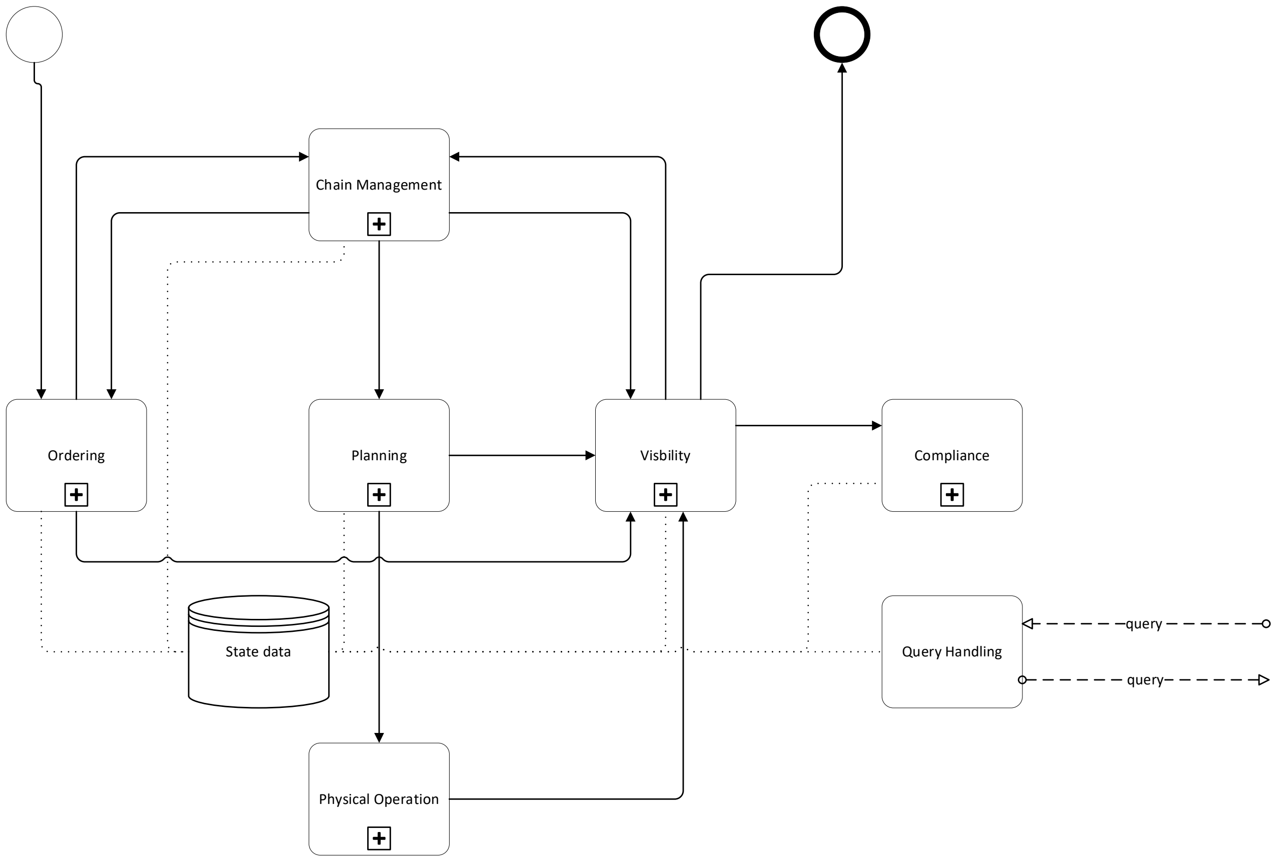


*Figure 1: Visibility components*

The functionality of these components is:

- **Ordering** – reception of customer orders and management of orders with service providers, including their planning.
- **Planning** – assignment of customer orders to resources (e.g. transport means) according to some planning algorithm. A customer may already provide a relation to a particular resource via for instance a flight number or a vessel/voyage combination for ports of loading and - discharge.
- **Visibility** – registering progress of a transport order, both with a customer and service provider.
- **Compliance** – providing a relevant link to additional data related to a resource to a competent authority for regulation(s).
- **Query handling** – authorizing access to data and/or federating a query to another data holder.

- **Chain Management** – composition, planning, and management of a chain, where management focusses on coordination of the different legs.

Ordering and planning are outside scope for this specification, including chain composition and - planning. The latter function requires additional input like available business services, quality of these business services, infrastructure – and/or resource related restriction (e.g. war situations causing delays), etc.

The output of these three functions is relevant to the other components, which is why they are visualized. The output and input of all components is shared in the database called 'State Data'. This stores all transaction – and visibility events according to state transitions.

### *2.3 Examples of distributing components to roles*

This functionality can be distributed across various stakeholders. We distinguish five roles:

- **Customer** – the role that requires transportation of cargo.
- **Intermediate** – the role that organizes the transportation on behalf of a customer or coordinates transportation of carriers. An intermediate can be customer of a carrier or of another intermediate.
- **Carrier** – the organization that is responsible for a transport leg.
- **Operator** – the (natural) person responsible for operating a transport means, e.g. a truck driver and captain of a vessel.
- **Competent Authority** – an authority that is responsible for supervision of (one or more) regulation(s) in an area.

This distribution of functionality shows how settings based on an order that is shared between a customer and service provider is used for propagating events in chains. Implementation of such a common order can be via for instance a subscription Application Programming Interface (API), where a customer indicates to a service provider that he wants to subscribe to visibility events of cargo offered for transport, e.g. a container.

The simplest case is where a customer orders transport of a carrier. In case of paperless transport, the operator provides upon on request a link to transport data to a competent authority. According to the principle 'data at the source', this data can be stored by the customer but can also be stored by a carrier in case that customer enters the data via a (web) application on a smart device. For remote inspection, a competent authority may receive links to data and request access to data data. The request is based on data requirements of a regulation like the electronic Freight Transport Information (eFTI) Regulation. Technically, a customer and carrier may agree to store relevant data with a third party like an eCMR provider.

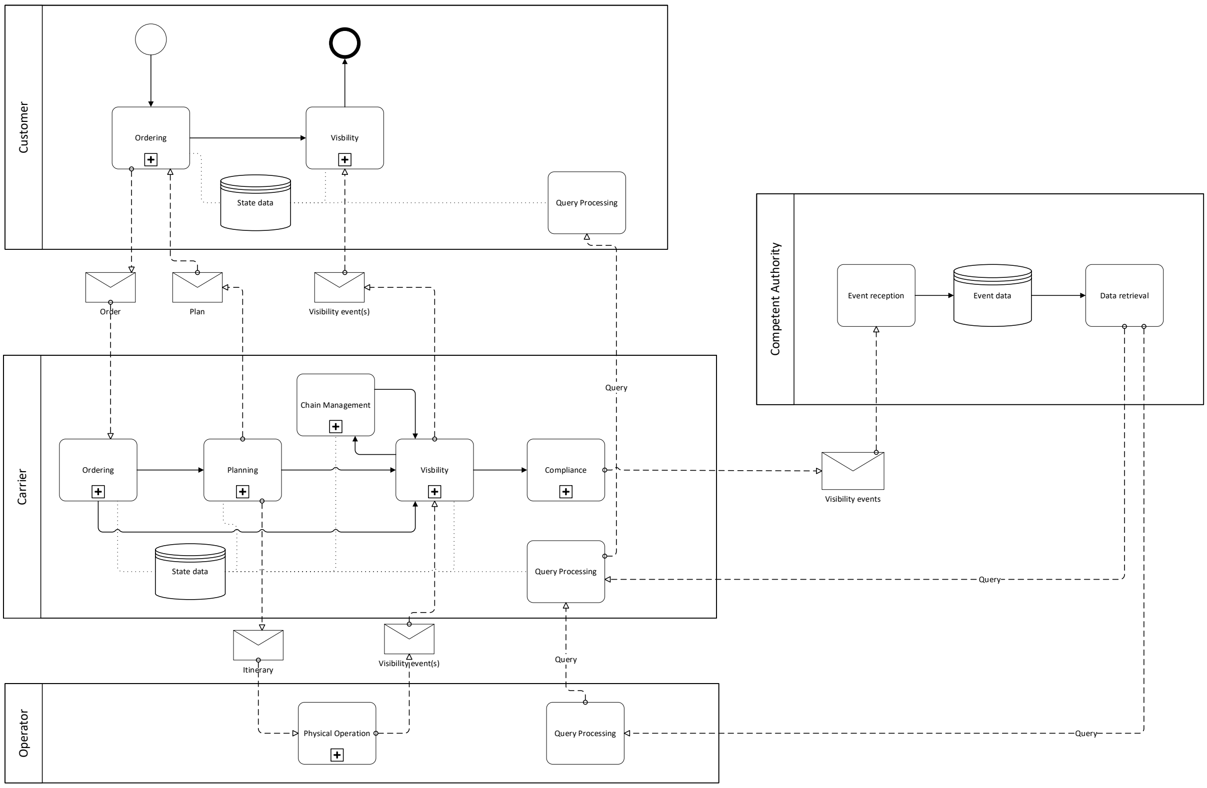


*Figure 2: Distribution of visibility functionality for a customer and carrier*

Only part of the functionality may be implemented by a particular role. For instance, a customer will require ordering and visibility for interacting with a carrier. However, a customer will also require a chain management function, but moreover to plan its own operation like production. This type of chain management and planning is outside scope of logistics and part of other application areas like industry and retailing.

The functionality of an operator can be implemented by onboard systems, like On Board Units for trucks, and/or applications on smart devices handled by an operator. This can be the same for other transport means like vessels, barges, and airplanes. An operator will have a minimal data set required for physical operation; access to additional data is via evaluating a link received from its carrier.

The planning functionality of a customer can be to simply record the plan received from a carrier as shown in the figure. The planning functionality of a carrier results in producing or updating an itinerary of a transport means, e.g. a trip of a truck, for loading and unloading cargo at ceraint locations. A carrier may also coordinate its itinerary planning with other carriers, which can be further specified by detailing the planning function. This latter will result in additional (planning) interactions between carriers.

In case of a carrier, the chain management functionality for visibility generates events to customers based on events received from an operator.

The previous figure can be extended by including an intermediate as shown in the next figure, where this intermediate does not have resource of its own that require planning. As the figure shows, a plan is received by ordering and shared with a customer according to chain management.

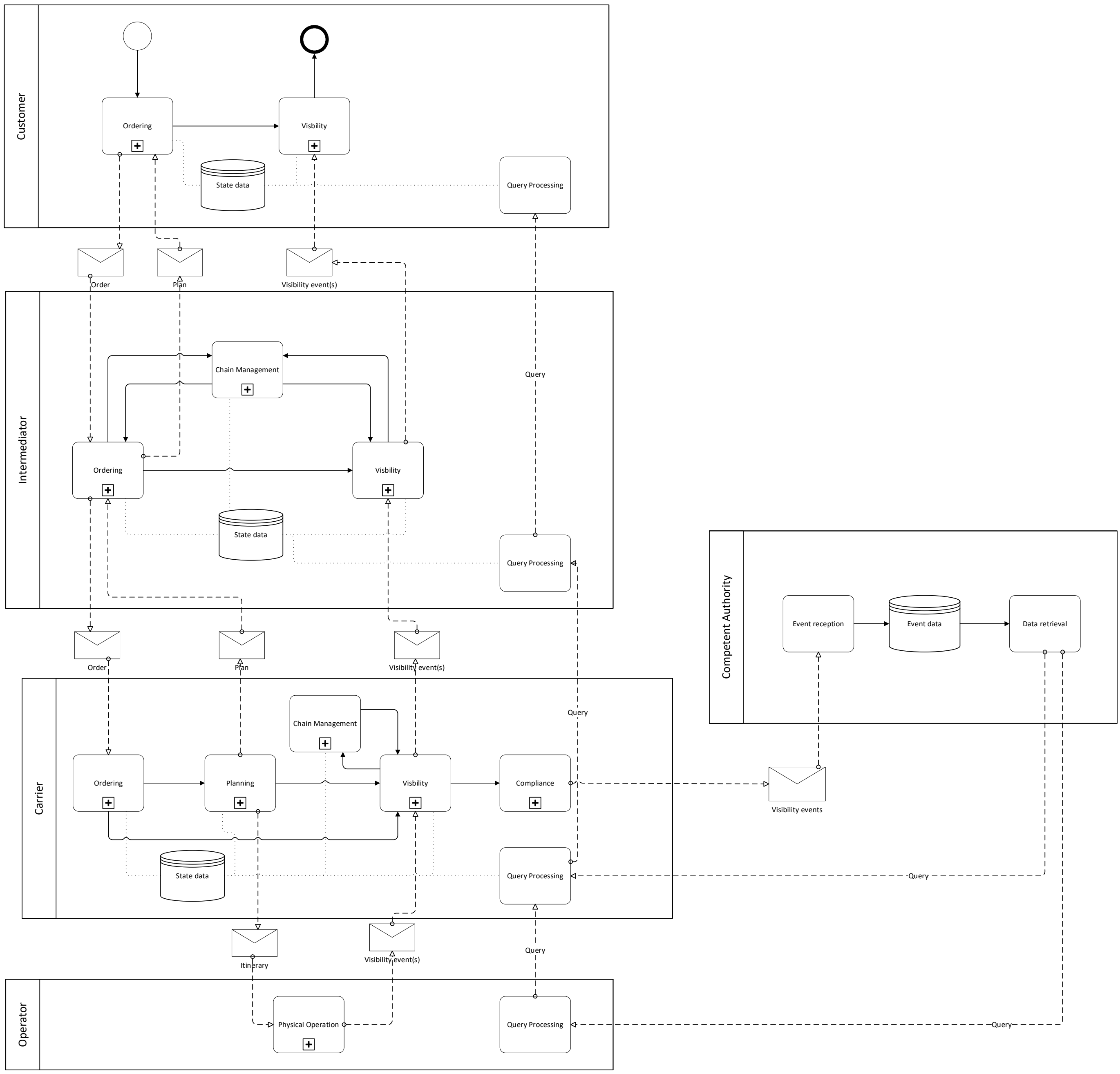


*Figure 3: Distribution of visibility functionality, including an intermediate*

As shown, the chain management functionality of an intermediate is about chain planning and coordination, where this document only focusses on sharing events with customers. Updating orders for next legs in a chain are not specified, these can easily be included as part of chain management.

The specification of ‘chain management’ in the context of visibility is based on the relationship between business transactions with customers and with service providers. It will show how events can be propagated in a chain.

# 3 Customer – service provider visibility

This section provides the specification of the multimodal visibility service for transport of any type of cargo, where a service provider informs a customer of progress of performing a transport activity on behalf of that customer. The multimodal visibility service specifies interaction sequencing by events in a bilateral collaboration. The next section specifies the processing of events by an individual actor in an organizational network.

The multimodal visibility service is given by a Business Process Modelling notation (BPMn) choreography. The data sharing module of the FEDeRATED ontology specifies the concepts of such a model; these concepts will be specified in this document. The concepts are:

- Interactions – these are used to model events.
- State transitions – these represent the sequence of interactions.
- States – a start, end, and intermediate status of the services.

The start state of the multimodal visibility service is the existence of a transport order. This state must be synchronized between a customer and service provider for event distribution. State transitions specify the event logic. Transport order (or also eFTI data in the context of the eFTI Regulation by a Competent Authority) or document data sets (like eCMR for road - or eBL for sea transport) can be retrieved by accessing a link shared at synchronization of the visibility service.

First, the choreography of the service is described. Secondly, the event structure is given, followed by the structure of the states of the choreography. Hereafter the state transitions are given, and the implementation of these state transitions by a customer and service provider are given, resulting in event distribution.

## *3.1 The multimodal visibility service*

This section presents the interaction pattern of the multimodal visibility service for transport. It is used to identify the various events that can be shared and their sequencing, where the latter is the input for event logic.

This interaction pattern is per order between a customer and a service provider. Any events shared for individual orders can be triggered by an operation at the level of a transport means. This is especially the case for 'ETA –' and 'position events' where the ETA and position of all cargo carried by a transport means is updated. This is not (yet) part of this specification.

The visibility interaction pattern (next figure) consists of activities by which events can shared between a customer and service provider, where these events can also be shared with an authority. For instance, a service provider submits a load event to its customer, followed by an ETA event. The following events are supported: load event, ETA event, Incident event, and Unload event. Their allowed sequencing is given in the following diagram, where circles represent a state (states: agreed order, in execution, completed, to be cancelled), rectangles represent data sharing processes (processes; start, ETA update, Position update, Incident/accident, complete), and envelopes with an arrow the initiation of an event by one of the roles (the blank role like 'LSP' for 'start' process) and the other the recipient (the grey role like 'customer' for the start process).

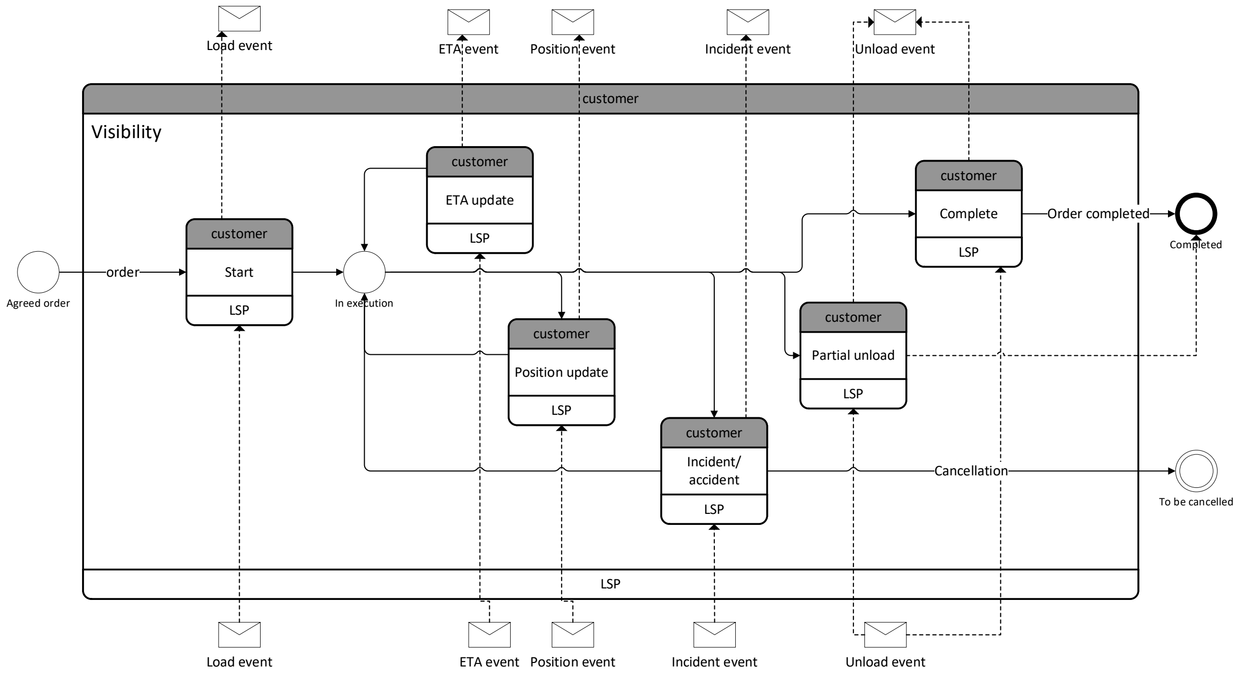


*Figure 4: Interaction pattern of the multimodal visibility service (specified as BPMn 2.0 choreography)*

The most basic example of interactions between a customer and LSP are by sharing a load event, followed by an ETA event, and completed with an unload event.

One of the processes of the pattern shown by the previous figure requires decomposition, namely the process 'incident/accident'. The start and estimated end of an accident or incident can be provided by an event, whereas the end time can also be given by a separate event (see the description of the events). This is not yet done and will have an impact on the event logic. However, the start or end of an accident or incident may not always be shared since a human may not be capable of doing this. A loss or damage can be reported by a single incident event.

The service specification shows that cargo can be unloaded in steps. This is by introducing the state 'partially unloaded'. The incoming load event is processed by the appropriate transition when the pre-condition of that transition is met. Thus, whenever an unload event is received, the pre-condition of both events is validated to decide on its result. Of course, partial unload can be related to a partial load, for instance transport of cargo with multiple trucks or trailers from between two locations. Whereas a partial unload is visualized in the choreography, the partial unload is not shown but can easily be included (leading to an extra state transition).

The states represented in the interaction pattern relate to 'transport' as the movement of cargo between two locations. These locations have different names for different modalities:

- **(Deep)sea**. The locations are Port of Loading (POL) and Port of Discharge (POD). This visibility pattern refers to loading, departure, arrival, and discharge of cargo in these ports, where the events refer to the port area. Each port will have more detailed events referring to business services of third parties (on behalf of a port authority) taking place in those port areas, e.g. tugging and piloting.
- **Air**. The locations are the airport of departure and the transit or destination airport of a flight. A flight is comparable with a voyage of a vessel, a trip of a truck or a path of a train. A flight has a slot at an airport; flights are managed and coordinated by air traffic control.
- **Road**. The locations are the Place of Acceptance (PLA, the place where the cargo is taken over by

a carrier) and the Place of Delivery (PLD, the place where the cargo is handed over by the carrier).

- **Rail**. These are the stations where the cargo is loaded onto or in a railway wagon and the station where it is handed over. A railway wagon is part of a train that has a path on the (inter)national railway infrastructure. National paths are assigned by a national Infrastructure Manager; EU paths are assigned via Railnet Europe in coordination with national Infrastructure Managers.

The next pages specify the events with an event distribution mechanism, potential queries, and event logic.

All data sets will be expressed as SHACL constraints to the semantic model and configure the semantic adapter.

### *3.2 Event structure*

Conceptually, each event of the multimodal visibility service has the following structure (see also the architecture):

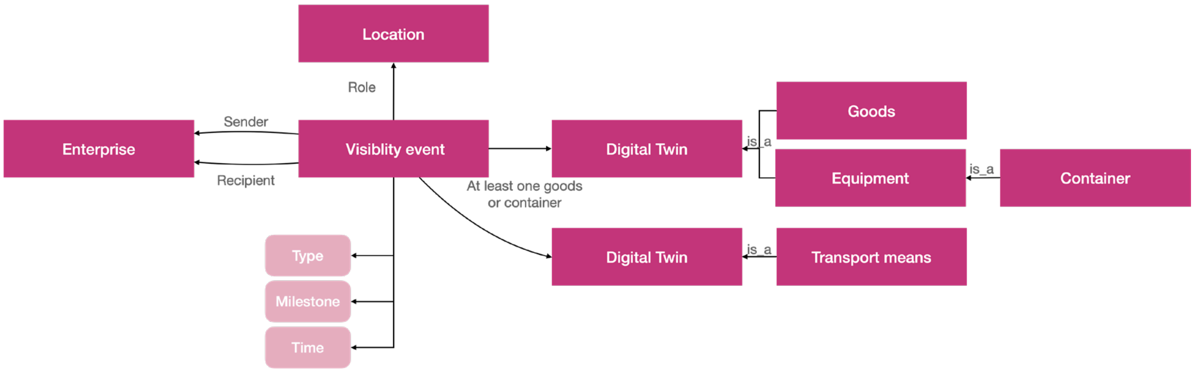


*Figure 5: Conceptual structure of visibility events (load, unload, position, accident/incident)*

The figure shows that an event represents an association of Digital Twins (at least one 'Goods' or container and a transport means) at a location with a role. The role can be Place of Acceptance (PLA), Port of Loading (POL), or any other relevant to the visibility service. An event is of a type, where the type refers to its function in the choreography. Types are for instance 'loading', 'unloading', and 'position'. A type has a specific value of 'milestone' for creating (start) and ending (end) an association. Visibility events will always have the time 'estimated' (ETA event) or 'actual' ((un)loading, position) and are send by an enterprise in its role of 'service provider'.

This results in the following structure for visibility events that will be expressed by the ontology:

| Visibility events | Load | ETA | Incident | Position | Unload |
|---|---|---|---|---|---|
| UUID (event) | ✅ | ✅ | ✅ | ✅ | ✅ |
| UUID (sender organisation) | ✅ | ✅ | ✅ | ✅ | ✅ |
| UUID (recipient organisation) | ✅ | ✅ | ✅ | ✅ | ✅ |
| milestone | start | start/end | start | start | end |
| estimated date/time | | ✅ | ✅ | | ✅ |
| actual date/time | ✅ | | ✅ | ✅ | |
| External reference | ✅ | | | | |
| Reference type | Document data set | | | | |
| Location | | | | | |
| Location role: place of acceptance, place of delivery | PLA | PLD | Position | position | PLD |
| UUID (location) | ✅ | ✅ | ✅ | ✅ | ✅ |
| Cargo (at least goods or equipment as a rule; can be multiple) | ✅ | ✅ | ✅ | | ✅ |
| For goods - UUID (goods) | ✅ (o) | ✅ (o) | ✅ (o) | | ✅ (o) |
| for equipment - UUID (equipment; can be multiple)) | ✅ (o) | ✅ (o) | ✅ (o) | | ✅ (o) |
| UUID (transport means) | ✅ | | | ✅ | |
| Digital Twin - transport means - (truck/vessel/airplane/barge/train) | | | | | |
| UUID | ✅ | | | | |
| Transport means ID | ✅ | | | | |
| Transport means ID provider | ✅ | | | | |
| Transport means Nationality | ✅ | | | | |
| Transport mode | ✅ | | | | |
| Digital Twin - goods | | | | | |
| UUID | ✅ | | ✅ | | ✅ |
| Number of packages | ✅ | | ✅ (o) | | ✅ (o) |
| Type of packages | ✅ | | | | |
| remark | ✅ | | ✅ | | |
| Digital Twin - equipment - (container/trailer/wagon) | | | | | |
| UUID | ✅ | | ✅ | | ✅ |
| Equipment ID | ✅ | | | | |
| Remark | ✅ | | ✅ | | |

An incident (or accident) is represented by three events, namely the actual start, the estimated end, and the actual end. These events can be shared separately; they are applied to calculate the delay caused by any of these events. They can be shared with at least two events: the first (milestone = start) indicates the time at which an accident or incident occurs with the estimated end and the second (milestone = end) the actual completion. Note that accidents or incidents cannot always be generated, since a human may not be capable of signaling such an event. Any delays caused by a traffic jam impact an ETA and can be reported as such.

As the previous table shows, an incident event may cause loss or damage to cargo, i.e. goods or equipment. This should be indicated when detected. There can be different types of incidents like damage, loss or theft of cargo. Accidents are also processed as incidents; they might only result in delays.

The previous table shows that five types of transport means can be given, one per transport mode. Thus, the transport modality indicates the type of transport means. The identification of a means of transport is assigned by an authority, that may be a national authority in for instance the case of license plates of trucks. In the case of an airplane, it is recommended to use the flight number as identification; a flight number indicates the movement via air from one airport to another using an airplane. This is not correct but will do for the moment.

The table shows that three subtypes of equipment can be used, namely containers, trailers, and (railway) wagons. Any additional subtypes could be included. Goods are identified via the types of packages: all packages of the same type are grouped. Either equipment or goods are given as cargo. In the case of equipment, different subtypes of equipment can be provided.

In case the event does not have a reference to cargo (goods or equipment), the visibility event is applicable to a transport means. It may for instance give the ETA of a transport means for arrival at a location (note that

A load event may have a reference to a document data set (optional), which can be the customer order reference. The document data set differs per modality. Road for instance uses a CMR data set, air the Air Way Bill (AWB) and sea the Bill of Lading (B/L). Such a reference may also be considered a reference to a customer order:

- For a customer -service provider business relation, the transport order can be applied.
- An authority also does not require this reference, since it can search on other criteria like 'transport means ID' (license plate of a truck, vessel code, etc.) and equipment identification (container number, license plate of a trailer, wagon number, etc.).

A CA will at least receive load and discharge events.

### *3.3 Data structure of 'state'*

The data structure of the states and interactions (events) is specified by a prototype tool. Therefore, the data structure is flexible. However, for event logic, a data structure must be known, which imposes rules for specifying states and events.

The following data structure is required for event logic (see also the architecture document):

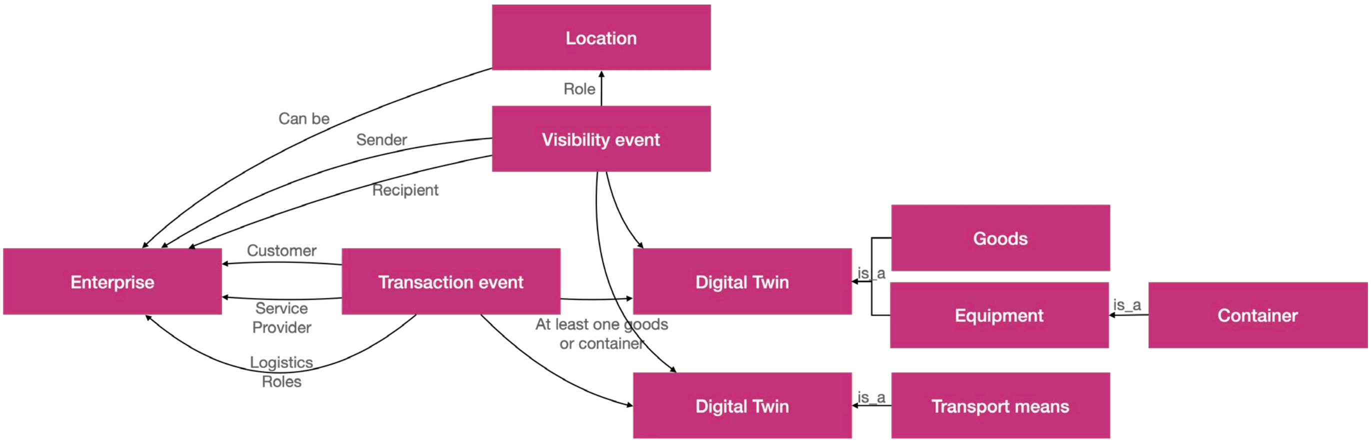


*Figure 6: Conceptual structure of states*

The figure shows two types of events, namely a state – and visibility event. The visibility events have been specified. The state event is like the visibility event but contains at least two visibility events for a 'transport' activity, namely the load and unload event. This is an extra constraint to the visibility events.

Furthermore, a visibility event represents the associations between Digital Twins (in place and time) related to a state event. This relation is given by the fact that Digital Twins of visibility events completely overlap or are a subset of those given by a state event and the sender/recipient associations of visibility events equals the customer/service provider associations of state event as specified by the choreography.

The 'agreed order' state contains data reflecting a customer order (this table must be updated for multimodal transport; recipient must also still be included):

- General event data (event (state data) reflecting header data. This refers to cargo (goods or equipment), organizations involved (consignor, carrier, consignee), and a transport modality and/or means.
- Associations are via UUIDs (Universal Unique Identifiers)
- Each concept (organization, location, etc.) has a user interpretable identifier like an equipment id.
- Equipment is generic, in the sense that it reflects a trailer, container, or any other type of equipment.
- Actual details of the movement of goods or equipment are given by two visibility events, one with

the Place of Acceptance (PLA) and the other with the Place of Delivery (PLD).

- If required, any intermediate location can be included, for instance that of border crossing for cargo going into or moving out of the EU.
- Any queries on individual concepts (like Digital Twin – goods) will only result in those data properties given for these concepts.

All concepts like event, organization, location, and Digital Twin have UUIDs, meaning that a ‘state’ only stores relevant data once and data can be shared by multiple orders and visibility events. If for instance multiple orders are transported by a single truck or trailer (LTL – less than truck load), the UUID of the truck or trailer is stored only once. In this case, each order and its visibility events are only shared by a service provider with a single customer of that order. The operator of a means of transport may provide the event to a node, which distributes it to the relevant CAs and customer.

The following table shows the various states of the multimodal visibility service. It shows that visibility events update the associations between digital twins and digital twins and locations. These events add the actual aspects for load/unload, position, and accident/incident as is specified by the state transitions hereafter.

| Business activity: transport | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Main structure of states | Agreed order | | Visibility<br>in execution | | | | Completed | | | |
| | Load | Unload | Load | Unload | position(s) | incident | Load | Unload | position(s) | incident |
| Event (state data) | | | | | | | | | | |
| UUID (event) | x | | x | | | | x | | | |
| UUID (customer) | x | | x | | | | x | | | |
| UUID (service provider) | x | | x | | | | x | | | |
| event type (value - assigned for the abstract message tree) | x | | x | | | | x | | | |
| milestone | | | | | | | | | | |
| expected date/time | | | | | | | | | | |
| estimated date/time | | | | | | | | | | |
| actual date/time | | | | | | | | | | |
| External reference (this is the user identification) | x | | x | | | | x | | | |
| Reference type (related to the abstract message tree) | Order number | | CMR number | | | | CMR number | | | |
| Organizations | | | | | | | | | | |
| Organization Role: consignor, consignee, carrier | x | | x | | | | x | | | |
| UUID Organization | x | | x | | | | x | | | |
| Cargo (at least goods or equipment as a rule; can be multiple) | x | | x | | | | x | | | |
| For goods - UUID (goods) | o (details) | | o (details) | | | | o (details) | | | |
| For goods - UUID (goods) | x (totals) | | x (totals) | | | | x (totals) | | | |
| for equipment - UUID (equipment) | o | | o | | | | o | | | |
| UUID (transport means) | x | | x | | | | x | | | |
| Person - legal person | | | | | | | | | | |
| UUID | x | | x | | | | x | | | |
| External ID | x | | x | | | | x | | | |
| External ID provider | x | | x | | | | x | | | |
| Name | x | | x | | | | x | | | |
| Infrastructure - Location | | | | | | | | | | |
| UUID | x | | x | | | | x | | | |
| Location code | x | | x | | | | x | | | |
| Location code type | x | | x | | | | x | | | |
| Address | x | | x | | | | x | | | |
| Postal code | x | | x | | | | x | | | |
| Street name | x | | x | | | | x | | | |
| City name | x | | x | | | | x | | | |
| Country code | x | | x | | | | x | | | |
| Digital Twin - goods | | | | | | | | | | |
| UUID | x | | x | | | | x | | | |
| External ID | x | | x | | | | x | | | |
| Type of external ID | x | | x | | | | x | | | |
| Type of packages | x | | x | | | | x | | | |
| Number of packages | x | | x | | | | x | | | |
| Gross mass (kg) | x | | x | | | | x | | | |
| Net mass (kg) | x | | x | | | | x | | | |
| Gross volume (m3) | x | | x | | | | x | | | |
| remark | | | | | | o | | | | o |
| Digital Twin - equipment | | | | | | | | | | |
| UUID | x | | x | | | | x | | | |
| Equipment ID | x | | x | | | | x | | | |
| Equipment Type | x | | x | | | | x | | | |
| Equipment Size | x | | x | | | | x | | | |
| Transport Equipment Packed Status | x | | x | | | | x | | | |
| Remark | | | | | | o | | | | o |
| Digital Twin - transport means | | | | | | | | | | |
| UUID | x | | x | | | | x | | | |
| Transport means ID | | | x | | | | x | | | |
| Transport means ID provider | | | x | | | | x | | | |
| Transport means Nationality | | | x | | | | x | | | |
| Transport mode | x | | x | | | | x | | | |
| Event (organization location) | | | | | | | | | | |
| UUID (event) | x | | x | | | | x | | | |
| event subtype | | | | | | | | | | |
| milestone | start | | start | | | | start | | | |
| expeted date/time | | | | | | | | | | |
| estimated date/time | | | | | | | | | | |
| actual date/time | x | | x | | | | x | | | |
| External reference | | | | | | | | | | |
| Reference type | | | | | | | | | | |
| UUID Organization | x | | x | | | | x | | | |
| UUID (location) | x | | x | | | | x | | | |
| Event (visibility data) | | | | | o | o | | | o | o |
| UUID (event) | x | x | x | x | x | x | x | x | x | x |
| UUID (sender organisation (equals UUID service provider) | x | x | x | x | x | x | x | x | x | x |
| UUID (recipient organisation (equals UUID customer)) | x | x | x | x | x | x | x | x | x | x |
| event type (value - assigned for the abstract message tree) | x | x | x | x | x | x | x | x | x | x |
| milestone | start | end | start | end | start | start | start | end | start | start |
| expected date/time | x | x | x | x | x | | x | x | x | |
| estimated date/time | x | x | x | x | x | | x | x | x | |
| actual date/time | | | x | | x | x | x | x | x | x |
| Locations (at least two) | | | | | | | | | | |
| Location role: place of acceptance, place of delivery | PLA | PLD | PLA | PLD | Position | Position | PLA | PLD | Position | Position |
| UUID (location) | x | x | x | x | x | x | x | x | x | x |
| Cargo (at least goods or equipment as a rule; can be multiple) | x | x | x | x | | x | x | x | | x |
| For goods - UUID (goods) | o (details) | o (details) | o (details) | o (details) | | o (details) | o (details) | o (details) | | o (details) |
| For goods - UUID (goods) | x (totals) | x (totals) | x (totals) | x (totals) | | x (totals) | x (totals) | x (totals) | | x (totals) |
| for equipment - UUID (equipment) | o | o | o | o | | o | o | o | | o |
| UUID (transport means) | x | x | x | x | x | | x | x | x | |

### 3.4 State transitions – event logic

All event primitives that are shared as part of the multimodal visibility service update visibility details of all cargo known by state data. This allows for instance to share an ETA event for all cargo, but also to share a loss of single piece of cargo (an instance of Digital Twin that is cargo).

To enable this functionality, event data is stored at the level of individual cargo (instance of all Digital Twins in an order), thus enabling the overall structure of an order with its references to logistics stakeholders. The proposed approach can be amended to handle multiple transport operations for a single order.

All event primitives must be validated before they are processed as part of a state transition. This is a separate function that is not given by event logic.

The way by which the state transitions are specified, and events can be shared, transport can be completed for the remaining cargo after part of it might be lost or damaged. This is by including these details at the level of a particular piece of cargo in the state 'in execution'.

The state transitions are specified as follows (see the interaction pattern of the service; a service provider is always the sender of an event, a customer the recipient)

| **State transition** | **Start** | |
|---|---|---|
| Input state | Agreed order | Initial upload of the state must be performed for this transition to properly execute.<br>A load event contains all cargo (all UUIDs referring to goods or equipment) that have been loaded. Missing cargo must be detected. |
| Event primitive | Load event | |
| Pre-condition | Event (sender, recipient) exists agreed_order (service provider, customer).<br>(For all cargo in agreed_order) exists load_event (UUID)<br>Load_event (PLA) equals (agreed_order (cargo (event (PLA)))<br>Load_event (actual_date/time) in period (agreed_order (cargo (event (milstone = start; estimated))))) | First check always: is a visibility event related to a state event for enterprise roles, second check on Digital Twins.<br>Additional checks that the cargo is loaded at the place indicated by the order and the time is within the estimated period. |
| Error | Error with a code identifying one of the parts of the pre-condition that is not met:<br>- Order unknown<br>- Difference in place of acceptance<br>- Too late loaded (too early is not relevant, since cargo should not be available before a planned loading time) | First error must be shared with the sender of the event to prevent any unrequired data sharing. The event is not shared.<br>The other two errors are indicated to the sender of the event. The event is shared with the recipient, who receives the same error at reception. The sender of the event may recalculate the planned date and provide it as estimate time for completion by a separate event (potentially at a later stage). |
| Firing rule | If too late or too early then recalculate | - |

| State transition | Start | |
|---|---|---|
| | planned date | |
| Post-condition (output state) | For all cargo (event (visibility details): store load_event<br><br>Update event (state data) with load_event (transport means)<br><br>Optional: update event (state data) with load_event (equipment_trailer) | The transport means given in the load event is updated and included in the order.<br><br>If the cargo is loaded on a trailer (railway wagon can be included lateron), the trailer is added.<br><br>This transition can be expanded to cover partial loading of all cargo in a business transaction (e.g. a shipment or consignment); partial unloading is already supported. |

| State transition | ETA update | |
|---|---|---|
| Input state | In execution | There can be two different transitions, where an initial update of an ETA for unload is updated at a later stage.<br><br>An ETA event is applicable for all cargo given by the state. It can be given at the level of a transport means. |
| Event primitive | ETA event | |
| Pre-condition | Event (sender, recipient) exists agreed_order (service provider, customer).<br><br>(ETA event (UUID)) equals (event (state data) (UUID) and state 'in execution')<br><br>ETA event (estimated time; milestone = end) within period (event (state data) – for all cargo (UUID) – event (visibility details; estimated; milestone = end)) | A validation that an agreed order is in execution and the ETA is still within the time window. |
| Error | 'order in the execution state unknown'<br><br>'too late or early unload' | The first error is submitted to the sender of the event to prevent any unrequired data sharing.<br><br>The second error is shared with the sender of the event after which the ETA event is shared with the recipient. At reception, the error is shared with the recipient. |
| Firing rule | - | - |
| Post-condition (output state) | For each cargo (event (visibility details)) with milestone = end: update (estimated time) | In execution |

| State transition | Position update | |
|---|---|---|
| Input state | In execution | It is assumed that a position is only given once with its actual state, where the result is 'milestone = start' indicating the position is passed.<br>This could be updated by given an estimated time at which a position will be passed.<br>A position is applicable to all cargo loaded and given at the transport level. |
| Event primitive | Position event | |
| Pre-condition | Event (sender, recipient) exists agreed_order (service provider, customer).<br>(position event (UUID)) equals (event (state data) (UUID) and state 'in execution')<br>(position event (location) not in (all cargo for event (state data) with event (visibility details) (position)) | There must be an order in a state of execution and the position must not yet be given |
| Error | 'order in the execution state unknown'<br>'position already shared' | This error is submitted to the sender of the event to prevent any unrequired data sharing.<br>The sender of the event receives an error when the position is already shared and the event is not shared with a recipient. |
| Firing rule | - | - |
| Post-condition (output state) | For each cargo in (event (state data)) include event (visibility details) (milestone = start: location = position event (location) | In execution |

| State transition | Incident/accident | |
|---|---|---|
| Input state | In execution | Three types of incidents or accidents can be reported:<br>- Loss of cargo<br>- Damage of cargo<br>- Delay caused by an accident<br>The type of incident or accident is given by a code of the event. |
| Event primitive | Incident or accident event (in brief 'incident event' is given hereafter) | |
| Pre-condition | Event (sender, recipient) exists agreed_order (service provider, customer).<br>(incident event (UUID) exists in (event (state | First the order should exist and secondly for loss or damage the cargo must be present. |

| State transition | Incident/accident | |
|---|---|---|
| | data) (UUID; state = in execution)) Incident event (loss or damage) and (incident event (cargo UUID)) exists in event (state data) cargo (UUID) | |
| Error | 'order in execution unknown'<br>'cargo given for loss or damage unknown or not loaded' | These errors are shared with the sending actor to prevent any update with the recipient.<br>The second error indicates a state error. First action is to synchronize states between two participating stakeholders. |
| Firing rule | If Incident event (accident) then calculate (ETA unload; delay) and indicate error in case ETA unload exceeds the period given by the state | This is the single transition with a firing rule where the delay's impact is calculated. This calculation may be simple by for instance adding the delay to the estimated time of unloading (milestone -= end) or considering any other details that may occur after the accident. |
| Post-condition (output state) | Case<br>- Accident: for each cargo in (event (state data)) include accident event as event (visibility details) (milestone = start: location = accident event (location); accident event (actual time)<br>Update the estimated time for milestone = end of all cargo specified by the event (state data)<br>- Damage or loss: for all incident event (cargo UUID) include the incident event to the applicable cargo via the UUID. | In execution |

| State transition | Partial unload | |
|---|---|---|
| Input state | In execution | |
| Event primitive | Unload event (partial) | |
| Pre-condition | Event (sender, recipient) exists agreed_order (service provider, customer).<br>((unload event (UUID equipment)) exist in (event (state data) and (all UUID equipment; state = in execution) or (((unload event (UUID goods) exists in (event (state data) (UUID goods; state = in execution)) and ((unload event (UUID goods – number of packages)) equal or less than (event (state data) (each UUID goods – number of | Partial unloading is indicated via the event.<br>The unload must also fit in the time window of the estimated/planned data, since it is the basis for synchronization with adjacent legs. The location must also equal the expected. |

| State transition | Partial unload | |
|---|---|---|
| | packages)) and (for each cargo in (event (state data) (location for visibility event with milestone = end, expected date given) equals (unload event (location) | |
| Error | 'not all cargo unloaded'<br>'not all packages unloaded'<br>'unloading at a different location than expected'<br>'unloading too early or too late as planned' | Following type of errors:<br>- Not all equipment or goods are unloaded<br>- Not all packages of goods are unloaded (missing packages that are not reported)<br>- Cargo is unloaded at a different location than expected by a customer<br>- Unloading too early or too late. In case of a partial unload, only the completion will specify the final time and its position with respect to the time window (too late or too early).<br>These errors may not necessary give rise to an action by a sender, like the unloading location might change and has to be reported.<br>In case of unloading of partial shipments this is reported via the transition 'partial unload' |
| Firing rule | - | - |
| Post-condition (output state) | For each cargo in event (visibility details) include event (visibility details) (milestone = end: location = unload event (location) in event (visibility details) | Only those cargo will be in state 'completed' that are reported. All others remain in the state 'in execution' |

| State transition | Complete unload | |
|---|---|---|
| Input state | In execution | |
| Event primitive | Unload event (final) | |
| Pre-condition | Event (sender, recipient) exists agreed_order (service provider, customer).<br>((event (state data) (all UUID equipment; state = in execution) exist in (unload event (UUID equipment)) or (((event (state data) (UUID goods; state = in execution) exists in (unload event (UUID goods)) and ((event (state data) (each UUID goods – number of packages) equals (unload event (UUID goods – number of packages)) and (for each cargo in (event (state data) (location for visibility event with milestone = end, | All cargo that was loaded is reported by unloaded by a single unload event, unless a loss has been shared during transport with an incident event.<br>The unload must also fit in the time window of the estimated/planned data, since it is the basis for synchronization with adjacent legs |

| State transition | Complete unload | |
|---|---|---|
| | expected date given) equals (unload event (location) | |
| Error | ‘not all cargo unloaded’<br>‘not all packages unloaded’<br>‘unloading at a different location than expected’ | Three types of errors:<br>- Not all equipment or goods are unloaded<br>- Not all packages of goods are unloaded (missing packages that are not reported)<br>- Cargo is unloaded at a different location than expected by a customer<br>These errors may not necessary give rise to an action by a sender, like the unloading location might change and has to be reported.<br>In case of unloading of partial shipments this is reported via the transition ‘partial unload’ |
| Firing rule | - | - |
| Post-condition (output state) | For each cargo in (event (state data)) include event (visibility details) (milestone = end: location = unload event (location) | Completed, where any damage to cargo is stored as mentioned by the event. |

Any pre-condition can be implemented by SPARQL queries where event data is used to query state data. A positive result of the query makes the pre-condition ‘true’; a negative must generate an ‘error’, where depending on the error action is taken (see the specifications).

The pre-condition is an update to the state data with the event data. Existing events (visibility) at the level of cargo may have to be updated or new events (visibility) must be inserted. For instance, it will be good to keep track of any ETA updates by including them to each digital twin representing cargo.

In case any ‘ETA –‘, ‘position –‘, or ‘accident/incident event’ is shared in the context of an order after the unload event has been processed and the state is ‘completed’, these events are discarded. In case this is at the sender (i.e. the service provider), they are not shared with the recipient (i.e. the customer). In case this is at the recipient (i.e. the customer), they are out of sequence and can be discarded. Any data contained by those events could have caused a delay or damage or loss of cargo, which is already detected when processing the unload event.

Any unload event must be preceded by a load event. If a sender still tries to share an unload event without having shared a load event, a warning must be given, and the unload event will not be shared. If a recipient receives an unload event without having received a load event, this is an error (the unload event can only be processed when the state is ‘in execution’, see the transitions). In case of such an error, synchronization of state between sender and recipient might be necessary.

### 3.5 Query formulation

The UUIDs of events and their referenced transport means, equipment, and goods are the basis for retrieving more information by enterprises. Each enterprise can formulate its own queries or re-use standardized queries. Examples of those queries are:

- **Retrieve general information** – based on the UUID of a load event, the order or document data set is retrieved.
- **Retrieve detailed information** – based on the UUID of a visibility event that is retrieved via for instance the UUID of a transport means or equipment, an enterprise may want to receive details of the cargo of, including its agreed order (consignment data). For an enterprise this might be for instance the weight of a container.
- **Retrieve specific information of a Digital Twin** – the UUID event of load/discharge must contain the UUIDs of the relevant Digital Twins, since the user defined identifier (e.g. license plate of a truck or trailer, container number) are used to by a data user like a CA for remote monitoring. The specific information may contain details of the goods or content of a trailer/container, which is specified by the query formulated by that CA, where this goods details may only be available to a consignor and not a carrier. This requires access policies and authorization to be specified, based on the fact that links are shared by a data holder with a data user as specified by the FEDeRATED architecture.

The following table lists an example of these queries. This example is not complete and does not reflect the actual situation like the eFTI, AWB, or any other data set. It serves as a basis for demonstration.

| Functional data requirements and mapping to the ontology | | | business (b2b) | | | | |
|---|---|---|---|---|---|---|---|
| **Functional** | **data properties** | **Semantic model ('concept.property')** | **road** | **air** | **inland waterways** | **rail** | **sea** |
| | User identification | | eCMR number | AWB number | | CIM number | B/L number |
| Consignor | | Consignor | x | x | x (customer) | x (customer) | x |
| | Consignor ID | organization.ID | o | o | | | o |
| | Consignor Name | Organization.name | x | x | x | x | x |
| | Consignor Address | Location.postal address | x | x | x | x | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | x | x | x | x | x |
| | Country code | location.country code | x | x | x | x | x |
| Consignee | | consignee | x | x | | | x |
| | Consignee ID | organization.ID | o | o | | | o |
| | Consignee Name | Organization.name | x | x | | | x |
| | Consignee Address | Location.postal address | x | x | | | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | x | x | | | x |
| | Country code | location.country code | x | x | | | x |
| Carrier | | Carrier | x | x | x (barge operator) | x (RU) | x |
| | Carrier ID | organization.ID | o | o | x | x | o |
| | Carrier Name | Organization.name | x | x | x | x | x |
| | Carrier Address | Location.postal address | x | x | x | x | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | x | x | | x | x |
| | Country code | location.country code | x | x | | x | x |
| Forwarder | | Forwarder | | | | | x |
| | Forwarder ID | organization.ID | | | | | o |
| | Forwarder Name | Organization.name | | | | | x |
| | Forwarder Address | Location.postal address | | | | | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | | | | | x |
| | Country code | location.country code | | | | | x |
| Locations (pla/pld, pol/pod, etc.) | | | | | | | |
| Goods identification | | Digital Twin-goods-identification | xor (goods or equipment) | xor (pieces) | | | |
| | | | | | | | |
| | Type of goods | Digtal Twin - goods - Type of goods | x | x | | | |
| | Gross mass (kg) | Digital Twin - goods - gross maas | x | x | | | |
| | Net mass (kg) | Digital Twin - goods - net mass | | | | | |
| | Gross volume (m3) | Digital Twin - goods - gross volume | x | x | | | |
| | Number of packages Total Packages | Digital Twin - goods - number of units | x | x | | | |
| | | Digital Twin - goods - nature of the cargo | | | | | |
| | Goods description (textual) | Digital Twin - goods - goods description | x | x | | | |
| Transport Equipment | | Digital Twin - Equipment - Container | xor (see goods) | | x | xor (container or | xor (container or |
| | Transport Equipment ID | Digital Twin - Equipment - Container - ID | x | | x | x | x |
| | Transport Equipment Type | Digital Twin - Equipment - Container - type | x | | x | x | x |
| | Transport Equipment Size | Digital Twin - Equipment - Container - size | x | | x | x | x |
| | Transport Equipment Packed Status | | | | | | |
| | Seal Quantity | | | | | | |
| | Sealed Indicator | | x | | x | | x |
| | Seals | | o | | o | | o |
| | Seal number | | x | | x | | x |
| | Incident code | | | | | | |
| | Sealing Party Role Code | | x | | x | | x |
| Transport means | | Digital Twin - Transport means | x | x (flight) | x (barge) | x (train) | x (vessel or ferry) |
| | Transport means / mode type | | x (road) | x (air) | x (inland waterways) | x (train number) | x (sea) |
| | Conveyance reference number | | o | x (flight number) | | | o |
| | Transport means ID | | x (license plate truck) | o | x (barge ID) | o | x (vessel ID) |
| | Transport means Nationality | | x | o | x | x | x |
| Transport Equipment | | Digital Twin - Equipment - trailer | o | | | xor (euipment or trailer) | xor (container or trailer) |
| | Transport Equipment ID | Digital Twin - Equipment - trailer-ID | x | | | x | x |
| | Transport Equipment Type | Digital Twin - Equipment - trailer - type | x | | | x | x |
| | Transport Equipment Size | Digital Twin - equipment - trailer - size | x | | | x | x |
| Transport Equipment | | Digital Twin - Equipment - ULD | | xor | | | |
| | Transport Equipment ID | Digital Twin - Equipment - ULD - ID | | x | | | |
| | Transport Equipment Type | Digital Twin - Equipment - ULD - type | | x | | | |
| | Transport Equipment Size | Digital Twin - Equipment - ULD - size | | x | | | |
| Transport Equipment | | Digital Twin - Equipment - wagon | | | | x | |
| | Transport Equipment ID | Digital Twin - Equipment - wagon - ID | | | | x | |
| | Transport Equipment Type | Digital Twin - Equipment - wagon - type | | | | x | |
| | Transport Equipment Size | Digital Twin - Equipment - wagon - size | | | | x | |

### 3.6 *Additional conditions - release*

One issue that is not yet described is that next steps in the process may only be performed when other conditions are true. These are for instance in transhipment from a (deep) sea vessel to another transport means for incoming cargo. For instance, the following conditions must be met:

- **Commercial release** – transport and handling charges of the previous transport leg and trans-shipment have been paid. A bank can produce such a release; other relevant stakeholders require that such a release token is published by an authorised bank.
- **Customs release** – especially for incoming cargo, customs has the ability of inspection and must issue a release relevant for a terminal operator and carrier. Like a commercial release, relevant stakeholders need to know that such a release token is provided by the appropriate customs authority.

Further research is required as to the support of these tokens by Verifiable Credentials. Such tokens would identify the holder like customs.

These conditions will be part of the state transitions to validate compliance. Of course, they need to be validated at physical hand-over of cargo.

# 4 Physical operation

Physical operations result in reports according to instructions. These reports, i.e. visibility events, can be generated by sensors, e.g. a sensor on a crane reporting load or unload of a container from a vessel or truck or an ETA of a truck generated by its route planning software, or by personnel operating devices, e.g. a truck driver registering handling part of its instructions via an On Board Unit (OBU).

This section provides a specification for these types of events and their logic. It takes two views on cargo movements, namely that of a carrier (transport means) and a hub (location). Both will have states that need to match for loading and unloading cargo. The carrier view represents an itinerary with multiple places of call, positions, and border crossing locations. The hub view is about multiple transport means calling upon that hub for load and unload of cargo.

Both a carrier and hub can produce visibility events. This part defines how these visibility events are generated from a physical operation by an operator, driver, or sensor to IT systems of their organization, potentially utilizing a node (see the introduction and the architecture). Both the carrier and the hub operator may share these events with their customers, whereby these customers can synchronize operations between carriers and hubs (see next section).

Instruction sets as a basis for events with their sequencing are given for carrier – and hub operators for (un)load of cargo. These can be further decomposed for a port (arrival and departure, relating to additional services like bunkering and piloting) and other activities like warehousing and stuffing/stripping. The approach is applicable to transshipment in any form, i.e. between any two modalities, with further refinement for warehousing and stuffing/stripping. (Un)load operations may also be replaced by others like ‘cleaning’ for tankers. These activities can be followed by other activities like container cleansing or - storing (the hub is then called ‘empty depot’).

## *4.1 Instruction sets for carrier - and hub operators*

Physical actions are represented by (physical) state transitions that can be modelled for ‘transport’. The start or end of an itinerary is specific to an organization and/or modality. For instance, an itinerary by air, called ‘flight’ with a ‘flight number’, has a start at an airport where cargo is loaded (place of call), can have an intermediate airport where cargo is (un)loaded, and a destination where all cargo is unloaded. The airport where a flight starts can be the home base of an airplane. A trip of a truck will most often start at its home base and will last till it returns to that home base or is empty and parks somewhere. Similar rules can be formulated for a voyage of a vessel.

From a data perspective, an itinerary of a transport consists of a set of instructions for loading/unloading cargo at locations. Such an instruction set has the following structure:

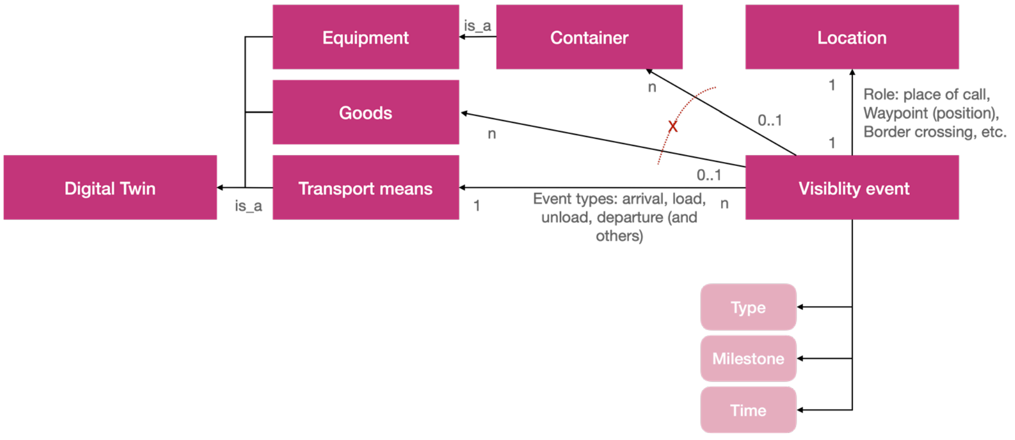


*Figure 7: Instruction set for an itinerary of a transport means*

The figure shows that a single means of transport has multiple instructions, represented as 'visibility events', of different types (arrival, departure, etc.), where each event refers to a single location. In case of load or unload events, these refer to a location where either containers or goods are loaded or unloaded. Thus, a visibility event must have a single location and depending on its type can have multiple goods or containers (this is indicated by the association with '0..1' and 'n').

Additional remarks to this figure are:

- The role of a location may change or differ per use case or modality. Other locations may have to be included like the location where a pilot leaves or comes aboard a vessel, berth location of a vessel, etc. These examples of locations are refinements in a port area.
- The event types may refer to the roles of locations, like the event that a pilot comes aboard or a vessel is at its berth location.

An itinerary thus consists of a sequence of instructions represented by events with places of call. Individual calls or (part of) an itinerary can be published by its owner, either as open data via the Service Registry or accessible to a community. Notice that for mobility of individual persons, most itineraries of transport means are published as open data.

The instruction set for operators in a hub relate to handling a means of transport (arrival, departure, etc.) and cargo operations (load, unload). These instructions are represented by multiple visibility events related to a single location, where each event refers to one transport means.

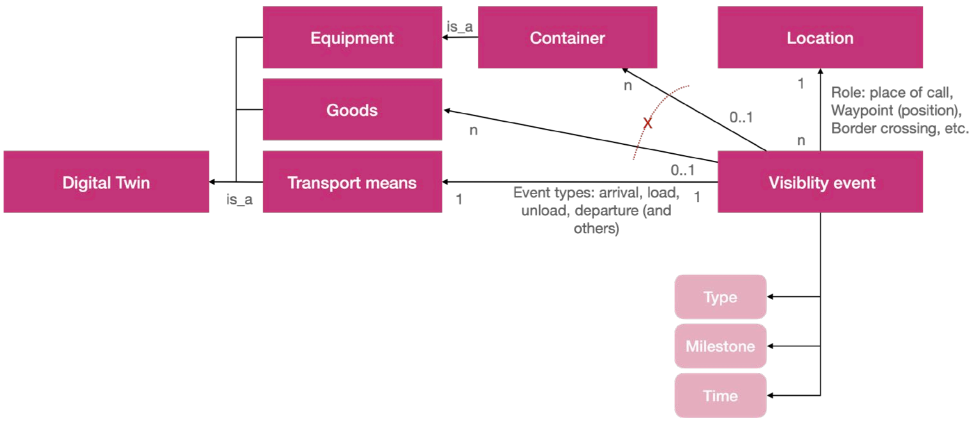


*Figure 8: Instruction set for hub operators.*

Note that multiplications of the associations are changed in the previous figure.

The instruction set for hub operators is internal to that hub, but events generated by an operator in a hub are shared with a customer. For instance, the arrival of a means of transport is reported to a customer via an ATA event.

Where the instruction set composing an itinerary is mostly handled by a single operator, e.g. a truck driver or barge operator, the instruction set of a hub operator can be split for different operators. For instance, an instruction set for a gate operator can be distinguished from that of another operator responsible for load/unload. Furthermore, a distinction between and gate for trucks and quays for vessels can lead to different instruction sets.

The previous event structures combine all activities that can take place a location. Individual events can be extracted from the previous ones. Firstly, the arrival and departure events. As the figure indicates, these are shared between a sender and recipient. For carrier and hub operators, the sender is always the employee (or a sensor) generating and thus the sender of the event.

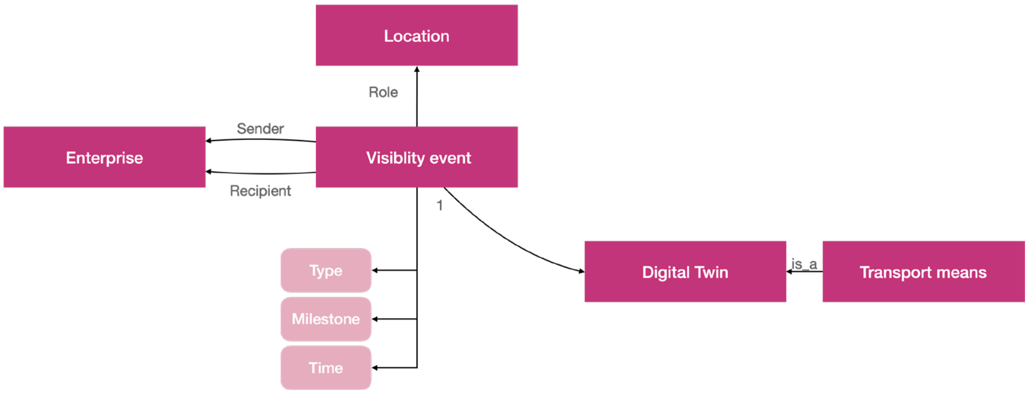


*Figure 9: arrival and departure event structure (expected, estimated arrival (ETA), actual arrival (ATA), estimated departure (ETD), actual departure (ATD), and required arrival (indicated by a hub))*

A hub will never be able to provide an ETA event. A hub operator may provide a required arrival event to optimize its internal operations and the period that a transport means is at a hub. A carrier may not be able to generate an ETD event, since this depends on the activities performed by the hub operator. A carrier will never send or receive a required arrival event. In case a hub operator generates a required arrival event, this is routed by its customer to the carrier. If the carrier is the customer of the hub, he will receive it and provide an 'expected arrival event' to its operator.

Secondly, the load and unload event.

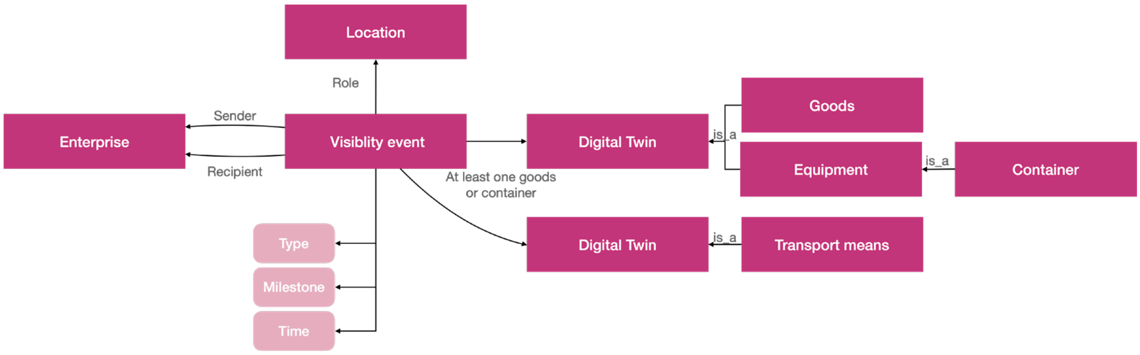


*Figure 10: load/unload events (expected, planned/estimated, actual)*

## 4.2 The process – event sequencing

Carrier – and hub operators (or sensors) generate events in certain sequence. The events that they will generate are those that are given by their instruction set. This instruction set contains the expected dates/times (periods) for performing a task; operators will provide the actual ones.

The following events can be generated by a carrier – and hub operator to their back office as a basis for providing relevant information to customers:

| Event type | Carrier operator | Hub operator | Remarks |
|---|---|---|---|
| Estimated Time of Arrival (ETA) | X | | |
| Actual Time of Arrival (ATA) | X | X | There can be multiple ATA events, like 'gate-in' and 'at dock', or 'port' and 'alongside quay'. |
| Required Time of Arrival | | X | This event is generated by a back office to its customer |
| Planned load time | (x) | X | Whether actual (un)load events are generated by a carrier - or hub operator depends on the commercial relation of a carrier with the hub. If a carrier is a customer, a hub will provide these events. Otherwise, both actors can provide them. In most cases, a hub operator can provide the planning details. |
| Actual load time | X | X | |
| Planned unload time | (x) | X | |
| Actual unload time | X | X | |
| Missing load (optional) | X | X | It could be worthwhile to distinguish these as separate events to indicate that cargo could not be (un)loaded. Implicitly, missing cargo can be detected by a back office after departure of a transport means. ATD signals that nothing can be (un)loaded from a transport means at a location |
| Missing unload (optional) | X | X | |
| Estimated Departure Time (ETD) | (x) | X | This depends on the planning of a hub and will thus most probably be generated by its back office to its customer. It could be generated by a carrier operator, informed by a hub operator. |
| Actual Departure Time (ATD) | X | X | |
| Border crossing event | X | | |
| Position event | X | | |

The sequence by which the events are submitted by a carrier operator to its back office is as follows.

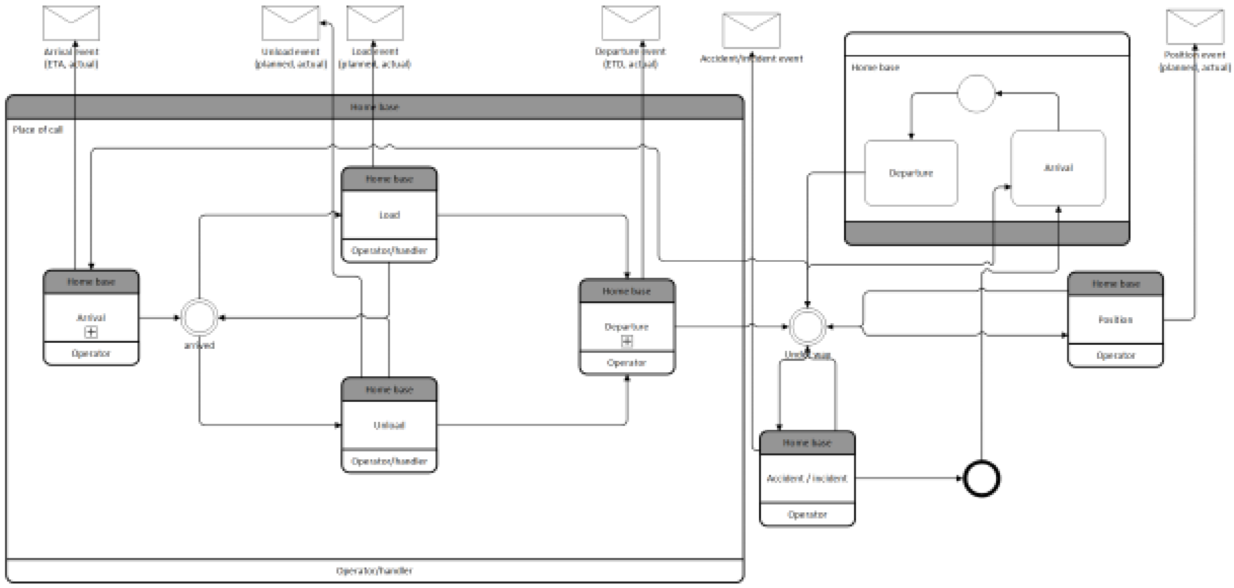

*Figure 11: Generation of events for finishing instructions by a carrier operator*

The transitions shown in this figure represent physical activities. The states represent a transport means and its cargo. For instance, 'arrived' indicates that a transport means is at a location with its cargo. It is still there after (un)load activities have been performed, but the cargo of the transport means and present at the hub has changed.

The figure shows that an itinerary can be endless (like voyages of a vessel), between two locations with an intermediate stop (like flights), or a transport means can return to its home base after finishing its instruction (e.g. relevant to road carriers).

The previous figure also shows that arrival and departure can be decomposed. This is especially the case for vessels calling upon a port or flights calling upon an airport. The decomposition may differ per (air)port, thus giving specific procedures for each (air)port.

In between a port call, i.e. between arrival and departure, additional physical activities can take place like piloting, bunkering, and change of crew. Piloting is of course between a waypoint where pilots come onboard till the vessel is alongside quay (or ready for (un)loading); bunkering can interrupt (un)load activities. This decomposition can be given per port.

In a similar way, event sequencing for a hub can be given. The physical activities are not detailed since these don't take place in IT systems.

At reception by a back office, events are the basis for informing a customer (see 'event federation') or a supervising authority (see 'compliance to regulations'). A back office will have processes to detect exceptions (like 'too late', 'too early', and 'missing cargo') as indicated in the previous section (these are state transitions like 'start' and 'complete'). These state transitions can be decomposed for representing the complete instruction to a carrier – or hub operator (these decompositions are not provided). They also consider 'state data' of all individual instructions.

# 5 Event federation

Event federation is an internal Value-Added Service (VAS) of an organization operating in a supply and logistics network. The VAS consists of a single, internal transition that is triggered by reception of events. The functionality of this VAS is derived by examples of logistics chains, thus identifying the potential cases supported by the VAS. It is all about synchronizing legs in a chain and informing customers of the progress of chains.

This section is about utilizing received events for generating new events. It is an internal transition, called 'event federation'. To be able to provide its functionality, an example of a chain with its transaction tree is given (next figure). The transaction tree depicst roles and responsibilities for transport of a container via a port by sea to another country.

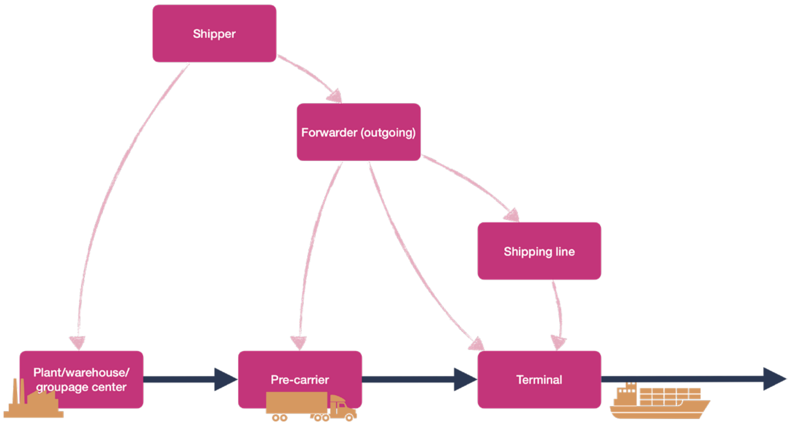


*Figure 12: example of a chain*

As specified in the architecture, visibility in the context of this transaction tree is configured by 'order' using for instance a publish/subscribe mechanism. The combination of orders resulting in visibility events is visualized by the next sequence diagram for the previous chain. It shows in brackets the visibility events identified in this document.

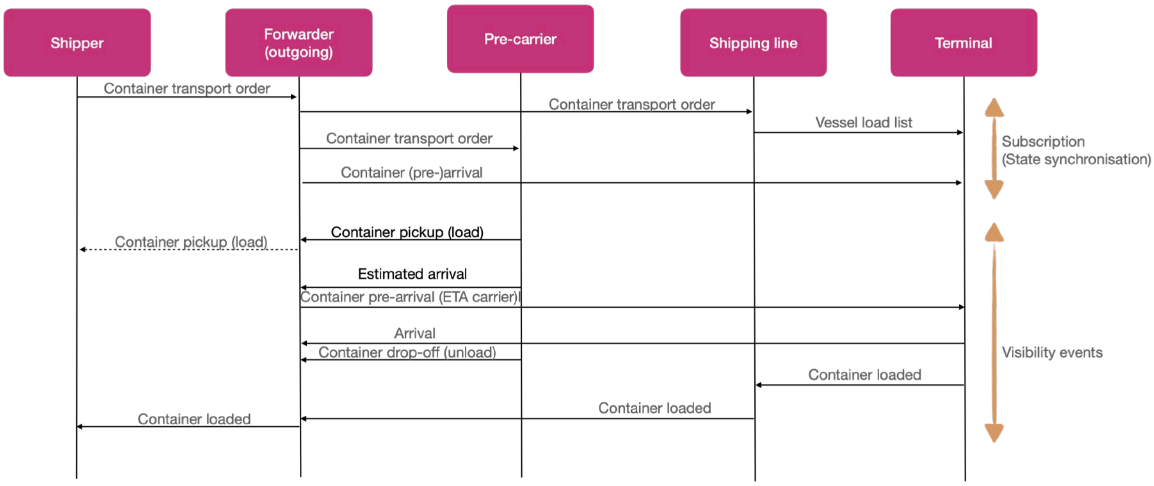


*Figure 13: event sequencing in the example*

The figure shows that a pickup event (i.e. a load event) of a carrier can be passed on to a shipper by a forwarder. Another example is given by an estimated arrival (ETA) of a carrier that is past to the terminal and reconfiguring the state information. The carrier ETA is now the expected time of arrival given by the forwarder.

These are two examples where events tat can trigger informing a customer and next leg (semi-)automatically.

This example shows two situations, namely updates of expectations received from a customer and updates to next legs and a customer based on a progress event received from a service provider. Additionally, there is the case whereby a next leg is a type of activity with a fixed schedule like flight or train departure. This requires the previous leg to be finished on time. These three situations are visualized as follows.

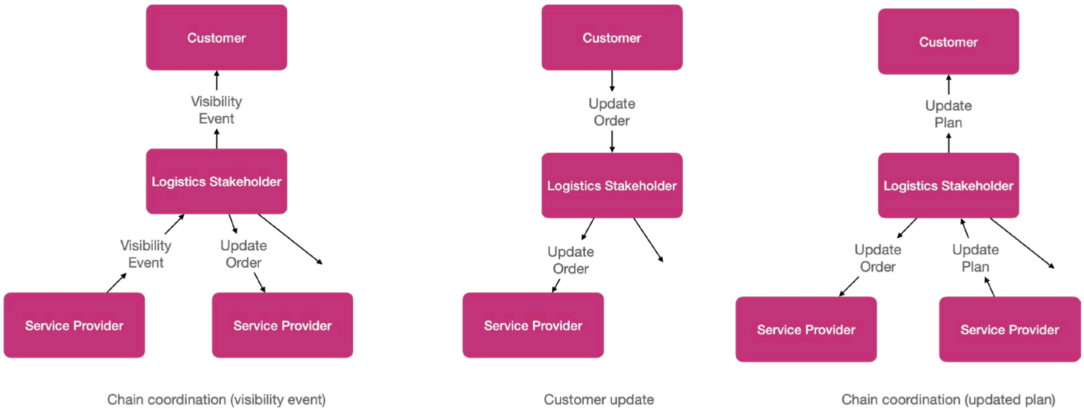


*Figure 14: three potential situations for automatically informing chain participants*

The previous figure shows updates of expectations by a customer that can be based on progress made by previous legs in a chain. These can be updates on the expected time at which a next leg could start, based on an ETA of a previous one.

The simplest case of the chain coordination is where a carrier informs a customer on delivery of goods, whereas this customer informs its customer. This reflects for instance an eCommerce delivery activity, or the example given before of the container pickup and pre-arrival.

Only the first situation will be specified since it deals with visibility events where the other two situations are relevant to ordering.

Chain coordination is triggered after reception of a visibility event of a service provider, where the ETA for and/or the completion of the activity is given (or the ETA is calculated). The relevant state transitions of event logic are extended by producing an internal event as part of their firing rule. The following state transitions are relevant:

- ETA update – this can affect the execution of the next leg (or provide an indication of completion to a customer).
- Incident/accident – similar as with ETA update, with the addition that a next leg may have to be cancelled and the activity cannot be completed.
- Completed – this will provide an indication to a customer, whereas the next leg will already have received the relevant cargo.

The firing rule of these transitions is updated with ‘generate (internal_event)’, resulting in the following transition (the transition may require further refinement in alignment with the previous section):

| State transition | Event federation | |
|---|---|---|
| Input state | - | - |
| Event primitive | Internal event (UUID_receivedVisibility_event) | The UUID of the original visibility event is the trigger. The assumption is that it has been processed successfully by the recipient. |
| Pre-condition | ReceivedVisibility_event (Digital Twin (UUIDs)) exist in State_event (receivedVisibility_event (recipient = (customer and serviceProvider not equal receivedVisibility_event (sender))) or (recipient = (serviceProvider and customer not equal receivedVisibility_event (sender)))<br><br>(state_event (serviceProvider; visibility_event (unload, location) = receivedVisibility_event (location)) or (state_event (customer; visibility_event(load, location) = receivedVisibility_event (location)) | First check: the Digital Twins of the received visibility event participate in another state event.<br><br>Second check: validate if the received visibility event represents the final leg of a chain or there is an adjacent leg. |
| Error | - | If the pre-condition is false, no action will be performed. |
| Firing rule | If (ETA event) and finalLeg then generate (ETA event) to customer<br><br>If (ATA event) and finalLeg then generate (unload event) to customr<br><br>If (ETA - or ATA event) and (ETA or ATA is not in adjacentLeg (plannedPeriod)) then generate (orderEvent(expected time = (ETA or ATA)) to adjacentLeg (serviceProvider)<br><br>If receivedVisibility_event = incidentEvent (damage or loss) and finalLeg then duplicate incidentEvent (recipient = customer)<br><br>If receivedVisibility_event = incidentEvent (damage or loss) for all adjacentLegs cancel(adjacentLeg) | The output generated by the firing rule depends on the type of visibility event received. If it is an ETA or ATA of the final leg, the customer must be informed.<br><br>If it is an ETA or ATA relevant to an adjacent leg (i.e. the ETA or ATA is not within the agreed period), that service provider receives an update.<br><br>The previous rule might be upgraded in case the delay is too long and the adjacent legs must be cancelled.<br><br>In case of an incident event with (total) damage or loss, at least the customer must be informed. If there are adjacent legs, these must be cancelled. |
| Post-condition (output state) | All generated events are stored for a customer.<br><br>The agreed_order state with a Service provider is updated in case an orderEvent is shared.<br><br>The agreed_order state is towards 'cancelled' in case a cancelation is shared. | |

The previous process may require an update if the location given by an unload or ETA event that has been received is not equal to the unload location given by the customer and there is no adjacent leg. In that case, the adjacent leg may have to be cancelled or start at the new location. An example is where a container that has been discharged in Antwerp was expected to be transported from Rotterdam to its destination.

In the previous case, it could also be that all adjacent legs are cancelled and a new adjacent leg to the destination is to be organized.

# 6 Compliance to regulations

Compliance to regulations implies pushing data (like for import declarations) or providing links to data to competent authorities (CAs) whereby these CAs can access data they require in their supervision role for a (one or more) regulation(s). The eFTI (electronic Freight Transport Information) Regulation is an example of a regulation by which a CA must be able to access relevant data when required in the case that an economic operator is completely digitized for transport activities.

Links to additional data provide visibility to CAs. As such, this mechanism is specified. It consists of two aspects, namely the distribution of events with links and the data that must be accessible to a CA upon evaluating a link. This may require an enterprise to share additional events (with links to data) to a CA and/or to support data that is only relevant to a CA (all in the context of a regulation).

This section presents an event distribution mechanism and makes a proposal for accessing additional data. The latter can easily be replaced by, for instance, the eFTI data set. An event (with a link) can also be physically handed over to a CA in case of physical inspection or shared via an infrastructure for remote surveillance. In case of remote surveillance, a CA requires access to data of all cargo onboard of a means of transport. Especially for emergency situations like accidents this data is required.

## *6.1 Event distribution to CAs*

A CA requires visibility events as to the movement of cargo and transport means coming in, going out, or passing their competency domain. This is visualized by the following choreography:

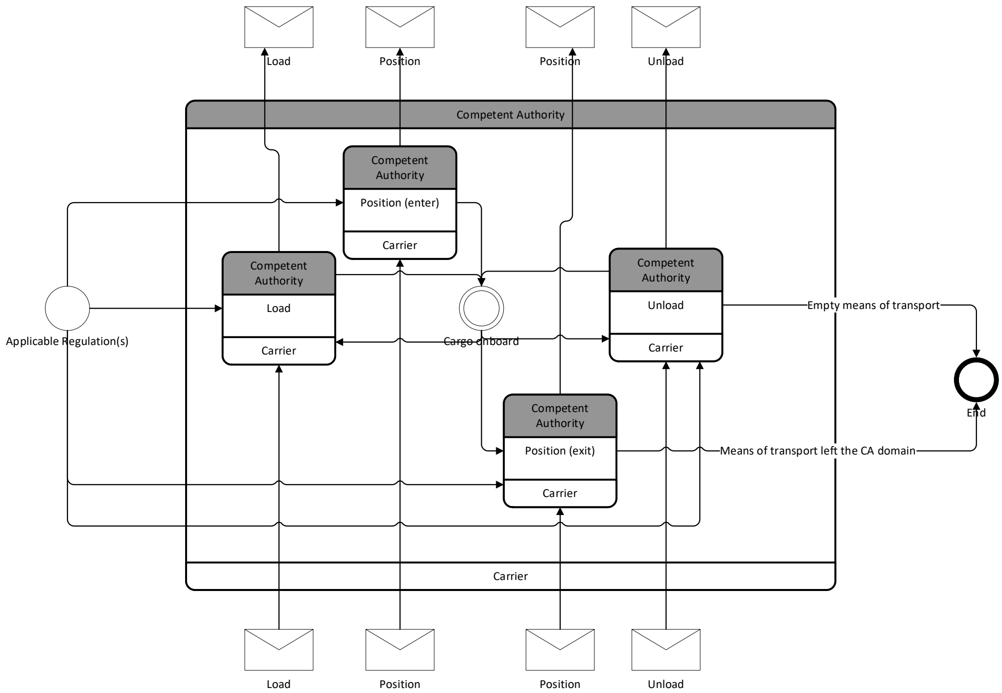


*Figure 15: event distribution of carriers to CAs*

The previous figure shows that events are shared by a carrier to a CA, based on an applicable regulation. This is to be provided by a CA according to the legislation process (e.g. that can include EC, MS, and national/local authorities). Part of the applicable regulations are:

1. **Competent Authorities** – those that supervise the applicable regulations(s).
2. **CA territory** – the supervision domain of a Competent Authority, for instance a country, region, or

municipality. A service is assumed to be available that can match a location with any territory.

The rules for event distribution are based on state data of the multimodal visibility service as specified in section 3.3. It is about the PLA (Place of Acceptance), PLD (Place of Delivery), and position (border crossing) shown in the visibility state data. The event structures are equal to those specified in section 3.2.

There are different ways to implement the state transitions of the previous figure. They can be included in the firing rule of the visibility service or they can be implemented separately triggered by the events that are part of the visibility service. Considering separation of concerns, the latter solution is better, since it can easily be used to develop additional functionality or change existing functionality.

The state transitions visualized in the previous figure are triggered by visibility events that are already shared in a customer – service provider relationship (section 2). The state transitions are as follows:

| **State transition** | **Load** | |
|---|---|---|
| Input state | Applicable regulations | The applicable regulations with CAs and their competence domain |
| Event primitive | Load event | |
| Pre-condition | match (extract(event(visibility)_PLA (Place of Acceptance), applicable_regulations (CA territory) | This indicates that particular cargo is onboard a transport means in the domain of a CA. The match function must result in true for next actions. |
| Error | - | In case of 'false', no action. |
| Firing rule | Share load event with applicable-regulations (CA, CA territory) | |
| Post-condition (output state) | Cargo loaded contains the load event | |

| **State transition** | **Unload** | |
|---|---|---|
| Input state | Cargo loaded<br>Applicable regulations | |
| Event primitive | Unload event | |
| Pre-condition | (match (extract(event(visibility)_PLD (Place of Acceptance), applicable_regulations (CA territory)) and (unload_event (external_reference) equals cargo_loaded (load_event (external_reference) or (entry position (external_reference))) | The unloading is in the CA territory and the reference is part of the cargo loaded state. |
| Error | Match_function = true and (unload_event (external_reference) equals cargo_loaded (load_event (external_reference) or (entry position (external_reference))) is false | This means that the cargo is not reported in the CA territory, whilst it is was on a means of transport moving in that territory.<br>In this case, the carrier may be informed to |

| State transition | Unload | |
|---|---|---|
| | | make a correction and provide the carrier with details of applicable regulations. |
| Firing rule | Share unload event with applicable-regulations (CA, CA territory) | |
| Post-condition (output state) | If cargo_loaded (nr_of_external_references >1) then include load event to cargo loaded else move cargo_loaded (events with the external references) of the unload event to end state and include unload event. | Either the means of transport is empty (end state) or not (still in cargo loaded). |

| State transition | Position (entry) | |
|---|---|---|
| Input state | Applicable regulations | The applicable regulations with CAs and their competence domain |
| Event primitive | Position event (entry) | An entry (border crossing) event is possibly generated by on board (navigation) systems of a means of transport. This may thus not require any manual intervention. |
| Pre-condition | match (extract(event(visibility)_position, applicable_regulations (CA territory)) | This is a check that the entry position is at a territory |
| Error | - | In case of 'false', no action. |
| Firing rule | Share position event with applicable-regulations (CA, CA territory) and include all external references of the cargo onboard | |
| Post-condition (output state) | Cargo loaded contains the position event with all external references to the cargo onboard | |

| State transition | Position (exit) | |
|---|---|---|
| Input state | Applicable regulations<br>Cargo loaded | The applicable regulations with CAs and their competence domain |
| Event primitive | Position event (exit) | An exit (border crossing) event is possibly generated by on board (navigation) systems of a means of transport. This may thus not require any manual intervention. |
| Pre-condition | match (extract(event(visibility)_position, applicable_regulations (CA territory)) and and (positon_event (external_reference) | This is a check that the exit position is at a territory and the external references are part of the state cargo loaded. |

| State transition | Position (exit) | |
|---|---|---|
| | equals cargo_loaded (load_event (external_reference) or (entry position (external_reference))) | |
| Error | Match_function = true and (position_event (external_reference) equals cargo_loaded (load_event (external_reference) or (entry position (external_reference))) is false | This means that the cargo is not reported in the CA territory, whilst it is was on a means of transport moving in that territory.<br>In this case, the carrier may be informed to make a correction and provide the carrier with details of applicable regulations. |
| Firing rule | Share position event with applicable-regulations (CA, CA territory) and include all external references of the cargo onboard | |
| Post-condition (output state) | Move all events with the exit_event (external references) to end state and include exit event.. | All external references to data sets of cargo of a means of transport exiting a territory are moved to the end state. |

### *6.2 Accessible data sets*

The UUIDs of events and their referenced transport means, equipment, and goods are the basis for retrieving more information. Each CA can formulate its own queries or re-use standardized queries where the output of these queries may be different for an enterprise and a CA. Examples of those queries are:

- **Retrieve general information** – based on the UUID of an event, the order or document data set is retrieved.
- **Retrieve detailed information** – based on the UUID of an event that is retrieved via for instance the UUID of a transport means or equipment, a CA may want to receive details of the cargo of, including its agreed order (consignment data). For a CA this could be a container track.
- **Retrieve specific information of a Digital Twin** – the UUID event of load/discharge must contain the UUIDs of the relevant Digital Twins, since the user defined identifier (e.g. license plate of a truck or trailer, container number) are used to by a data user like a CA for remote surveillance. The specific information may contain details of the goods or content of a trailer/container, which is specified by the query formulated by that CA, where this goods details may only be available to a consignor and not a carrier.

The following sheet shows the data set that can be made accessible (as example) by a CA evaluating a link. This example is that of a functional data set, which is incomplete and which may require changes by a CA according to regulations. The sheet is similar to the one shown in section 2.

| Functional data requirements and mapping to the ontology | | | authorities (g2b) | | | | |
|---|---|---|---|---|---|---|---|
| **Functional** | **data properties** | **Semantic model ('concept.property')** | **road** | **air** | **inland waterways** | **rail** | **sea** |
| | User identification | | eFTI identification | AWB number | | CIM number | B/L number |
| Consignor | | Consignor | x | x | x (customer) | x (customer) | x |
| | Consignor ID | organization.ID | o | o | | | o |
| | Consignor Name | Organization.name | x | x | x | x | x |
| | Consignor Address | Location.postal address | x | x | x | x | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | x | x | x | x | x |
| | Country code | location.country code | x | x | x | x | x |
| Consignee | | consignee | x | x | | | x |
| | Consignee ID | organization.ID | o | o | | | o |
| | Consignee Name | Organization.name | x | x | | | x |
| | Consignee Address | Location.postal address | x | x | | | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | x | x | | | x |
| | Country code | location.country code | x | x | | | x |
| Carrier | | Carrier | x | x | x (barge operator) | x (RU) | x |
| | Carrier ID | organization.ID | o | o | x | x | o |
| | Carrier Name | Organization.name | x | x | x | x | x |
| | Carrier Address | Location.postal address | x | x | x | x | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | x | x | | x | x |
| | Country code | location.country code | x | x | | x | x |
| Forwarder | | Forwarder | | | | | x |
| | Forwarder ID | organization.ID | | | | | o |
| | Forwarder Name | Organization.name | | | | | x |
| | Forwarder Address | Location.postal address | | | | | x |
| | Postal code | Location.postal code | | | | | |
| | Street name | Location.street name | | | | | |
| | City name | location.city name | | | | | x |
| | Country code | location.country code | | | | | x |
| Locations (pla/pld, pol/pod, etc.) | | | | | | | |
| Goods identification | | Digital Twin-goods-identification | xor (goods or equipment) | xor (pieces) | | | |
| | | | | | | | |
| | Type of goods | Dgital Twin - goods - Type of goods | x | x | | | |
| | Gross mass (kg) | Digital Twin - goods - gross maas | x | x | | | |
| | Net mass (kg) | Digital Twin - goods - net mass | | | | | |
| | Gross volume (m3) | Digital Twin - goods - gross volume | x | x | | | |
| | Number of packages<br>Total Packages | Digital Twin - goods - number of units | x | x | | | |
| | | Digital Twin - goods - nature of the cargo | | | | | |
| | Goods description (textual) | Digital Twin - goods - goods description | x | x | | | |
| Transport Equipment | | Digital Twin - Equipment - Container | xor (see goods) | | x | xor (container or trailer) | xor (container or trailer) |
| | Transport Equipment ID | Digital Twin - Equipment - Container - ID | x | | x | x | x |
| | Transport Equipment Type | Digital Twin - Equipment - Container - type | x | | x | x | x |
| | Transport Equipment Size | Digital Twin - Equipment - Container - size | | | | | |
| | Transport Equipment Packed Status | | | | | | |
| | Seal Quantity | | | | | | |
| | Sealed Indicator | | x | | x | | x |
| | Seals | | o | | o | | o |
| | Seal number | | x | | x | | x |
| | Incident code | | | | | | |
| | Sealing Party Role Code | | x | | x | | x |
| Transport means | | Digital Twin - Transport means | x | x (flight) | x (barge) | x (train) | x (vessel or ferry) |
| | Transport means / mode type | | x (road) | x (air) | x (inland waterways) | x (train number) | x (sea) |
| | Conveyance reference number | | o | x (flight number) | | | o |
| | Transport means ID | | x (license plate truck) | o | x (barge ID) | o | x (vessel ID) |
| | Transport means Nationality | | x | o | x | x | x |
| Transport Equipment | | Digital Twin - Equipment - trailer | o | | | xor (euipment or trailer) | xor (container or trailer) |
| | Transport Equipment ID | Digital Twin - Equipment - trailer-ID | x | | | x | x |
| | Transport Equipment Type | Digital Twin - Equipment - trailer - type | x | | | x | x |
| | Transport Equipment Size | Digital Twin - equipment - trailer - size | | | | | |
| Transport Equipment | | Digital Twin - Equipment - ULD | | xor | | | |
| | Transport Equipment ID | Digital Twin - Equipment - ULD - ID | | x | | | |
| | Transport Equipment Type | Digital Twin - Equipment - ULD - type | | | | | |
| | Transport Equipment Size | Digital Twin - Equipment - ULD - size | | | | | |
| Transport Equipment | | Digital Twin - Equipment - wagon | | | | x | |
| | Transport Equipment ID | Digital Twin - Equipment - wagon - ID | | | | x | |
| | Transport Equipment Type | Digital Twin - Equipment - wagon - type | | | | | |
| | Transport Equipment Size | Digital Twin - Equipment - wagon - size | | | | | |

# 7 Final remarks

This specification of a Multimodal Visibility Service is an example of a Technology Independent Service specified according to the concepts of the architecture and using the MVP for demonstration purposes. It demonstrates how the capabilities can be applied for service design.

This Visibility Service can be validated in practical settings, supported by the MVP. Practical validation will improve the quality of the specification, i.e. its completeness and correctness.

This service can also be applied for regulations like the electronic Freight Transport Information (eFTI) regulation. In fact, is a blueprint for eFTI and its quality can be improved by piloting in Living Lab(s). The MVP is a basis for implementing eFTI and B2B visibility.

This specification shows that by configuration of nodes of participants (via the pub/sub mechanism for state synchronization as described in the architecture), physical operations are the basis for distributing visibility events to customers and for compliance (event federation). Thus, it is all about planning and instructions provided to operators (or sensors) for reporting physical activities as a basis for informing relevant stakeholders. As some logistics service providers have this functionality in place, the quality of the event federation specification will also improve by validation by pilots in Living Labs.

Event federation will never be part of any standard for visibility since it is an internal state transition of an enterprise. It can be considered a recommendation showing how value-added services can be developed.