FEDeRATED
NETWORK OF PLATFORMS

MASTERPLAN - TECHNICAL SPECIFICATION - INDEX

An index of an organization is about sending events (with links to data) in its role of data holder to and receiving in its role of data user from other organizations. Each organization must implement its own index functionality. In brief, each organization must have its own index.

1. Index functionality

The functionality of an index is to support:

- **Data correctness** and - **completeness** validation by a SHACL validator configured from an organizational profile.
- **Linked-based access control** by validating that a data user can only access data of a link that is shared with that data user. Access is provided to data properties of these links that are defined by a query (these can be business document data sets or subsets for compliance to a regulation as specified in a Service Registry).
- **Query federation** for accessing data of a link that is stored by a third-party acting is the data source.
- **Event logic** to validate the state transitions modelled by an organizational profile.
- **Event distribution** for sharing event data with the proper nodes. The distribution rules are specified as follows:
  - **Compliance** – an authority specifies the events of a (subtype of a) business activity that it requires, potentially with links to additional data (e.g. business document data set). This type of event distribution can be configured with for instance a publish-subscribe mechanism by an enterprise acting as data holder to an authority.
  - **Commercial relation** - an initial event of a business transaction must provide details of its recipient. Any events following this initial event are shared between customer and service provider in this commercial relation.
  - **Leg synchronization** – synchronization of adjacent legs (previous or next leg) is automatically if expected/planned/required time windows are not exceeded. If some leg exceeds its time window for finishing, a business transaction of that next leg may have to be cancelled (as specified in a choreography).[1]
- **Data sharing** for sharing events between two stakeholders. This includes data agnostics protocols for a safe, secure, and reliable sharing of any type of data. Data sharing is based on existing open (and defecto) standards like TLS (Transport Link Security), https, ebMS (electronic business Messaging Service), message queueing, and many others.

[1] *Leg synchronization requires knowledge of the chain structure. Based on the chain structure representing commercial relations, service providers of adjacent legs may synchronize their activity via a shared node (customer) in a transaction hierarchy.*

The execution of a leg maybe prone to a condition, where the condition provided by another stakeholder. Examples of conditions are payment of transport costs and customs release. The condition is shown by a decoupled transaction and must be shared with service providers performing a leg. Data integrity of the condition must be assured (see IAA).

- **Non-repudiation** providing the capability of immutable proof for sharing data with another stakeholder (both sending and receiving).

To support this functionality, all events with links to data that are shared (send and received) must be available, i.e. stored somehow.

2. The implementation of index functionality and data

An organization can implement the functionality of an index in its internal IT systems. To enable plug and play, a separate index with the semantic model and all functionality using semantic web technology must be implemented.

All indexes share data using semantic web standards (RDF, SPARQL) amongst each other utilizing a connectivity protocol and implementing IA. An index can have openAPI endpoints for integration with other environments, implementing a connectivity protocol for those environments.

The functionality and data of an index can be implemented in different ways, thereby supporting an organizational profile of an organization as specified in its service registry. Thus, an organizational profile specifies the complexity of the required functionality.

The following options are feasible:

- **Internal** – all functionality and data of an index is implemented by internal IT systems.
- **External** – all functionality and data described in this document is supported by the index.
- **Mix of internal and external** – such a variant may support for instance data quality validation on events, queries and query results, access policy validation, and simple variants of event distribution and – logic. Chain management and other types of functions are implemented outside scope of the index.

Each organization must decide itself what to implement in the index or in its internal IT environment.

3. Index data

Data remains at the source and events with links to data are shared in a commercial – or legal relationship between any two stakeholders. These events are part of a design (see Service Registry). Thus, events (with links) can support any type of interaction specified by a design like a transport order, ETA event or load event.

To support various functions, an index also must store state information based on events (and links to data) shared with peers either as customer and/or service provider. A state can be derived from all events related to a state transition, but this may decrease performance. Thus, it is recommended to store state data separately as an aggregation of all events.

State transitions are performed by executing pre- and post-condition of choreography processes. These conditions must be specified by a design.

The process aspects of the semantic model define the way to visualize business transaction hierarchies by transaction trees. Each index must provide a means to extract transaction dependency. This is relevant to automatically share updates with customers and service providers of different legs in a chain. For any given index, this is visualized in figure 1.

Figure 1 illustrates how a forwarder uses an index relating the business transactions with its shippers in their roles as data users to its business transaction with its service providers acting as its data users. The depicted transactions are set in a chain structure. This chain structure can be derived by relating the start of a business transaction of a shipper to a transport leg that has the same start (time and place) and the end to a business transaction with a transport leg likewise. All legs are associated by relating their end to a start of the next leg (place, time). To improve performance, the relation between business transactions can be stored separately in the index.

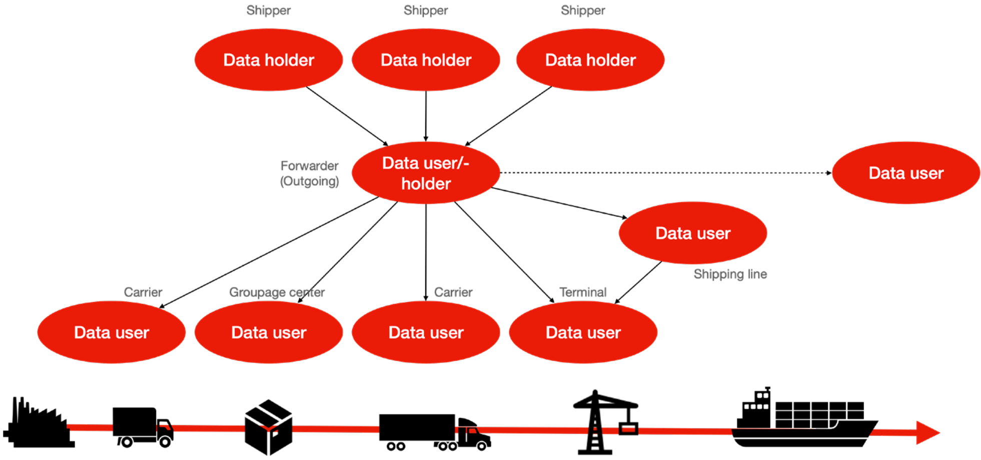

Figure 1 – transaction dependency for a node in chains (complex example outgoing cargo)

A chain structure can only be supported if the organizational profiles of any two organizations participating in a business transaction have at least one transaction phase in common e.g., visibility including its start