FEDeRATED Reference Architecture Adoption and deployment phases

INTRODUCTORY REMARKS

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

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

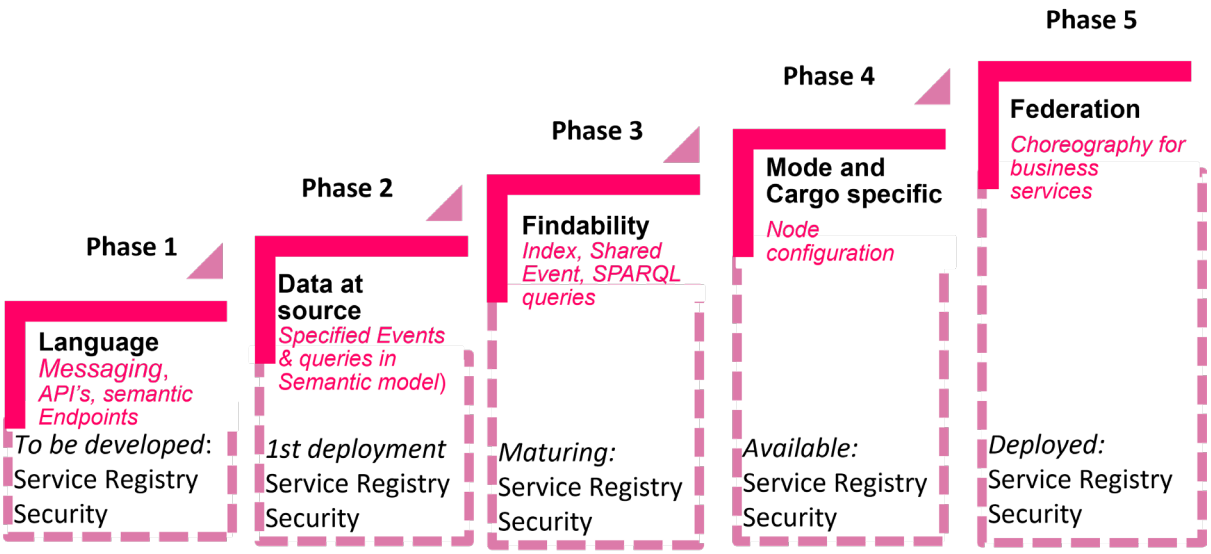
2. **Data at the source** – this phase is about specifying events and queries with the semantic model. They can be deployed according to the 'language' and the 'findability' phase.

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

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

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

The phases are illustrated hereunder:



1. LANGUAGE a distinction of specification and deployment

This phase distinguishes between specification and deployment:

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

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

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

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

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

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

The solution also has **threats** like:

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

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

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

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

2. DATA AT SOURCE

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

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

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

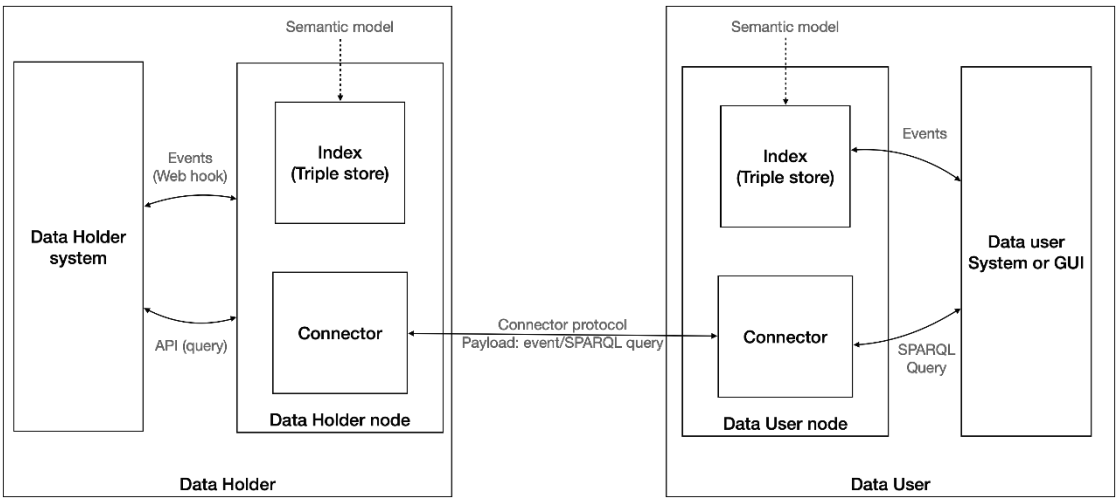
In this phase, the Service Registry requires specific functionality to specify 'events'. Those are called 'user-events': they combine more than one atomic event of the semantic model into an event that has meaning to participants in a use case.

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

3. FINDABILITY – IMPLEMENTING THE SEMANTIC MODEL AND INDEX

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

The following figure depicts the implementation:



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

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

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

The **strengths** of this solution are:

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

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

- Complexity of local interface. Local REST APIs must be aligned with SPARQL queries by a data holder. This decreases flexibility of the solution. This may imply that the number of REST APIs for local interfaces may increase by the number of SPARQL queries to be supported.
- New technology. New technology needs to be implemented in the domain of a participant.
- Lack of interoperability. The interoperability between LLs and use cases is not guaranteed since each LL/use case can specify its events and queries.
- Local interface – semantic adapter. The semantic adapter may require additional functionality for implementing a local interface with an IT system. Data structures of internal interfaces may not be identical to the structure provided by the semantic adapter, code values may differ, etc.
- Query federation. The query federation mechanism needs to be implemented for optimal application of the 'data at the source' principle.

The **opportunities** are:

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

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

4. MODE AND/OR CARGO SPECIFIC

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

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

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

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

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

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

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

The **threats** are identical to those of the previous phase, with the exception that knowledge of

the semantic model is hidden to users. However, they still have to be able to support complex, ad-hoc queries.

5. FEDERATION – TECHNOLOGY INDEPENDENT SERVICS AND PLUG AND PLAY

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

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

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

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

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

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

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

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

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

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

Threats are basically in intermediation of existing solutions (i.e. platforms and community systems) with their existing business models. There is no requirement for using more than one platform or community system if the TIS are available and deployed by those solutions. One can simply call the TIS locally and have business transactions with all others involved. Another potential thread is disintermediation of the role of Industry Associations. Standardization of TIS and plug and play does not require any mode specific solutions.

Regulating bodies: standards and the phases

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

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

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

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

More than one semantic model

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

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

(FEDeRATED) other ontology. This might be required for One Record that is specific to airway bill data.

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

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