MIGRATION STRATEGY

The federative network of platform concept stimulates any company or public authority involved in supply and logistics chain operations to evolve as a Node in a data sharing network. This network can be defined as an *infrastructure provision containing a set of agreements and technical applications to enable data in existing IT systems (platforms) of companies and public administrations to become available to authorised users through a publish and subscribe approach*. The technical applications can also be called capabilities enabling companies and public administrations to provide services.

Gradual development and implementation of capabilities and services enabling any organisation to act as Node fosters the smooth transition towards an federated infrastructure provision Initially, no capabilities (or their prototypes) may be available for organisations, let alone services (or maybe services are available in proprietary formats). Whenever services are available in proprietary formats, these can be transformed into formats required by the infrastructure provision (easily).

A community (or project) can develop (prototypes of) capabilities and develop or transform services for end-users. Since there might be multiple communities working in parallel, quality assurance between these communities must be established to prevent creation of community-based dataspaces.

Thus, a distinction is made in migration of a community and an individual organisation. A migration strategy should incorporate the following considerations:

- Start small, grow big – this is about the scale of adoption. Start with a small community of opinion leaders with the intention to scale to a larger one. The small community must have the ability to validate all capabilities for an appealing business case.
- Add value – this is about the business case of not only end-users, but also that of Service Developers and technology providers. Where the first is about a logistics business case, the latter is about implementing the Service Registry.
- New services – start with services that are not yet implemented by the community (and add value). A visibility service is an example; capacity sharing services could be another example. A visibility service may not yet require a ‘profile’. Gradually, ‘profiles’ can be introduced by replacing existing data sharing standards with services.
- From experiment to application – this is about validating the capabilities of the infrastructure provision in their business context. It can imply to start with a node and migrate to a gateway or decide to develop an own adapter. It can also imply to partly implement the Service Registry functionality for service development and – customization.

- Hide complexity – provide a common set of interfaces of the infrastructure provision to end-users. It enables ad hoc technology selection with the ability to migrate to horizontal standards.
- Best practices – re-use of what others have developed and learned. This is about service re-use and its customization to users in a community (creating profiles).

Community migration – capability development

Figure 1 shows the migration phases for a community. It is about development and validation of capabilities by end-users for use cases.

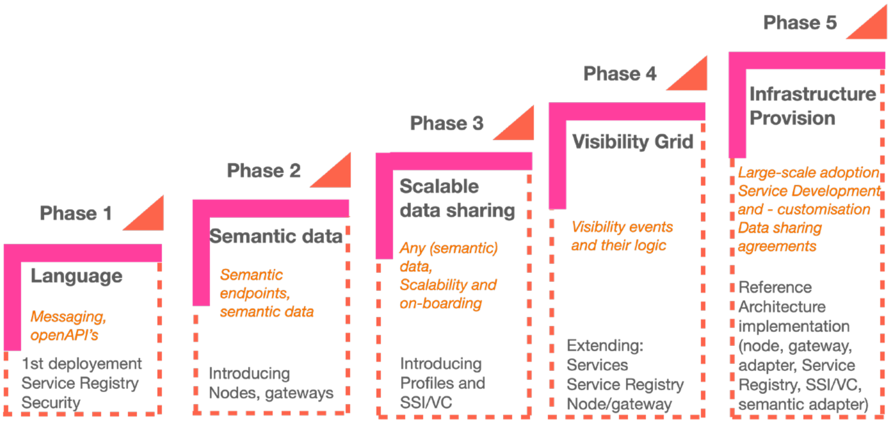

Figure 1 Migration path for a community

Phase 1 - Language

In this first phase, the semantic model for generating openAPIs for services required by a community is applied, covering existing services and possibly new services. A service can be about sharing the data set of a business document or cover multimodal supply chain visibility. It is advised toto start small in this phase, thus applying existing Identification and Authentication means and reducing complexity of the great many APIs that must be integrated. The openAPIs produced by the Service Registry are based on considerations given in section 6.3.

Phase 2 – Semantic data

The second phase is about introducing semantic technology implemented by a node or gateway hiding complexity and the Index APIs supporting the functionality developed in the first phase. These Index APIs are implemented at the interface of an end-user with the infrastructure provision, i.e. a node (section 7.2). This reduces complexity by limiting the number of APIs that must be implemented.

The first implementation of node (or gateway) can be based on ad hoc selection of capabilities and new versions can be constructed to support horizontal standards. Updates of this software always have at least the Index APIs (or a relevant subset), which means that updates don't have implications for an end-user.

Phase 3 – scalable data sharing

The third phase covers scaling, including the implementation of an SSI/VC based Identity and Authentication infrastructure, and implementation of the Index APIs for data pull mechanism with semantic technology. Accommodating policies based on data sharing agreements and its legislation provide for a sound governance framework.

Phase 4 – Visibility grid

The fourth phase is about implementing new services like a multimodal visibility service. It is about re-use and harmonization of overlapping services and applying all concepts of the data sharing ontology resulting in event logic for a node and gateway implementation. A multimodal supply chain visibility is an example of such a new service, shown in the figure. Nodes and gateways can still have the interfaces (openAPIs) with IT systems of end-users as in the third phase, extended with openAPIs for event logic.

Phase 5 – Infrastructure Provision

In the fifth phase, all capabilities of the infrastructure provision and sufficient services to support organisations in data sharing should be made available . Each end-user can decide to implement the Index functionality by a standard node or gateway or develop their own adapter.

End-user migration

The objective is the migration of an end-user, an individual organisation, to fully implement their capabilities. It requires technology support of these capabilities (developed in a community, see migration of a community) and availability of services.

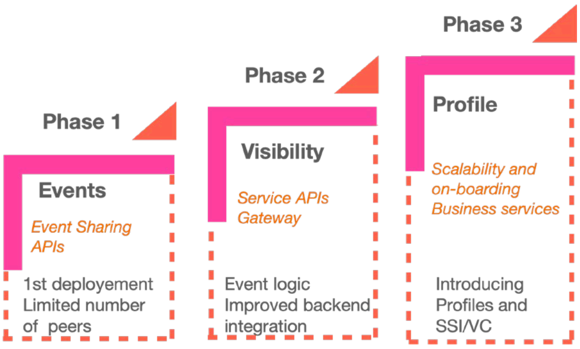
Phase 1
Events
Event Sharing APIs
1st deployement Limited number of peers
Phase 2
Visibility
Service APIs Gateway
Event logic Improved backend integration
Phase 3
Profile
Scalability and on-boarding Business services
Introducing Profiles and SSI/VC

Figure 2 Migration path for an individual organisation

Phase 1 – Events

This first phase covers the sharing of events; - the implementation of the event sharing functionality of the Index API preferably with a node implementation with a GUI. In a pilot, a node with a GUI can be used without further integration with IT backend systems. The Index API hides complexity of semantics, it is fully configured for events of a service. Links to additional data can be shared and access to that data is via the Index APIs.

Phase 2 – Visibility

The second phase is about implementing the logic of a service and extending the functionality of the Index API. It can also be on integration with IT backend systems via a gateway solution that provides a mapping of internal APIs and Index APIs.

End-users may already have IT facilities for visibility like events with their APIs and a web interface provided to their customers. These APIs can be matched to events of a visibility service. Event logic will most probably be implemented in IT backend systems and needs to be matched with that of a visibility service.

Phase 3 – Profile

Where the previous phases are experimenting and implementing functionality with a limited number of peers, the third phase is about becoming a full end-user of the infrastructure provision. It is about the support of 'profile' with Verifiable Credentials (VC)., Enterprises, acting as service providers, publish their business services. Authorities publish their data requirements in the context of regulations. This third phase requires complete functionality of the Service Registry for development and discoverability of service specifications of Service Developers.

Since Phases 1 and 2 fully implement sharing of events with links to data and event logic, the third phase can support services complementary to visibility like booking and ordering.

General observations

There are some common statements to be made with respect to migration, namely:

- Don't wait. Communities can always start by adopting semantics for generating openAPIs. This is the first phase for a community. Organisations can start with the first phase with the Index API supported by the prototype node[1] and Service Registry.
- Continue after Phase 1 to the next Phases, even after introduction of SSI/VC in Phase 3 for communities. Phase 1 is only applicable for a small community. A large community as all logistics enterprises in the EC cannot manage the implementation of 'a forest of' many openAPIs, even in most of them are functional identical.
- Sticking to phase 1 for a community is only applicable with a limited number of platforms. This reduces the 'API forest'.
- Supervising bodies need at least a Phase 3 implementation (organisational migration). They must supervise a many logistics enterprise, including SMEs. Thus, managing 'an API forest' (Phase 1 for a community) comes with too high costs.
- End-users require service re-use and harmonisation. This is fully supported by at least Phase 3 and Phase 5 for all services.
- Especially (a community of) first movers i.e., the opinion leaders, will develop capabilities for the first four phases collaborating for data sharing agreement legislation. Followers and laggards will adopt these capabilities according to the migration phases of an individual end-user.

[1] See: Node prototype and installation, incl codes. The latest version of the node prototype and updated documentation can be found at: https://github.com/Federated-BDI/FEDeRATED-BDI
Updated Docker installation instructions are available at: https://github.com/Federated-BDI/Docker-BDI-Node

- Platform – and COTS software providers can be amongst the first movers or communities for developing capabilities. Especially platform providers can figure out the impact on their business model, starting in phase 2.
- Platform – and COTS software providers are most likely to develop adapters to their system. This provides optimal support of non-functional requirements.
- As end-users, SMEs require standard (COTS) solutions or platforms with standard applications. These provide large-scale adoption.
- Service developers and -customizers require a testing environment for testing the technical implementation.
- Service developers also require first movers for new services; these are most probably the end-users that require these new services.

Considerations for a pilot / Living Lab

Moving towards a migration path, communities can be organized as pilots or Living Labs. The various issues to be explored in a Living Lab, including one or more use cases or pilot relate to Scope, Stakeholder Engagement, Technical Setting, Testing and Impacts. [2]

[2] For this purpose FEDeRATED developed a Living Lab project book, containing the various items: LL Project Book (federatedplatforms.eu)