LIVINGLAB ASSESSMENT FRAMEWORK - VALIDATION CRITERIA

Introduction

FEDeRATED has developed a reference architecture with functional requirements and technical specifications (technology independent). The technical specifications are supported by technical components (implementation).

Many FEDeRATED LivingLabs (and individual stakeholder) have implemented a technical setting that will be validated against the FEDeRATED technical specifications, using its own technical components. Thereto an assessment framework has been developed. This framework is elaborated in various tables in this document. The tables contain information about:

1. The technical specifications and the technical components
2. The weighting scale regarding the technical components, based on which every LL can identify its compliance.
3. The non-functional requirements
4. The weighting scale regarding the non-functional requirements

The FEDeRATED IT architecture development process has led to an architecture of a 'federated network of platforms' or possibly an EU Mobility 'data space':

1. Vision is detailed into 37 leading principles.
2. Leading principles are supported by functional requirements. One functional requirement can support one or more leading principles and a leading principle can affect one or more functional requirements.
3. Technical specifications detail the functional requirements, or rather they indicate what capabilities a LivingLab or node should comply with.
4. Technical specifications lead to technical components. Their functionality is specified in more detail.

In line with the standard for IT architecture, TOGAF these aspects cover the vision, business architecture (leading principles), information architecture (language: data and processes), and technology architecture, where the latter is not completely covered since the technical specifications and the functionality of the components is technology independent.

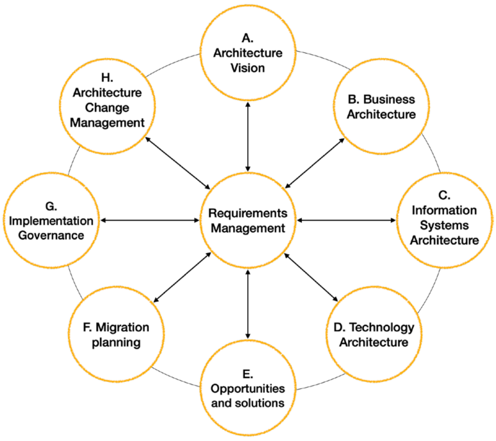
A. Architecture Vision
B. Business Architecture
C. Information Systems Architecture
D. Technology Architecture
E. Opportunities and solutions
F. Migration planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

Illustration: The TOFAF IT architecture standard

LIVINGLAB ASSESSMENT FRAMEWORK - VALIDATION CRITERIA

Hereunder the various tables based on which the technical setting of every LivingLab can be validated against The FEDeRATED Architecture will be validated against the LivingLabs. It is a two-way street.

1. Description of the technical components

TECHNICAL SPECIFICATIONS - CAPABILITIES		
No	**Technical component**	**Description**
1. SEMANTICS		
1.1	**Semantics - specification**	Specification of the data that can be shared by all stakeholders. The specification may take various forms: • A model per interaction • A consignment/ shipment based model • A model for all data that can be shared. Such a model can also have various forms, e.g. an ontology, a class diagram, or a hierarchical structure (similar to XML structures)
1.2	**Interaction pattern**	The structured sequence of interactions. There are different options: • There is only a single interaction (e.g. a data representation of a business document) • Sequencing is represented by sequence diagrams for the use case (chain) • Sequence diagrams for any two stakeholders • Support of (part of the) normal operation, for instance booking, ordering, and/or visibility Interaction patterns can also be specific to a particular business activity like transport of containers by rail. Interaction patterns are the technology independent services, e.g. a booking -, ordering - , and visibility service. These interaction services can be implemented differently, e.g. with multiple openAPIs and as triples (RDF), see later questions.
1.3	**Modeling alignment or -mapping**	In case a LL has developed its own model, the model can be aligned or mapped to the FEDeRATED semantic model: • Alignment – identifying overlapping concepts and data between two models • Mapping – construct an overlap of a LL model with the FEDeRATED model Alignment is achieved via a representation of a LL model as ontology, most probably as a manual exercise. Mapping can be supported by technical components like a mapping tool and a semantic adapter, see next questions.

TECHNICAL SPECIFICATIONS - CAPABILITIES		
No	**Technical component**	**Description**
1.4	**Access policy specification**	Specification of access policies. Access policies are required in case of a data pull. As such they are specified by the individual interactions taking the relevant parts of the semantic model that is applied by a LL. In case of data push, no specific access policy is required; a message supporting data push contains for instance all data that is duplicated. The syntax and technology (messaging, (open/webhook) APIs (Application Programming Interfaces) with JSON(-LD) (Java Script Object Notation – Linked Data), semantic web protocols (SPARQL (Standard Protocol and RDF Query Language), RDF (Resource Description Framework))) used for sharing data.
2. SERVICE REGISTRY		
2.1	**Modelling toolset**	The capability to specify and publish the organizational profile of a user participating in a Living Lab. An organizational profile must refer to a LL model and/or the interactions that are applicable for the LL. The latter could be formulated by for instance APIs or standards applied for data carriers. The capabilities must be accessible for rapid on-boarding and upscaling of a use case to new users.
2.2	**Organizational profile**	The technical component(s) for a user to configure and publish its organizational profile. These tools should refer to capabilities like import/export of models and must support open standards. An openAPI environment like Swagger can be an example of publishing openAPIs with their endpoints.
2.3	**Toolset to construct and publish an organizational profile**	The syntax applied for sharing data. Options are: XML, EDI(fact), JSON(-LD), RDF, or a proprietary format.
2.4	**Syntax**	The technological paradigm to share data messaging, (open/webhook) APIs, etc. In case APIs are applied, the toolset to publish an organizational profile will be probably an environment like Swagger.
2.5	**Technology**	Use of an (open/defacto) standard for sharing data, This can be any standard (GS1, UN CEFACT, other) and/or a specific implementation guide of a standard (e.g. UN CEFACT eCMR, DCSA eB/L, etc.). Please mention.

TECHNICAL SPECIFICATIONS - CAPABILITIES

No	Technical component	Description
2.6	**Data carrier / standard**	A technical component that transforms data between an external syntax/data carrier to another, where the latter is mostly an internal format. The semantic adapter is a specific implementation where RDF is used as external format and needs to be integrated with existing standards, technology, or databases. This can be via so-called RDF plugins, RML (Rule Markup Language) tools, etc.
2.7	**Data transformation (semantic adapter)**	A technical component to configure data transformation. Data transformation can be supported by mapping tools. Examples are those provided by integration brokers/enterprise service busses; others are so-called RML mappers. LLM (Large Language Models) can also be considered, although they are still in an experimental phase.
2.8	**Data mapping tools**	A users' view of events that are received from or send to other users. Event storage is required in case events have links to additional (upstream) data. It supports data provenance and authorization. Event storage can be part of a log and audit trail for non-repudiation.
3. INDEX		
3.1	**Event storage**	Rules for sharing events with another user. Event distribution can be implemented in different ways, for instance based on a legal obligation (mandatory) or a commercial relation (dynamic configuration). A user may apply publish/subscribe, where the subscription is configured by the one that publishes the events.
3.2	**Data validation**	Validation of agreed interaction sequencing. Validation is only applicable in case multiple interactions and their sequencing is defined
3.3	**Event distribution**	The right to access data and use functionality This is about data provenance: links to data are passed between stakeholders and need to be accessible downstream. Delegation might be a mechanism for avoiding query federation but is considered to be static.
3.4	**Event logic**	Access to data by a data user via an intermediary acting as data holder to the data user. This is about data provenance: links to data are passed between stakeholders and need to be accessible downstream. Delegation might be a mechanism for avoiding query federation but is considered to be static.

TECHNICAL SPECIFICATIONS - CAPABILITIES		
No	**Technical component**	**Description**
3.5	**Authorization**	A technical component for presentation of data presentation to a human. A (temporary) GUI might be provided in case full integration with existing IT systems is not yet feasible. The GUI will include data validation functionality (see Linked Event Protocol).
3.6	**Query federation**	The technical capability for reliable, safe, and secure data sharingwith a (defacto) standard. Current list of connectivity protocols: FENIX connector protocol, IDSA connector protocol, EDS (Eclipse Data Space) protocol, Message queueing protocols (like AMQP), blockchain protocols (like Baseline, Hyperledger Fabric, Ethereu), and AS4 implemented by CEF eDelivery. Note: not all data sharing implementations require a separate connectivity protocol since they may use a openAPIs wit https/TLS.
3.7	**Graphical User Interface (GUI)**	The technical component (and its vendor or open source/freeware) implementation of a single or multiple (layered) protocols. Please be aware that even if the protocols are identical, their implementation by a component is not necessarily interoperable with an implementation of another component.
3.8	**Connectivity protocol**	The immutable proof that data is shared. An implementation is by a log and an audit trail. It contains all data that is shared according to the presentation protocol (events, messages, queries, etc.). Although there may not be a specific connectivity protocol, there may still be a log and audit trail.
3.9	**Connectivity component**	The connectivity between various stakeholders should be supported by an individual user In case an external agreed protocol is implemented, this might not be supported by existing systems and solutions. For instance, APIs using https may have to be mapped to the eDelivery or IDS protocol.
3.10	**Non-repudiation**	The safe and secure sharing of data with PKI certificates, utilizing standard protocols (e.g. https, TLS).
3.11	**Internal connectivity**	Unique identification and authentication of users (organizations). Use of open standards like OAUTH2.1, Verifiable Credentials (VCs) and Decentralized Identities (DIDs), JWT (JSON Web Tokens), or others.
3.12	**System security protocol**	The right to access data and use functionality. This relates to access policies (see before) and is supported by index functionality like event storage and - distribution. In case an event storage and - distribution are not implemented by a technical component, authorization must be defined separately.

TECHNICAL SPECIFICATIONS - CAPABILITIES		
No	Technical component	Description
4. IDENTIFICATION, AUTHENTICATION, AND AUTHORISATION		
4.1	Identity and Authentication (IA)	Specification of the data that can be shared by all stakeholders. The specification may take various forms: • A model per interaction • A consignment/ shipment based model • A model for all data that can be shared. Such a model can also have various forms, e.g. an ontology, a class diagram, or a hierarchical structure (similar to XML structures)
4.2	Authorization (other than link)	The structured sequence of interactions. There are different options: • There is only a single interaction (e.g. a data representation of a business document) • Sequencing is represented by sequence diagrams for the use case (chain) • Sequence diagrams for any two stakeholders • Support of (part of the) normal operation, for instance booking, ordering, and/or visibility Interaction patterns can also be specific to a particular business activity like transport of containers by rail. Interaction patterns are the technology independent services, e.g. a booking -, ordering - , and visibility service. These interaction services can be implemented differently, e.g. with multiple openAPIs and as triples (RDF), see later questions.
4.3	Distributed versus centralized implementation	In case a LL has developed its own model, the model can be aligned or mapped to the FEDeRATED semantic model: • Alignment – identifying overlapping concepts and data between two models • Mapping – construct an overlap of a LL model with the FEDeRATED model Alignment is achieved via a representation of a LL model as ontology, most probably as a manual exercise. Mapping can be supported by technical components like a mapping tool and a semantic adapter, see next questions.

2. Measuring against the technical components – scoring/weighting

No	TECHNICAL COMPONENT	SCORING APPROACH – VALIDATION		
		Low	Medium	High
SEMANTICS				
1.1	**Semantics - specification**	A model per message/interaction	Proprietary model	FEDeRATED model as basis
1.2	**Interaction pattern**	Single interaction between stakeholders	Message sequence diagrams	Interaction patterns specifiying interaction sequencing between two participants in a business transaction for a business activity. Please mention which you support and from which perspective (visibility of a transport means or cargo, booking a shipment, etc.)
1.3	**Modeling alignment or - mapping**	Users must implement the data carriers and semantics developed for the use case.	Mapping with FEDeRATED model, implying data can be expressed in the semantics of ones' own model and the common ontology. Users can select to implement the data carrier and semantics of either the use case or provided by the common ontology.	Alignment with the FEDeRATED model, meaning that common concepts and properties in two aligned models are part of the upper ontology. Users are able to implement both the functionality of the common ontology and that of the specialization.
1.4	**Access policy specification**	Data push based on peer-to-peer solution	Platform arranging Identity and Access management based on message structures	Access policies related to interaction patterns with business transaction states and events for state synchronisation
SERVICE REGISTRY				
2.1	**Modelling toolset**	Technical level (e.g. API toolset like Swagger)	Technical and functional level (metadata related to openAPIs)	Technical, functional, and business level (business activities, business services)

No	TECHNICAL COMPONENT	SCORING APPROACH – VALIDATION		
		Low	Medium	High
2.2	**Organizational profile**	Unstructured (word, excel, drawing tools, etc.)	Proprietary toolset based on the solution/platform for publishing the profile	Toolset supporting the agreed structures for specifying a profile
2.3	**Toolset to construct and publish an organizational profile**	Proprietary format	One of the selected options (XML, EDI, JSON)	Full support of RDF/JSON-LD
2.4	**Syntax**	(EDI/XML) messaging	openAPIs	openAPIs, webhook APIs, SPARQL endpoint(s)
2.5	**Technology**	proprietary data carrier	support of an open, standard/defacto data carrier (including its potential subset like an eCMR based on UN CEFACT)	Structures in a syntax (RDF(s) or JSON-LD) directly integrating with a semantic model
2.6	**Data carrier / standard**	only a selected data carrier is supported, no data transformation	Data transformation to a selected number of data carriers	full support of data transformation to other data carriers
2.7	**Data transformation (semantic adapter)**	no tools, hardcoded data transformations	data transformation tools supporting the selected technology(-ies)	(semi-)automatic tools based on ontology alignment and matching
2.8	**Data mapping tools**	Events are directly derived as such in existing IT systems	Separate storage of events in existing IT system	Events that are shared are explicitly stored in a separate database or other mechanism (e.g. triple store)
INDEX				
3.1	**Event storage**	An event distribution mechanism implemented by internal data processing policies supported by humans	Support of pub/sub configurable by any data user/peer organization	(semi-)automatic distribution of events based on rules in all relevant commercial transactions and for compliance (implemented by for instance pub/sub), triggering by events that are received from stakeholders.
3.2	**Data validation**	Simple event logic based on order level	Validating progress of	Event logic based on common agreements of

No	TECHNICAL COMPONENT	SCORING APPROACH – VALIDATION		
		Low	Medium	High
		(order centric operation with for instance consignment/shipment identifier)	logistics operation based on time and place of the execution of the transport of a consignment/shipment	interaction patterns reflecting real world states (Digital Twins, infrastructure)
3.3	**Event distribution**	Authorization defined by a data holder receiving a query of a data user	Authorization by a data holder to access data is based on a link that is shared. Only access to the data holders' data	Authorization by a data holder to access data based on a link that is shared with a data user and a link that is received from another data holder (query federation)
3.4	**Event logic**	A data user duplicates data and makes it available as data holder to another data user	Manual evaluation a data holder of a query received from a data user, resulting potentially in a (manual) query to another data holder	IT capability by a data holder to combine internal data and data at the source upon a query of a data user
3.5	**Authorization**	simple (data carrier based) GUI	GUI functionality for one or more employee roles to support data sharing.	Integrated in the GUI (and processing functionality) of internal IT systems
3.6	**Query federation**	proprietary protocol	support of a single agreed protocol based on open/defacto standard(s)	support of more than one protocols (based on open/defacto standards) common to relevant relations (business relations, authorities)
3.7	**Graphical User Interface (GUI)**	a proprietary developed component	a single (open source/freeware/vendor) component	multiple (open source/freeware/vendor) components
3.8	**Connectivity protocol**	up to each organization to decide upon	a shared community component (e.g. a clearing house as identified in the IDSA reference architecture)	each paritcipant must implement non-repudiation functionality
3.9	**Connectivity component**	a single prescribed interface between a gateway/node/etc. to an internal IT system	more than one interface (e.g. open/REST API and webhook API)	Completely free, supported by for instance a gateways solution or enterprise

No	TECHNICAL COMPONENT	SCORING APPROACH – VALIDATION		
		Low	Medium	High
		(for instance an open/REST API)	supported by for instance a gateways solution or enterprise service bus acting as gateway	service bus acting as gateway
3.10	**Non-repudiation**	no link security	support of https with eIDAS certified PKI-certificates	support of TLS with eIDAS certified PKI-certificates
3.11	**Internal connectivity**	Peer-to-peer data sharing between known organizations only	IA is specific to a community	IA is independent of any business collaboration and reporting to authorities
3.12	**System security protocol**	Proprietary rules specified between any two peers that share data	Common rules specified by a community. These may include delegation	Common rules for commercial transactions and compliance implemented by stakeholders
IAA		*Completely centralized solution*	*Centralized solution with peer components interfacing with the central solution*	*A combination of centralized and distributed solution*
4.1	**Identity and Authentication (IA)**	A model per message/interaction	Proprietary model	FEDeRATED model as basis
4.2	**Authorization (other than link)**	Single interaction between stakeholders	Message sequence diagrams	Interaction patterns specifying interaction sequencing between two participants in a business transaction for a business activity. Please mention which you support and from which perspective (visibility of a transport means or cargo, booking a shipment, etc.)
4.3	**Distributed versus centralized implementation**	Users must implement the data carriers and semantics developed for the use case.	Mapping with FEDeRATED model, implying data can be expressed in the semantics of ones' own model and the common ontology. Users can select to	Alignment with the FEDeRATED model, meaning that common concepts and properties in two aligned models are part of the upper ontology. Users are able to implement both the functionality of the common

No	TECHNICAL COMPONENT	SCORING APPROACH – VALIDATION		
		Low	Medium	High
			implement the data carrier and semantics of either the use case or provided by the common ontology.	ontology and that of the specialization.

3. The non-functional requirements

NON FUNCTIONAL REQUIREMENTS		
No	Requirement	Description
1	**Performance**	i.e. the system's ability to respond to user requests in a timely and efficient manner. It includes factors such as response time, throughput, and scalability.
2	**Performance efficiency**	i.e. the system's ability to use resources (such as memory, CPU, and network bandwidth) in an optimal way. It includes factors such as efficiency, speed, and optimization.
3	**System security**	i.e. the measures taken to protect the system and its data from unauthorized access, modification, or destruction. It includes factors such as data encryption, access control, and authentication.
4	**Reliability**	i.e. the system's ability to perform its intended functions without failure over a period of time. It includes factors such as fault tolerance, error handling, and disaster recovery.
5	**Maintainability**	i.e. the ease with which the system can be modified, repaired, or enhanced over time. It includes factors such as modularity, documentation, and code maintainability.
6	**Usability**	i.e. the system's ability to be used effectively and efficiently by its intended users. It includes factors such as ease of use, accessibility, and user satisfaction.
7	**Availability**	i.e. the system's ability to be accessible to its users whenever they need it. It includes factors such as uptime, downtime, and service level agreements (SLAs). This also relates to MTBF (mean time between failure) and a contingency plan. It can also be the failure of a single component of one stakeholder in its role of data holder. Indicate mechanism/means for testing and expected form of results.

NON FUNCTIONAL REQUIREMENTS		
No	**Requirement**	**Description**
8	**Scalability**	i.e. the system's ability to handle increasing amounts of data, traffic, or users over time. It includes factors such as horizontal scaling, vertical scaling, and load balancing. This is of relevance in the case of a single platform; a P2P environment can probably handle more. Indicate aspects/means for testing and expected form of results.
9	**Compatibility**	i.e. the system's ability to operate with other hardware, software, or systems. It includes factors such as interoperability and compliance with industry standards.
10	**Contingency plan**	i.e. any fallback procedures when (crucial) systems components fail. Are there procedures, and if so outline type of procedures and to be tested aspects.
11	**Onboarding**	i.e. procedures for including new stakeholders to the LL. Are there procedures, and if so outline type of procedures and to be tested aspects.

4. Scoring against the non-functional requirements

		SCORING APPROACH		
No	**Requirement**	**Low**	**Medium**	**High**
1	**Performance**	Not considered	Performance per use case	Fully support of performance requirements required by individual stakeholders
2	**Performance efficiency**	Not considered	Manual intervention	Dynamically scalable
3	**System security**	not implemented	a limited number of measures taken	full system security (data encryption, access control, authentication, etc.) and cyber-security measures
4	**Reliability**	Not considered	a limited number of measures taken	Reliable system according to requirements
5	**Maintainability**	Not considered	Manual intervention required	Automatic distribution and avialability of updates

6	**Usability**	Not considered	Preset options provided	fully configurable to a users requirements
7	**Availability**	Not considered or requires manual intervention	Limited availability, no testing capabilities	24x7 availability supported by a published MTBF and a contingency plan, testing facilities provided
8	**Scalability**	An implementation based on predefined scalability requirements	Scalability for all users (manual/dynamical; central solution)	Dynamically scalable by each user (distributed implementation)
9	**Compatibility**	Applicable for a single type of hardware/OS	available for a predefined set of hardware/OS solutions	fully portable, independent of hardware/OS
10	**Contingency plan**	No contingency plan	fallback procedure with impact to a user (e.g. based on a central solution)	fallback procedures to provide 24x7 operation without impact to a user, operational for each user
11	**Onboarding**	Onboarding a user influences the configuration of all other users (bilateral agreements)	Onboarding of each user with installation requirements and data distribution to access capabilities of other users	Onboarding of each user with full (automatic) data sharing capabilities to all other users