Basic principles of the "Living Lab Toolbox"

Johann Sievering

Summary

The generic nature of Livings Labs (LL) allows them to invest almost all sectors. Each LL is a special case that requires tools and specific operational methods. It is probably for these reasons that few ICT applications (information and communications technology) have been developed for the LL. The Living Lab Toolbox (LLTB) project proposes an ecosystem of tools to dynamically build a digital environment adapted to each LL.

This article introduces the basic concepts of LLTB, but will not address the ontological constructions of either the reference frame nor the domain. We will not present the technical aspects either. These topics will be covered in other articles.

First, we will define what LLTB is and its relevance in the context of LL. In a second step, we will present the general concepts of LLTB. In a third step, we will analyse the technological aspects of LLTB by presenting the general architecture and the key technical points to achieve all the objectives outlined. Finally, we will conclude and present the development stages of the LLTB project.

Keywords: Living Lab, Toolbox, ENoLL cube, knowledge base, OWL.

LLTB members

- Raymond Morel
- Philip Koenig

Introduction

Outlook

The outlook of the project LLTB is, on the one hand, to provide a facilitator environment to LL designers and users. And, on the other hand, to allow the reusability of experiments, data and solutions developments in order to avoid reinventing the wheel at each LL.

Goal

The objective of the LLTB is to provide an ecosystem of IT components for the implementation of an LL, specific tools for the operational phases and provide an integrated environment for data management, communications, results and reporting.
This ecosystem is open and scalable at the level of each LL. But also, at the level of the entire the LLTB. This evolution is managed by the stakeholder community itself.

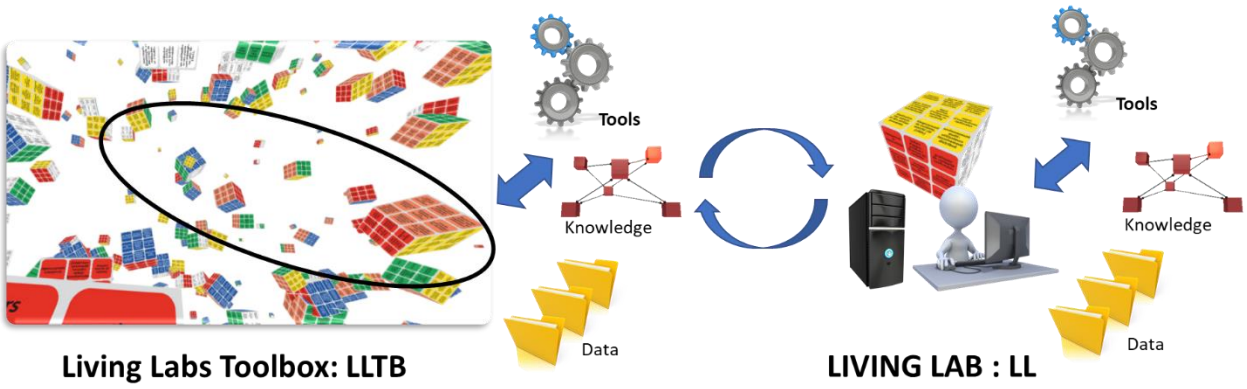

figure 1: Ecosystem LLTB and LL.

Global concept

The generic nature of LL allows them to invest almost all sectors, and each LL is a special case that requires tools and specific operational methods depending mainly on the stakeholders, the nature of the LL field, the goals, the resources available and the context.

It is probably for these reasons that few ICT applications have been developed for LL. In practice, each LL develops its own data management structure, operational processes and outcomes. This is consistent, because the heterogeneity of domains makes it difficult to create a generic application that works for all LL. But it is also what prevents reusability of results and developed software modules, as they are rarely compatible because they have been developed independently and not in concert.

Standard ICT solutions are efficient when it is possible to specify: their architecture, their data model, their business processes, the specifications of all their features, all their inputs and outputs as well as their HMI interfaces.

However, it is rarely possible to specify the above points in advance in the context of an LL. For this reason, the LLTB is not conceived as an "application", but as an "ecosystem of tools" (toolbox) that can be combined, and they can communicate with each other. With these tools, it is possible to build a specific LL in an ad hoc manner corresponding to the exact needs of the LL under consideration and adapted to each of the stakeholders

The basic principle of LLTB project is not to fix the features, nor the data structures a priori, but to develop a platform allowing to build and integrate the functionalities as required by specifications of each particular LL, as well as dynamically create the data structures according to the specific needs of each LL.

The LLTB is therefore an ecosystem of tools that can be organized to meet the specific objectives of each LL and can be dynamically modified to adapt to the evolution of the LL during its work.

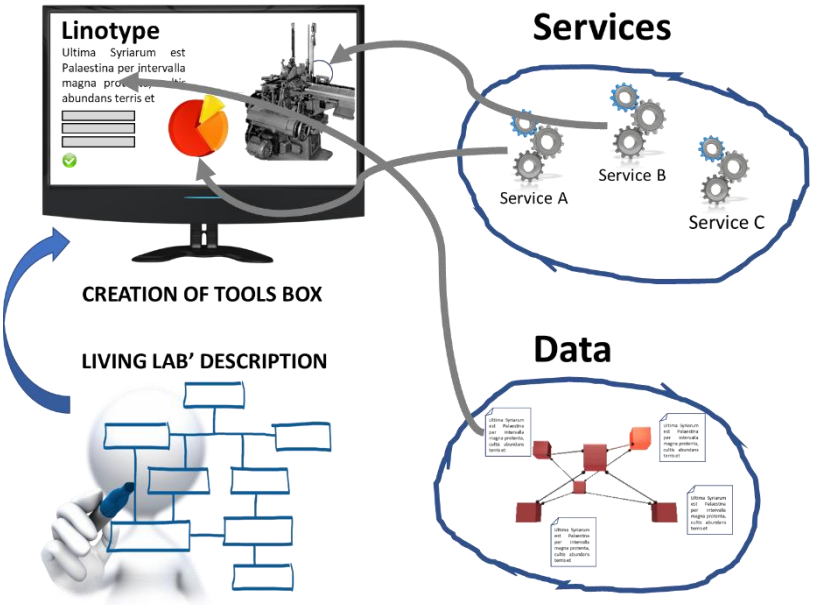

figure 2 : Principle describe to build HMI Interfaces.

LLTB General Concepts

The basic principle is the construction of a description reference frame structuring and linking the tools used in an LL. It is precisely this description-based approach that gives the LLTB its dynamic aspects and general characteristics. Each LL may describe its domain with its specific objects, relationships linking them together and give them meaning by grouping and classifying in sets (concepts, classes). The tools are integrated in this structure by specific properties.

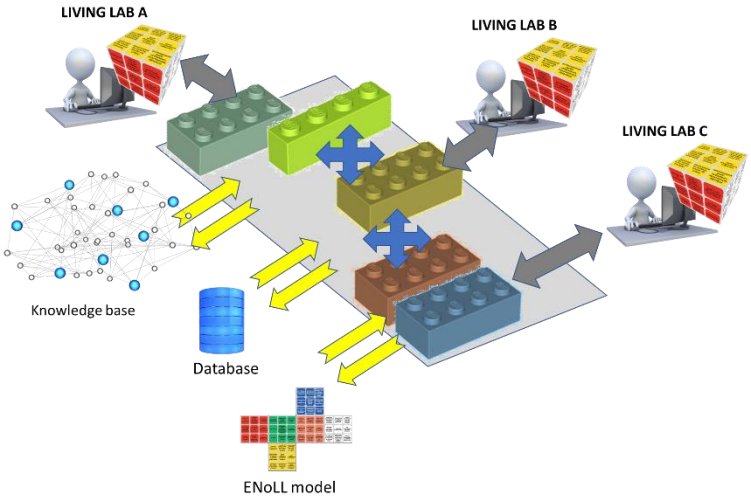

figure 3: Illustrative view of the LLTB.

It should be noted that in this article the approach is presented in a linear way, but that in the reality of the LL field, it is often iterative. The LLTB allows naturally manage iterative structures by using the versioning technique. Each version is assigned a unique identifier, a version number, data and annotations. It is then possible to browse versions to understand evolution of the LL objects, states and data.

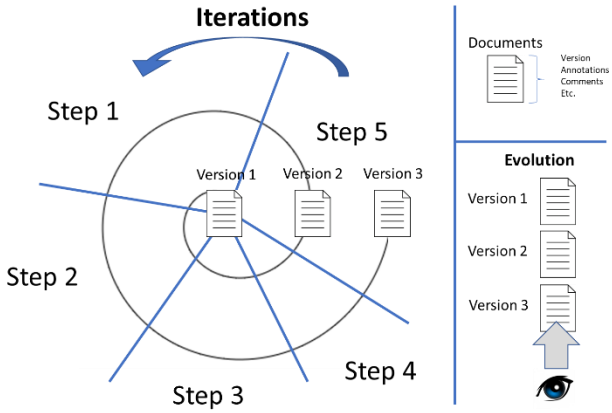

figure 4 : Versioning.

Design of the Living Lab

The LL design process with the LLTB is based on the ENoLL cube (European Network of Living Labs). This reference frame is the one chosen for structuring a LL with LLTB. It is technically possible to choose other reference frames, but the ENoLL cube is well known and allows to make all the steps without forgetting any dimensions. Furthermore, this common framework will exchange results or tools between all LL who have chosen this same framework.

This article is not intended to describe a method or the ENoLL cube, but the operating principle tools using these standards, that is why we will discuss neither the cube, nor its dimensions, but its use in the LLTB.

Reference frame

The reference frame is in the form of a 3D cube that can be manipulated with the mouse or keyboard and each element is selectable.
This choice of the use of 3D is motivated by the fact that the interface is intuitive. It allows the user to browse among information, links and tools in a simple way, without necessarily needing training neither of the LLTB, nor details of the operation of an LL. Most LLTB tools are visual, so the user can focus on the LL and not on the technical aspects of a computer application.

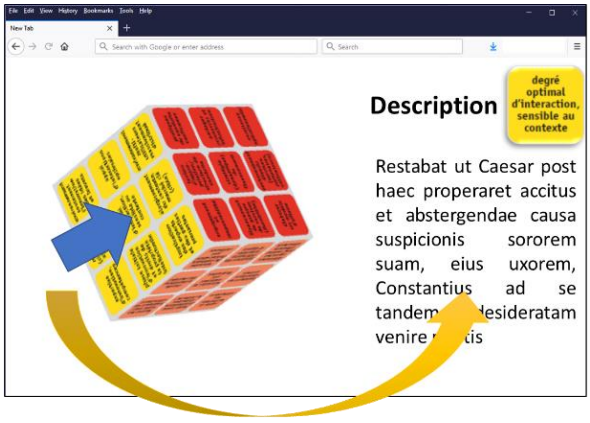

figure 5 : Reference frame ENoLL cube.

What is important to perceive the LLTB is that the user interfaces (HMI: human-machine interface) are not defined and fixed but are the expression of descriptions contained in the knowledge base.
It is therefore possible to modify texts, translate them, specify them. These changes are not simple user interface modifications, but a direct intervention in the semantic definitions of LL.

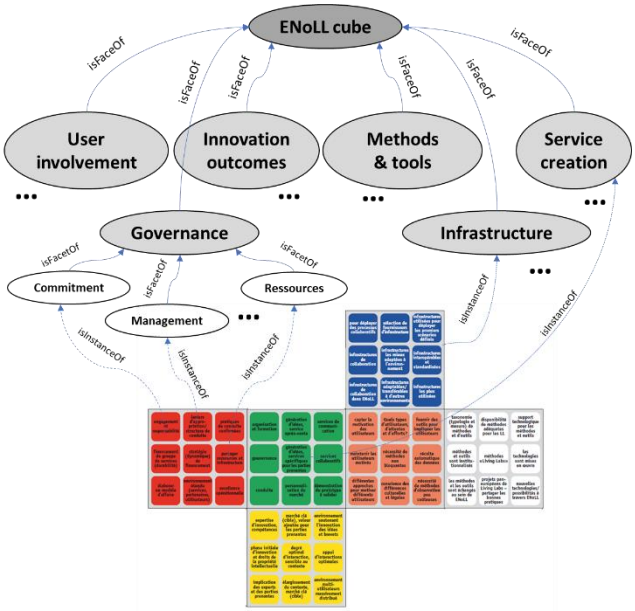

figure 6 : Relations between ENoLL model and knowledge base.

It is precisely for this reason that it is possible to propose a set of common tools to LL in various domains with different goals. Each LL defines its specificities that will be expressed in the interface presented to users.

Definition of vocabulary and concepts

Each domain defines a set of concepts, terminology and specific structures. The domain description is therefore an important point that must achieved by co-creation with all participants. Indeed, it is on this basis that will be built LL. Different levels of construction are possible, starting from scratch or using existing structures (for example developed by other LL in similar domains).

The vocabulary consists of keywords and specific terms, but it also includes synonyms and other semantic relationships. Each word can also be translated, it will then be linked to its language, but also to the concepts it describes and to the other words translated into the different languages considered. A connector on lexical databases helps build vocabulary LL.
The user interface can be connected to other sources, such as WordNet.

The domain is described by: individuals i.e. domain objects, properties i.e. relationships between individuals, classes (concepts) i.e. descriptions of sets of individuals belonging to the same definition and finally annotations i.e. metadata.
The management interface can be customized according to needs (simple or full).

This approach allows to reflect and harmonize the basic building blocks of LL and also share a common understanding.

LL Design

Initiating the process of LL is to describe the current state of the situation and to express the purpose and the goals.

To perform this step, it is possible to use the ENoLL cube reference frame. The principle is to choose the relevant dimensions and determine the data and tools to be used in the context of the LL under consideration.

Stakeholder Mapping

This involves identifying all stakeholders, end users, public bodies, companies, specialists and ICT stakeholders.

A directory and a list of stakeholders can be established, but in addition to the data itself, it is possible to include metadata and semantic relationships on concepts defined above. Thus, stakeholders will be placed in the LL space with semantic links between stakeholders, tasks, objects, and any other element defined in the LL.

This allows to query the LLTB not only about the data itself, but also about links, concepts, expressions, or metadata.

Measure

To measure progress in projects, it is necessary to determine performance indicators particularly in the dimensions of knowledge, social, and business, or on any element deemed appropriate.

The construction of the LL in the LLTB makes it possible to attach to each concept software agents which can monitor data, states or events. This software agent can generate data for HMI interface, report, log or even to trigger actions.

BMG guidelines for Livings Labs

The Business Model Generation (BMG) template is a recommended tool in the build phase of an LL. This is not the only model possible, but it is well suited to the LL.

This model is a tool which may be included in the LLTB and allows not only to build the BMG canvas, but also to semantically link each element with defined concepts.

Synthesis

LLTB is based on the description of the concepts and the relationships between domain objects with a defined vocabulary. This description is the heart of the LLTB and makes it possible to link all the items into a coherent structure. Each item is unique and not repeated. With each evolution on a LL element, it is the whole of the LL which is informed and impacted.

The base model structuring the space of LL is the ENoLL cube, which makes it possible not to forget any dimension, to share a common understanding between the LL, because it is well described and published, and it allows the sharing of results as well as tools.

To each concept it is possible to attach a software agent that can monitor and transmit data or trigger an action based on rules.

Architecture

LLTB technology is based on a Service Oriented Architecture (SOA) for actions, Web Ontology Language (OWL) for knowledge and service representation, Agent Oriented Programming (AOP) for complex processing and finally web page programming for user interfaces.

As mentioned, each LL have specific operation and architecture. But the coherence, interoperability, communications and reusability of the LLTB are supported by a common distributed architecture in which each LL can connect if it wishes and develop its specificities.

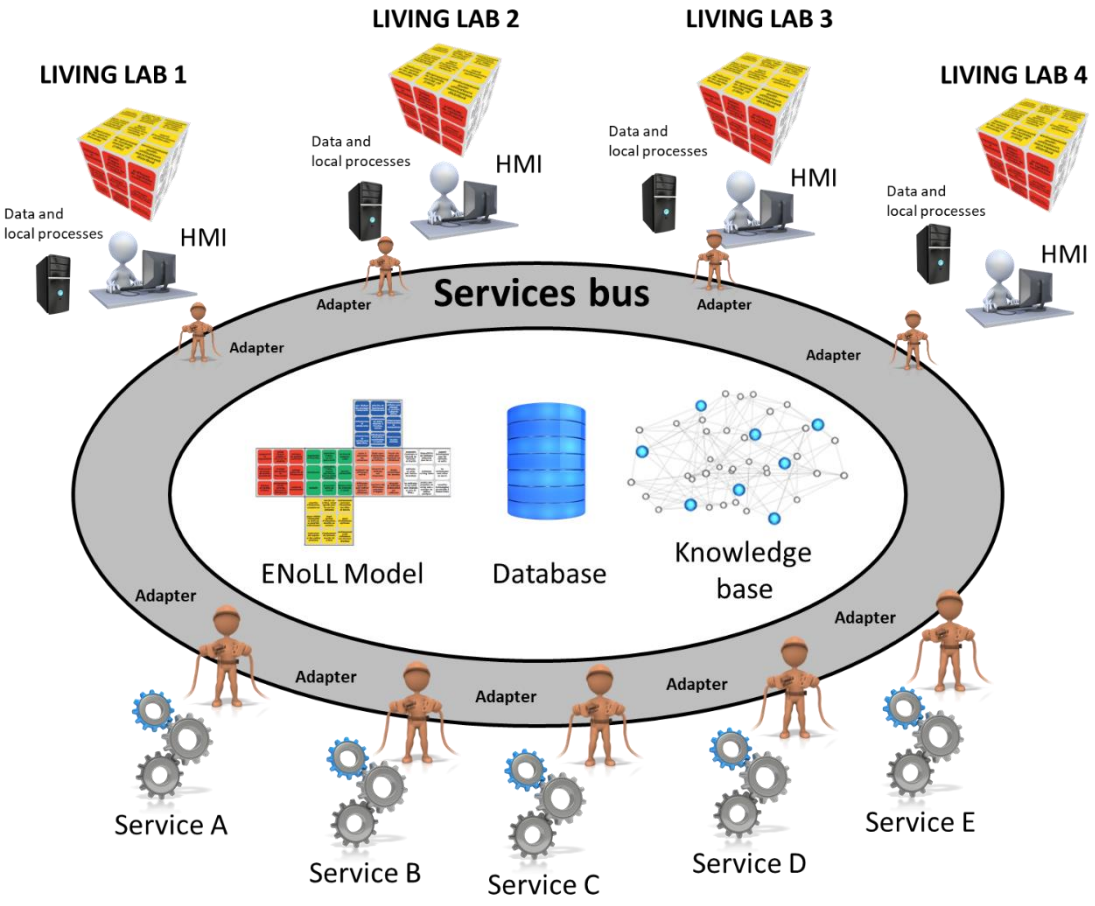

figure 7 : Global LLTB architecture.

Services

This architecture is based on a bus of services that has a role of mediator between users (or consumers) who wish to perform operations that they cannot do locally, and services made available (producers). For example, an LL wishes to statistical calculations based on geographic data. Suppose this LL does not have the applications to do this operation locally. He starts a search in the LLTB and gets a response from a specialized service in statistical calculations. The user can then integrate the service in its LL and perform its operations through this service. At the LL level, this will give the impression that this service is available locally.

This architecture is interesting, because at any time it is possible to add new services and register them in the LLTB. This allows the Living Labs community to create new services throughout their experiences and enrich LLTB.

Livings Labs

Each LL has a stand-alone environment and can connect to the "service bus" and use the services. The standalone environment can manage LL, its data, and applications independently.

The LLTB allow LL which wish to enrich their tools via the services, but also reuse data, practices or outcomes shared by another LL.

Data and resources are not just stored data, but they are enriched with metadata and relationships that give them meaning and allow LL software agents to search and process concepts and relationships. There are stored in the knowledge base in the form of triples.

LLTB's human-machine interfaces (HMI) are dynamic and based on HTML pages. Indeed, according to the use and the context, the HMI will federate several types of data, several results from different sources and will organize its components according to the specific needs of the user. It is therefore not possible to predict all cases (especially as new services can be created at any time).

In order to meet these needs, LLTB interfaces are containers in which are inscribed the elements and data. This registration is done by the user himself who can describe the HMI with a set of concepts and their relationships. Thus, each LL can adapt interfaces to their needs and their constraints.

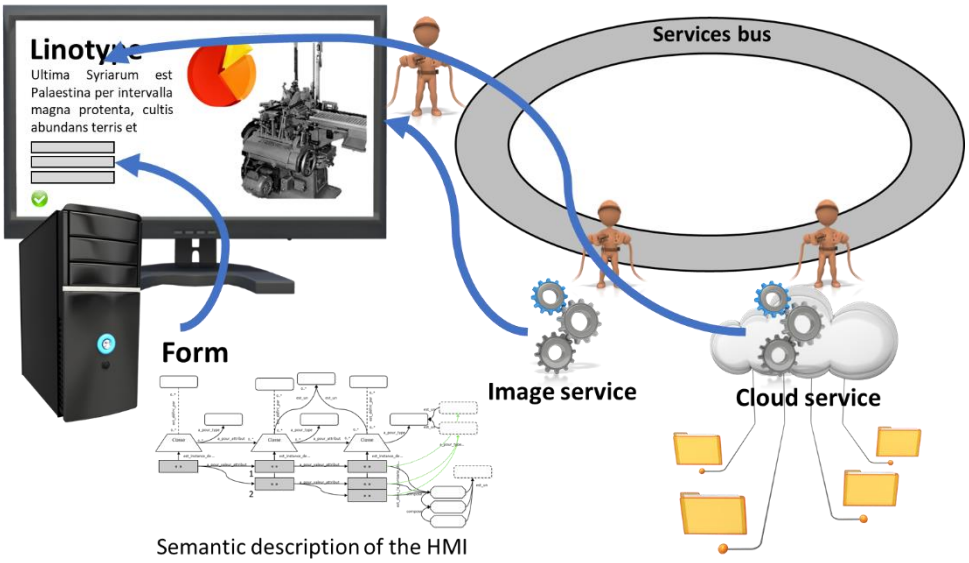

figure 8 : Federation sources and example of a dynamic HMI.

The core

The core of the LLTB mainly consists of a knowledge base, a database, a service bus, HMI interfaces generator, software agents and adapters.

The knowledge base stores and organizes metadata and allows to exploit them with programs based on rules and inference systems among others, to understand the relationships between domain objects, to implement ENoLL structure, to discover new knowledge, to do research through concepts, to control and to drive processes.

The database mainly stores resources such as files, multimedia documents (image, audio, video), text, configurations, activity logs. The roles of the database are to ensure the availability of resources, usage policy and data sustainability.

The services bus is the backbone of LLTB because it connects all the components together. It allows distributed and independent integration. This means that it is possible to distribute the components on a computer network and thus allow a common work independently of the place. It ensures the routing, distribution and eventually transformation of the data. But it is also responsible for implementing SOA governance and quality of service (security, confidentiality, scalability, reliability).

The main role of the HMI interface generators is to allow all LL and all stakeholders to customize interactions with LLTBs. As already mentioned above, each LL and each phase of the projects requires a specific organisation of its HMI interfaces. The role of interface builders is to make it possible to organize the elements to display by semantic description, to describe actions to perform in a simple and intuitive way.
In LLTB all HMI interfaces are created in Web browsers, allowing the LLTB to be used regardless of the operating system or the equipment (from smartphone to desktop).

The software agents are autonomous programs that operate independently to accomplish a set of tasks leading to the goal they have been given. They generally have the ability to access their environment, process the data in order to converge towards the assigned goal, communicate with other agents and make decisions according to established rules. The roles of software agents in LLTB are to be able to install triggers waiting for an event or to monitor a context. They can also be used as a data or resource mining agent, making relevant results or information difficult to obtain manually. They can also perform mass or distributed operations.

The adapters are small software programs, often without HMI interface, that adapt other software, or a means of communication, to the service bus. The adapters are used to install services, access other systems, integrate third-party applications.

LLTB structure

To ensure consistency throughout the LLTB, it is necessary to accurately model the architecture and define the roles and responsibilities of each component. The layered model below shows the LLTB main structure.

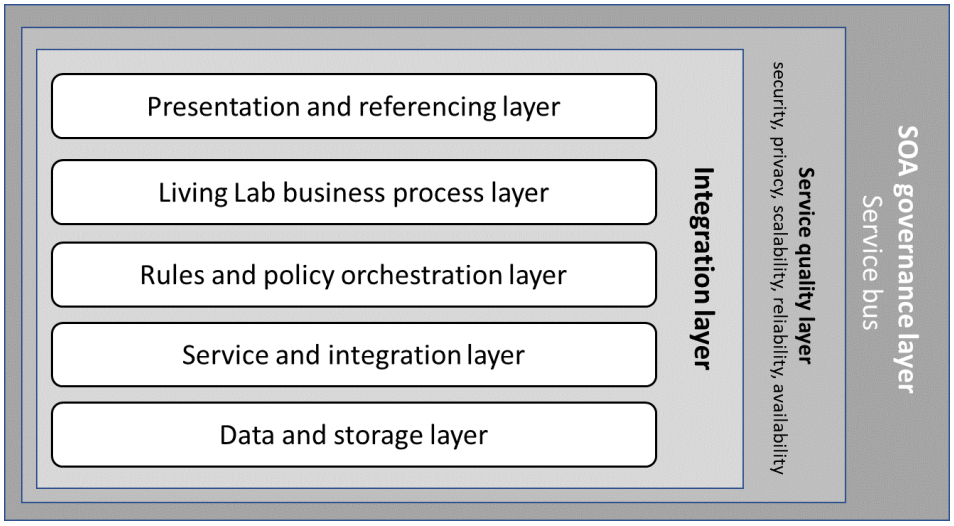

figure 9 : Main layers of LLTB (architecture model-oriented services or SOA).

The **presentation and referencing layer** is the most visible, because its role is to manage all user interaction components, such as web HMI generators. This layer also supports the referencing of data, of resources, and of users.

The **Living Lab business process layer** is the most rich and complex, because it has for role the management of the processes within the LL, the management of the data and the specific knowledge, the management of the operations and calculations related to each LL.

The **rules and policy orchestration layer** is the most sensitive, because it deals with all the management of rights, access, groups, roles, compliance with specific rules for each LL, but also for all workflows inside LL.

The **service and integration layer** is the most diverse, because it has the role of service management, data routing, adapter and operation integration, referencing all resources, responding to service access requests and searching for features.

The **data and storage layer** the most technical, as it is responsible for storing the data, restoring them, and responding to various requests.

The **integration layer** allows the management of communications between the layers seen above. Exchange protocols ensure that exchanges are controlled and that each layer communicates only with its neighbouring layers.

The **service quality layer** is responsible for ensuring the security, privacy, scalability, reliability, availability of the entire operation of the LLTB.

The **SOA governance layer** mainly manages the service bus and ensures its overall operations.

HMI interfaces regardless of the technologies

In a connected and mobile world, it is essential that the LLTB can be used as well from a fixed station as from a mobile equipment, even from a smartphone. The technology envisaged must therefore be, as far as possible, independent of equipment and operating systems.
Currently, the common denominator between all connected equipment is the web browser. There are also possibilities to use offline browsers for situations where the network is not available or not desirable.

This is the reason for choosing to develop a Web application hosted in browsers using (among others) Html5, JavaScript, Canvas, WebGL, Ajax, web-workers, etc.

Conclusion

While not all Livings Labs necessarily need to use IT tools for their operation, facilitators tools can be of significant assistance in complex configurations involving many stakeholders and actors do not necessarily residing in the location of the LL. It also facilitates data research and reporting. On the other hand, the constructivist aspect of LLTB structure allows a constant enrichment with experience and modules developed in the context of Living Labs.

The LLTB, is a facilitating environment that allows all users to customize it according to their specific needs.

Steps planned for the development of LLTB are:

- Development of the basic architectural structure;
- Ontology core and LLTB knowledge base construction;
- Development of the initial LLTB platform;
- Development of the core system;
- Development of semantic interfaces;
- Cube ENoLL representation on an interactive graphical interface;
- Prototyping with Living Labs simulations;
- Test in a real Living Lab.

References

Living labs

- Patrick Genoud et Andréas Schweizer et al., « Living Lab e-Inclusion, Exclusion et e-Exclusion: analyses, méthodes et outils pour maîtriser le changement et promouvoir l'innovation sociale et technologique », SATW, 2009.
  Accès http://www.ot-lab.ch/wp-content/uploads/2012/04/OT_SATW_Rapport-Living-Lab-eInclusion-Vfinal.pdf

- Patrick Dubé, Joëlle Sarrailh, Christophe Billebaud, Claire Grillet, Virginie Zingraff, Isabelle Kostecki, « Le livre blanc des LivingLabs », umvelt, 2014.
  Accès http://www.montreal-invivo.com/wp-content/uploads/2014/12/livre-blanc-LL-Umvelt-Final-mai-2014.pdf

- Mélanie Dallaire et al., « Le Living lab SAT », CHU Sainte-Justine, 2012.
  Accès https://www.chusj.org/fr/Calendrier-salle-presse/nouvelles/actualites/2012/Le-Living-lab-SAT-CHU-Sainte-Justine-est-officiell