Towards a MultiLingual Data & Services Infrastructure

Support Action

Grant agreement no.: 610951

D2.3 – Final design and specification

MLi – WP2-D2.3

<table>
<thead>
<tr>
<th>Project Number</th>
<th>FP7-ICT-610951</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>30/04/2016</td>
</tr>
<tr>
<td>Actual Date</td>
<td>30/04/2016</td>
</tr>
<tr>
<td>Document Author/s</td>
<td>Tomas Pariente Lobo, Mateusz Radzimski,</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
<tr>
<td>Document approved by</td>
<td>ATOS</td>
</tr>
</tbody>
</table>

This project has been funded with support from the European Commission’s ICT programme of FP7
### Document Version Control

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change Made (and if appropriate reason for change)</th>
<th>Initials of Commentator(s) or Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01/03/2016</td>
<td>Table of content</td>
<td>TPL / MR</td>
</tr>
<tr>
<td>0.2</td>
<td>30/03/2016</td>
<td>First draft</td>
<td>MR</td>
</tr>
<tr>
<td>0.3</td>
<td>22/04/2016</td>
<td>Second draft with most of the content included</td>
<td>MR / TPL</td>
</tr>
<tr>
<td>0.4</td>
<td>26/04/2016</td>
<td>Final draft for review</td>
<td>TPL</td>
</tr>
</tbody>
</table>

### Document Change Commentator or Author

<table>
<thead>
<tr>
<th>Author Initials</th>
<th>Name of Author</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPL</td>
<td>Tomas Pariente</td>
<td>ATOS</td>
</tr>
<tr>
<td>MR</td>
<td>Mateusz Radzimski</td>
<td>ATOS</td>
</tr>
</tbody>
</table>

### Document Quality Control

<table>
<thead>
<tr>
<th>Version QA</th>
<th>Date</th>
<th>Comments (and if appropriate reason for change)</th>
<th>Initials of QA Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>26/04/2016</td>
<td></td>
<td>SAB</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTIVE SUMMARY</td>
<td>5</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>6</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>6</td>
</tr>
<tr>
<td>1.2 Structure of the deliverable</td>
<td>6</td>
</tr>
<tr>
<td>2. METHODOLOGY TO DERIVE THE MLI HUB</td>
<td>7</td>
</tr>
<tr>
<td>2.1 High-level methodological approach</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Methodology to derive the MLI Hub architecture</td>
<td>9</td>
</tr>
<tr>
<td>2.2.1 Architectural Considerations</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2 Building blocks of the MLI Hub methodology</td>
<td>10</td>
</tr>
<tr>
<td>3. MLI HUB ARCHITECTURE REVISITED</td>
<td>13</td>
</tr>
<tr>
<td>4. ARCHITECTURE INSTANTIATIONS</td>
<td>17</td>
</tr>
<tr>
<td>4.1 LTi Cloud</td>
<td>17</td>
</tr>
<tr>
<td>4.2 Atos CaptureAN</td>
<td>18</td>
</tr>
<tr>
<td>4.3 FP7 Project “Pheme”</td>
<td>20</td>
</tr>
<tr>
<td>5. MLI HUB GOVERNANCE AND SUSTAINABILITY</td>
<td>22</td>
</tr>
<tr>
<td>6. CONCLUSIONS</td>
<td>24</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>25</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1. Adapting UP to MLi Hub definition process. .......................................................... 7
Figure 2. Enterprise Architecture reference models ................................................................. 8
Figure 3. Fundamental MLi architecture aspects .................................................................... 9
Figure 4. Enterprise Architecture reference meta-model. ......................................................... 11
Figure 5. Reference Architecture meta-model particularized to the MLi project .................... 12
Figure 6: MLi Hub Reference Architecture revisited ......................................................... 14
Figure 7: LTi Cloud architecture ......................................................................................... 17
Figure 8: LTi Cloud and MLi hub architecture .................................................................... 18
Figure 9: CaptureAN: a sample implementation of the MLi Hub Reference Architecture ...... 19
Figure 10: Pheme: a sample implementation of the MLi Hub Reference Architecture .......... 20

LIST OF ABBREVIATIONS

ADR | Alternative Dispute Resolution  | ADM | Architecture Development Method
--- | --- | --- | ---
AI | Artificial Intelligence | API | Application Programming Interface
ASR | Automatic Speech Recognition | BI | Business Intelligence
CAT | Computer-Aided Translations | CMS | Content Management System
CPU | Central Processing Unit | DB | Database
DMS | Document Management System | DSL | Domain-specific language
EA | Enterprise Architecture | ECM | Enterprise Content Management
FEA | Federal Enterprise Architecture | GPL | General Public License
HTTP | Hypertext Transfer Protocol | HPC | High Performance Computing
IaaS | Infrastructure as a Service | JSON | JavaScript Object Notation
LR(s) | Language Resource(s) | MoU | Memoranda of Understanding
LVC | Language Value Chain | Mx | Month x
MT | Machine Translation | NL | Natural Language
OER | Online Dispute Resolution | NLP | Natural Language Processing
POS | Part-of-Speech | QA | Question Answering
QoS | Quality of Service | REST | Representational state Transfer
RR | Resource Repository | RUP | Rational Unified Process
R&D | Research and Development | SLA | Service Level Agreement
SMT | Statistical Machine Translation | SN | Social Network
SOAP | Simple Object Access Protocol | SVN | Apache Subversion
TOGAF | The Open Group Architecture Framework | UGC | User-Generated Content
UP | Unified Process
EXECUTIVE SUMMARY

The MLi project aims at delivering the strategic vision and operational specifications for building the MLi (European MultiLingual data and service Infrastructure). One of the major outcomes of the project is the technical and service platform called the MLi Hub.

For the reader’s convenience this deliverable repeats a few paragraphs from previous deliverables concerning the methodology to study the MLi Hub in order to properly position the work in the scope of the overall process. However, most of the content on how the MLi Hub has been defined can be found in previous deliverables.

This document provides a final view and refined view of the MLi Hub, by framing the design into the reference use case, namely the eCommerce scenario. One of the main goals of MLi with respect to eCommerce is to help organizations to sell goods cross-border using LT tools to overcome language and cultural barriers. In practical terms, this means the potential increase of selling from a reduced market to all EU countries and beyond. Even organizations that currently have cross-border and cross-language eCommerce platforms often do so by creating local branches that localize everything for the specific countries they operate in. This is a sub-optimal solution and LT technologies may have a substantial role improving it. Technologies such as improved multilingual search and SEO, MT, text analytics, speech recognition, etc., are key to favour this approach.

From the eCommerce perspective, the MLi Hub aims at covering potential infrastructure needs for developing and packaging services and toolsets that help in this quest. It is not therefore a specific solution, but rather an umbrella that covers a set of LT activities, tools and services. The document presents a list covering some core eCommerce multilingual scenarios as viewed from the business point of view. Those activities are the main drivers of the technical architecture, outlining the desired processes to be supported by technical components, actors or services.

The document proposes a final Reference Architecture for the MLi Hub revisiting previous documents from work package 2. It also positions the reference architecture in the scope of actual implementations based on it, such as the LTi Cloud prototype delivered also in MLi in the scope of work package 4.

For the impatient reader, who is already familiar with the LT/MT environment and the business context of the MLi Hub, the main takeaways of this deliverable are the sections where the proposed reference architecture and the examples of its instantiation are explained in detail.
1. INTRODUCTION

1.1 Overview

The MLi project aims at delivering the strategic vision and operational specifications for building the MLi (European MultiLingual data and service Infrastructure).

WP2’s objective is to design technical infrastructure and service framework under the common MLi HUB platform. In this deliverable we continue the work outlined in the initial (MLi D2.1 Early Design of the Reference Architecture and the Hub, 2014) and the intermediate (MLi D2.2 Intermediate design and specification, 2015) specifications. The objective of the present document is therefore twofold:

- Provide the final iterative cycle of analysis and design of the reference architecture following the methodological approach outlined in the first WP2 deliverable.
- Frame the reference architecture into the concrete use case(s), and more in particular to the LTi Cloud implemented in WP4 as one of the possible prototype implementations of the MLi Hub.

In other words, this deliverable can be seen as a refinement of the MLi Hub architecture building on top of the previous WP2 deliverables and the feedback from other MLi work packages, most notably WP4 and the implementation of the LTi Cloud prototype. If we compare the process of designing the reference architecture to the work of a sculptor, then the previous deliverables represented the preparation of the base material for the future sculpture. This phase, when the reference architecture is tested to develop a concrete prototype (LTi Cloud), is by the same analogy like chiselling marble, shaping or removing unnecessary elements, giving it more recognizable form and desired features.

1.2 Structure of the deliverable

The structure of this deliverable reflects the process of architecture refinement. We start by grounding the process in the methodology, revising previously devised steps and identifying main goals and tasks for this iteration phase. We will recall both pillars that define our work, that are: (i) the UP-based MLi methodology for the overall process management and (ii) the Enterprise Architecture that provides the reference framework for the architecture construction. All this will be reflected in the Section 2.

In the next section (Section 3) we provide an overview of the final MLi Hub architecture and service framework, based on the work done in previous iteration and mapping each of the building blocks to potential instantiations. This is the main goal of this deliverable and most of the previous work will find its results there.

Section 4 provides examples of instantiations of the reference architecture for three different scenarios, paying special attention to the LTi Cloud prototype and its relation with the MLi Hub.

Section 6 summarises aspects related to the costed roadmap and sustainability of an implementation based on the MLi Hub architecture.

Section 6 provides the conclusions of the document.

For the convenience of the reader, we will recall some content of previous WP2 deliverables, especially regarding the methodology and previous iterations of the reference architecture. As the meta-model designed in the previous iteration is the starting point of the architectural refinement, therefore it is important to bring it to this phase.
2. METHODOLOGY TO DERIVE THE MLI HUB

2.1 High-level methodological approach

This section aims to summarize the methodological approach followed in WP2 for the definition of the Hub, which was originally presented within (MLi D2.1 Early Design of the Reference Architecture and the Hub, 2014). The methodological approach adopted in the process of designing the reference architecture of the MLi Hub consists of the following two main components:

- **Unified Process** as a reference methodology for flexible and customizable process of software development (IBM, 2009).
- **EA** to provide a reference model for architecture definition.

The Unified Process provides a methodological underpinning of the business-driven, iterative approach to the software engineering. The core of the process is an iterative methodology allowing for feedback loops for presenting and refining intermediate results. This allows for building common understanding of the project goals, ensures the proper level of collaboration and communication and facilitates identification of risks early in the process. Moreover, the lightweight cycles allow for early validation of the project artefacts by shortening the cycles and consulting intermediate results early and often.

The most important aspect of UP is the adaptability to the specific scenarios. Due to the fact that MLi is not building an actual implementation, but rather aims at delivering MLi Hub design, costed roadmaps and validation plans, the overall methodology should only focus on the relevant actions. The Adaptation of the UP to the MLi Hub definition process has been presented in Figure 1. In this case, the whole process has been trimmed down to the Inception and Elaboration phases. Although the methodology is tailored to architecture definition, it also comprises tasks which are not directly part of D2.1, such as the inception phase tasks, but largely influence its development.

![Figure 1. Adapting UP to MLi Hub definition process.](image)

The second methodological pillar is the EA reference model for architectural definition of the MLi Hub. The EA model puts the system in the broader context, which not only accounts for technical architecture, but also for such aspects as: management, service and infrastructure layers and business operation reference. The model is loosely based
on Federal EA\(^1\) reference model and TOGAF reference\(^2\) (The Open Group, 2011), which encompass all relevant architecture layers and provide guidelines for developing and managing the architecture. In this approach we present the architecture in a set of different but complementary and coherent views, depicting various aspects of the architecture and its context (IEEE Std 1471-2000, Recommended Practice for Architectural Description, 2000).

A complete architecture includes five reference models (The Open Group, 2011), as presented in the Figure 2.

![Figure 2. Enterprise Architecture reference models.](image)

The following list presents the alignment between EA reference model and the MLi project and MLi Hub creation.

- **Performance Reference Model**: This layer is in charge of setting the principles on which the architecture will be based, driving the strategic decisions, setting the project vision, identifying stakeholders and key enablers, and finally planning processes and activities. In MLi this has been partially accomplished during the inception phase of the project as a shared effort among all partners. The performance model in MLi is about ensuring a common vision among partners and stakeholders, interview of the main stakeholders to understand their requirements and needs: how MLi could help them and how they can contribute to MLi’s goals. Therefore, this layer is in practice a join-effort between WP2 and WP7.

- **Business Reference Model**: This layer describes the main business model and business operations of the project, taking into account aspects such as service delivery and support and resources management. For MLi the specific actions are to understand the stakeholders’ business approaches in order to define the possible business context in which the MLi architecture and service framework will be used. That means the specification of the main business cases or usage scenarios of MLi in order to ensure the definition of a sound reference architecture covering the current and future needs of the main stakeholders. This layer is therefore a task to be carried out in collaboration with WP4 (Language Enabler), WP6 (Innovation Services) and WP7 (Business models).

- **Service Component Reference Model**: The goal of this layer is to define a framework for functional classification of service components and understand how those components are supporting the business objectives. It follows therefore a component-based approach where components encapsulate end-user services, business logic and backend services. This is a more technical layer that for MLi should conceptualize the vision of the Service Framework. It will define the classes of services to be provided by the MLi infrastructure, identifying necessary back-office for service provisioning and defining MLi components (high level MLi architecture draft). This layer requires collaboration from most of the work packages, especially

---

\(^1\) Federal Enterprise Architecture (FEA): http://www.whitehouse.gov/omb/e-gov/fea

\(^2\) TOGAF is a framework (a detailed method and a set of supporting tools) for developing an EA. This specification is published by The Open Group consortium.
from WP4 (Language Enabler), WP5 (R&D Services) and WP6 (Innovation Services).

- **Data Reference Model**: The data layer is about describing and categorizing data resources and assets, their usage, the rules of data exchange between organizations and/or components, and the acquisition, management and distribution of data resources. It should cover also the resources’ access policies and rules. This layer is related mainly to the work done in WP3 (Resources).

- **Technical Reference Model**: This layer sets the actual technical definition of the reference architecture. It is based on existing standards, technologies and best practices that support the provision and delivery of the required service components. It depicts the reference service platforms, infrastructure, frameworks and the existing software components. For MLi it means to identify industry standards relevant to the LT and MT domains, to analyse other initiatives in the field (other LT projects, available technical resources, existing technical architectures and technologies, etc.), to analyse hosting platforms and deployment strategies, to design necessary infrastructure for handling data and service provisioning, etc. This layer is therefore related to most of the work packages, with especial collaboration with WP3 (Resources), WP4 (Language Enabler) and WP5 (R&D Services).

In the later sections of this deliverable we will present an overview of the main building blocks for the models directly relevant to the design of MLi Hub, namely: Service Component Reference Model, Data Reference Model and Technical Reference Model.

### 2.2 Methodology to derive the MLi Hub architecture

#### 2.2.1 Architectural Considerations

This section provides a conceptual overview of the MLi Hub architecture. The basis of this section is the MLi project vision and the fact-finding process leading to the definition of the most significant requirements. Figure 3 presents a “distilled” overview of the main MLi Hub characteristics in the form of five fundamental aspects that should be supported by the architecture.

![Figure 3. Fundamental MLi architecture aspects.](image)

Those aspects can be viewed as high-level, architecturally significant requirements towards the design of the MLi Hub, as explained below:

3 Please note that this is a high level architectural reference model, do not confuse with “data model” term used to describe low level representation of data.
• Rich language features through the reuse of multilingual resources – MLi Hub should allow for reusing as much as possible existing multilingual resources, such as those identified within the META-SHARE inventories. This implies taking into account various details such as resource metadata description, interoperability formats, resource storage, search, discovery and licensing. A resource governance management plan should be consolidated in order to sustain the existence of such catalogues and maintain the underlying quality. This aspect can be dimensioned in the context of deployment scenario. This can have a different impact in a small scale, where the system uses only a fixed, predefined set of resources and much of the aforementioned aspects could be minimized. On the other hand, large-scale systems will address all of them.

• APIs and Services – NLP and MT comprise a plethora of smaller, specialized tasks, each one providing a different functionality. Those services are typically chained together in order to obtain a desired goal or provide a higher level service. Apart from language-based services, also a framework for 3rd party services, support services and even human-based tasks should be taken into account. Service and APIs provisioning should be realized in a unified and consistent way, allowing for flexibility and including various quality metrics and SLAs.

• Scaling-Up & Cloud based deployments – a very important aspect of the MLi architecture is the multi-scenario deployment capability. It must take into account requirements toward the stakeholders’ business scenario, but also processing efficiency, data and resources capacity, and sustainability of computing infrastructure. Flexible architecture should take into account the ability of scaling-up and scaling out, depending on the scenario. Also the cloud based deployment should be considered where the computing resources can be acquired in the pay-as-you-go manner, giving the flexibility of dimensioning the overall solution to fit the increasing need for translation capacities. The use of cloud and big data and recommendations for the Hub has been provided within WP5 in (MLi D5.2 Programmatic Recommendations for Big Data and Social Language Data Infrastructure, 2015).

• Workflow approach – MT systems typically perform training or translation as a complex process spanning various, often repeatable tasks. In a fixed scenario, such process is created up-front and doesn’t change. Complex workflows should be manageable so the system can reproduce the step-by-step process. However, flexibility of translation process allows adapting them to different use cases or even languages, without the need of redesigning the whole solution. Workflow approach allows for creating flexible translation processes and enables to configure them and adapt to the changing requirements. This provides the foundation for a generic system that can already contain all necessary language services, but only needs to be properly configured to support specific use cases.

• LT Standards – the building blocks of MLi Hub should comply with industry established technology and service standards or best practices. This is crucial for ensuring architecture and component reusability, interoperability and adaptability. Relying on industry-agreed standards minimises the risk of technological obsolescence and enables the participation of technology suppliers, third party component developers and service providers.

2.2.2 Building blocks of the MLi Hub methodology

In order to put all relevant aspects into the common context of MLi Hub architecture, we follow the EA, taking the TOGAF approach as a reference architectural meta-model. Based on that, we focus on the relevant by properly tailoring the model to the MLi specifics and particular WP2 objectives. The reference meta-model provides a guide for
organizing different aspects of the whole MLi Hub into different logical blocks. Those “high-level building blocks” will be further specified in detail along with their relations with other blocks in the scope of their corresponding architecture layers.

It is important to mention that the meta-model covers the whole process of architectural definition and it goes far beyond the scope of WP2 and this deliverable (see Figure 4). Therefore, we present the overall picture, but later only focus only on the relevant aspects for constructing technical infrastructure and service framework for MLi Hub.

The reference meta-model is divided into five main parts (also called TOGAF ADM phases) (The Open Group, 2011):

- **Architecture Principles, Vision, and Requirements**: this layer aims at capturing the necessary context, common vision and strategic objectives of the project. It gathers input for architectural analysis, as well as for the requirements and the fact-finding study.
- **Business Architecture**: artefacts belonging to this layer capture the business context and specify the operational details and functional capabilities of the project.
- **Information Systems Architecture**: captures architectural models of IT systems, both data & resources and software components.
- **Technology Architecture**: this part captures technology aspects and deployment for realizing the ICT solution.
- **Architecture Realisation**: this layer’s artefacts capture the costed roadmap and rollout of the project, as well as manage the binding statements for governing and steering the architecture.

---

4 ADM stands for Architecture Development Method
The aforementioned model has been aligned with the MLi project to produce the contextualized outline covering main top-level artefacts of the project. Those artefacts have been collocated within their respective layers and presented in Figure 5.

<table>
<thead>
<tr>
<th>MLi goals</th>
<th>Business Strategy</th>
<th>Vision document</th>
<th>Key Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation with other initiatives</td>
<td>Architectural Constraints</td>
<td>Requirements Analysis</td>
<td>Stakeholders Interviews</td>
</tr>
<tr>
<td>Motivating Scenarios</td>
<td>Software Services (Service Framework)</td>
<td>MLi hub</td>
<td></td>
</tr>
<tr>
<td>Participating Organisations and initiatives</td>
<td>Business Context</td>
<td>Tools &amp; Interfaces</td>
<td>R&amp;D technologies</td>
</tr>
<tr>
<td>Procedures and compliance</td>
<td>Specifications for proposals and tenders</td>
<td>LT Assets</td>
<td>Deployment Platforms</td>
</tr>
<tr>
<td>Stakeholders Interviews</td>
<td>Validation</td>
<td>Costed Roadmap</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Reference Architecture meta-model particularized to the MLi project.

The two upper layers have been discussed in the scope of the first WP2 deliverable (MLi D2.1 Early Design of the Reference Architecture and the Hub, 2014), while the Business Architecture and the Information System Architecture layers where covered in detail in the previous deliverable (MLi D2.2 Intermediate design and specification, 2015). Therefore, the focus of this document is mostly related to the refinement of the reference architecture. Taking into account recommendations and inputs from other MLi work packages, section 4 describes the challenges to be faced by the Hub and offers a revisited architecture pointing to infrastructure requirements, LT assets and existing tools and services as the baseline to build the Hub.
3. MLI HUB ARCHITECTURE REVISITED

Deliverable (MLi D2.2 Intermediate design and specification, 2015) provided an overview of the main LT initiatives, business-driven architectural requirements and constraints, as well as an initial definition of what we call the Language Value Chain and the intermediate version of the Reference Architecture for the MLi Hub. In this section the focus is on revisiting the proposed design in the light of the feedback from other work packages of MLi.

The design of the Hub aims to be generic and use-case agnostic. It serves as a reference and a starting point for instantiating concrete implementations, tailored to concrete business cases (as articulated in the business vision, stakeholders’ interviews and requirements analysis).

It is worth noticing that the MLi Hub serves a twofold objective. On the one hand, the MLi Hub can be seen as an environment to develop and generate sophisticated LT/MT services, being therefore a framework for LT developers and providers to tailor their offering. On the other hand, the MLi Hub should allow customers to use LT/MT services, taking into account the usage and consumption of those services as well as other supporting functionalities ranging from being a dedicated platform to a LT service marketplace or brokerage systems for LT.

The MLi Hub reference architecture is centred on the Language Value Chain (LVC) that is the main LT high-level building block that contains the LT and NLP components for providing a broad range of LT/MT services, from basic and atomic functionalities to high level more abstract ones. By going from simpler to more complex services, the LVC is increasing in complexity until the final LT/MT results reach the Business LT Consumer.

The LVC is based on and supported by the IT Infrastructure layer. This layer provides the necessary technical means for storage, processing and lower-level organisation of resources and services. This layer might as well be instantiated locally or outsourced to a specialized Infrastructure Provider in IaaS or PaaS manner. In fact, the IT layer could be deployed in one or several instances, allowing for example brokerage of services provided by different parties in their own IT physical or logical infrastructure.

In this sense we can read the architecture around the two axes:

- On the **horizontal “Language Value Chain”** the value is created by combining LT/MT/NLP services and components into more complex and higher level features that fulfil requirements of the end user.
- On the **vertical axis “IT Value Chain”** the value is created by providing more abstract access to the system functionalities, starting from low-level infrastructural services, through LT/MT/NLP domain-specific services until the high-level workflows and LT marketplace services for support of the business services and information and process integration within the Business Entity.

It is worth mentioning that the MLi Hub Reference Architecture has been revisited after the delivery of (MLi D2.2 Intermediate design and specification, 2015). The main changes with respect to the previous version resides on the inclusion of a Broker component on the upper layer and a fine-tuning of the building blocks description, due to the feedback from the development of the prototype of the LTi Cloud reported in deliverable (MLi D4.5 LTi Cloud Prototype, 2016). Figure 6 shows the final version of the Reference Architecture of the MLi Hub.
The main building blocks of the MLi Hub Reference Architecture depicted in Figure 6, reading the figure bottom-up and left-right, are the following:

- **IT Infrastructure** – This block provides technical and lower-level support for IT services delivery and technological provisioning for such areas as: IT physical infrastructure, infrastructure virtualisation (in case of the cloud deployment scenarios), storage infrastructure (either centralized or distributed), IT storage and processing services, such as resource management, IT service provisioning infrastructure and processing infrastructure. It is important to mention that in certain scenarios this block may be implemented in very different manners. For instance, the infrastructure might be in-house (a single instantiation of the application in a single infrastructure, distributed or not); or outsourced to a single third party infrastructure provider (in case of adopting IaaS or other cloud-based solutions); or covered by several third-parties’ infrastructures (in case of the Hub providing third-party services hosted onsite in the providers’ infrastructures). In this sense there is no “one-fits-all” IT Architecture scenario, but rather a tailored solution to the particular business use case.
  - **Physical Infrastructure and virtualization** – consists of low-level IT stack, such as servers and network infrastructure, solutions for dimensioning IT resources for certain capacity requirements, operating systems virtualisation and management infrastructure, cloud infrastructure and isolated environment management.
  - **Storage Infrastructure** – provides means for data storage in a potentially distributed infrastructure. In case of large-scale deployment, this might include distributed file systems, highly-scalable distributed database Infrastructure, redundant, failsafe and replicated storage systems, etc.
  - **Processing and Service Frameworks** – based on the lower level IT stack, this IT service infrastructure delivers services for abstract access...
to the IT Infrastructure, regardless of the physical and organisational distribution and configuration. In the context of LT/MT systems, this typically consists of:

- **Resource Repository and Management** – services for storage and accessing of concrete data items (resources) within the system. Examples of the resources hosted can be large corpus of language-pairs for machine translation, sentiment data, etc. One examples of a systems that could be used to implement this is Meta-Share.
- **Service Containers** – a technical backbone for the service framework and service marketplace, providing a container and isolation for flexible service deployment and scaling. Example of this is the service model transformations needed in a brokerage system such as the one provided by the LTi Cloud.
- **Batch and Stream Processing Infrastructure** – access to the computational resources for executing computational tasks on the underlying infrastructure. Examples of these processing engines can be Map-Reduce frameworks, stream processing software components, etc.), but in general it covers any processing infrastructure needed to run the services and the analysis performed in the upper layers of the architecture.

- **Infrastructure Providers** – general term for third party IaaS and PaaS providers who facilitate the IT Service infrastructure as a whole or a particular part of it. While instantiating the MLi Hub architecture might require certain IT Architecture effort, the physical part might be in-house or acquired as-a-service from a third party if the business and technical requirements are satisfied. In the current version of the LTi Cloud, the infrastructure for running the LTi Cloud services is minimal, but the LVC services are actually hosted in the infrastructure of the providers.
- **Language Data Providers** – data assets, central to the particular Business Entity that are subject to LT/MT processes in order to provide an added value business service. E.g. product descriptions in the e-commerce company to be translated into different languages in order to provide a cross-border business service.
- **Language Value Chain (LVC)** – the central high-level building block encapsulating the whole process of LT/MT, from reading the input data to producing concrete information that is used to provide a business value. It consists of specialized, domain-specific and scenario-specific technical components that facilitate and implement concrete LT/MT services. They are mostly business-driven, functional components that provide system-wide means to support and realize business objectives.
  - **Data Ingestion** – accessing and reading input documents either from existing systems (e.g. company’s CRM system, CMS, Social Media, etc.) or from specialized user interfaces. E.g. e-commerce site, user-uploaded documents or texts to translate.
  - **LT/MT Training** – one of the core activities within the language technologies required is to provide a custom, domain-specific trained models for concrete LT/MT task. Note that the training is not an integral part of the LVC, as many systems would use only translation services already trained if needed. However, as one of the core activities of the MLi, it is considered as an integral part of the LVC. The training consists of LT Resources (multilingual resources for training that might be stored and accessed via the Resource Repository), and training algorithms and
models (domain-specific algorithms for model training, typically particularized to some concrete aspect of LT/MT/NLP task. An example of this is MOSES. A more comprehensive study on this particular subject is done within WP3.

- LT Processing – technical components for performing concrete LT/MT/NLP services. Those might be small low-level NLP services (such as, text pre-processing, language detectors, segments, stemmers, parsers, etc.) or higher-level and more complex ones (e.g. named entity recognition, statistical translation, sentiment analysis or question answering systems). These services can be deployed in a single or several IT Infrastructure, as for instance in the case of the LTi Cloud, where similar services can be selected from different service providers. It is not coincidence that this layer stays on top of other LVC components, due to the fact that it typically needs other components, such as trained models, data ingestion or advanced text analytics in order to provide a concrete service. A comprehensive study of components and APIs has been performed in several MLi work packages, especially WP4 and the LTi Cloud prototype.

- Advanced Analytics – technical component providing higher level features and services for advanced analytics, not necessarily focused only on text analytics, such as AI systems, BI systems, large-scale media mining, etc.

- Data Access Usage – a set of software APIs for accessing processed data, such as translated documents, processed texts, extracted features or advanced search results. It provides data and service consumption interfaces to the Business Entity.

- Business Process and Existing Systems Integration – this building block encapsulates operational details and functional capabilities of the devised LT/MT architecture. It is the top-level service layer, and contains most notably:
  - System workflow orchestrator – component that combines technical LVC services into a concrete business workflow that might also include other actors, specific to the business context of the Business Entity. The objective is to realize defined business goals in a continuous manner. This component could be implemented using existing tools and frameworks (e.g. PANACEA) or use simpler mechanisms based on data flows, pipelining or messaging.
  - Broker – this component is needed when the system provides functionalities to mediate between different LT services offering similar characteristics. It is in many cases essential when the intention is to offer a Service Marketplace, but it is probably not implemented when the platform is intended for specific applications that use a known set of services. An example of brokerage is the LTi Cloud prototype.
  - Service Marketplace – it is an umbrella term for service aggregate platforms to offer LT/MT services to third parties. This block may not be necessary for simple LT installations, while may scale to a fully-fledged service marketplace offering a menu card of services including pricing and delivery mechanism. An example is the LTi Cloud prototype; which main objective is to offer a LT services marketplace.

- Business LT Consumer – this building block represents the final user of the architecture. The final user is the ultimate LT consumer that uses the results of the LT Value Chain for their own business objectives (i.e. MT of an e-commerce site, social media analytics, a developer using services of a LT Services Marketplace —such as the LTi Cloud—, etc.).
4. ARCHITECTURE INSTANTIATIONS

The design of the MLi hub architecture was aligned with various initiatives and LT projects in order to gain insights into the practical realisations of LT systems, as described in (MLi D2.1 Early Design of the Reference Architecture and the Hub, 2014) and (MLi D2.2 Intermediate design and specification, 2015). This brings the design closer to the real-word usage and makes it relevant for a broad spectrum of use cases, through the tailoring of design methodology, and resulting architecture. In the subsequent sections we provide details on how the devised architecture has been already applied and tailored to the concrete realisations. In this sense this is the reference point for the architecture customisation, and provides sample scenarios showing the extensibility and flexibility of this design.

Next, we also present a few considerations for the architecture evolution in the foreseeable future, showing the lines for the MLi Hub architecture advancement for the LT policy makers, standardisation organisations and LT industry as a whole.

4.1 LTi Cloud

LTi Could is a prototypical LTC Broker that provides a marketplace (or something similar to the App Store concept) that connects LT SaaS endpoints into one unified entry point for LT services. The main idea is to bring LT Vendors and LT Consumers together on one platform and overcome the decentralisation and fragmentation of Europe’s LT market. It is also a platform for rapid prototyping of LT systems that need to provide more sophisticated language stacks that combine various services from other providers. The LTi Cloud has been implemented in the scope of MLi and it is reported as deliverable (MLi D4.5 LTI Cloud Prototype, 2016).

Most notably, the LTC Broker also provides a set of interfaces for:

- universal API access despite various formats or invocation schemes of the underlying services,
- message formats translation,
- authorisation and pricing,
- SLAs and other service metadata.

In this sense, the LTi Cloud is a broker intermediary that connects service consumers with service providers, but in the current version doesn’t provide any LT services on its own.

Figure 7 shows an overview of the LTi Cloud architecture.
The main components are:

- Static Website providing a landing page and self-service access to the marketplace
- LT Store with repository of services available on the platform, together with their metadata, SLAs, pricing schemes, etc.
- LTC Broker, taking care of the service calls routing, message translation, load balancing and endpoints selection.

LTi Cloud is also seen as an instantiation of the MLi Hub architecture, but focusing only on the Business and high-level service integration, by providing elements of the MLi Hub such as the Broker, LT Marketplace and Access service facilities, while leaving the lower layers to the third party LT providers (the LTC Broker provides a minimal infrastructure to handle the brokerage services). It is envisaged that in the future services from third parties might be deployed in the LTi Cloud, but then other business and technical issues would come into play to ensure the correct SLA of the services (now up to the service providers, and not to the broker). Therefore, service providers are currently providing their services in their own infrastructure layer, as shown in Figure 8.

4.2 Atos CaptureAN

Another example of instantiation of the MLi Hub for the implementation of a Social Network monitoring services is CaptureAN⁵, developed by one of the partners of the MLi project (Atos). CaptureAN provides features such as sentiment analysis over Twitter and media feeds, among others. CaptureAN is using an underlying big data infrastructure for scalability purposes and provides high-value LT services to the Atos’ customers. The

---

⁵ http://capturean.com/
The development of CaptureAN has taken place alongside the MLi project, providing a good proof-of-concept for MLi. The architecture of CaptureAN is shown in Figure 9.

The CaptureAN design consists of two main blocks:

- **IT Infrastructure**: CaptureAN implements the IT Infrastructure of the MLi Hub following the same building blocks. On the Physical Infrastructure and Virtualization layer, CaptureAN uses specialized cloud infrastructure and storage facilities. For the Storage layer, CaptureAN proposes a set of NoSQL databases and indexed repositories to cope with the requirements posed by the streaming nature and high data volumes associated to social networks, along with a query façade to ease the access to the storage. On top of that, CaptureAN proposes the use of several streaming and batch processing tools and frameworks (Apache Hadoop, Apache Flink, etc.) in order to serve as foundation of the analytical process needed by the LT components in a big data environment.

- **The Language Value Chain in CaptureAN** is performed in the Application Layer. This layer implements the LVC by providing a set of steps and analytics using LT tools and services. The results of the process are delivered via RESTful services to the customers and visualized using a web-based dashboard.

This is a clear example of instantiation of the MLi Hub Reference Architecture. However, there are parts of the reference architecture that are not mentioned in the CaptureAN diagram. For instance, there is no need in CaptureAN to specify workflow tools or LT marketplaces in the architectural diagram. The LVC orchestration is done natively using data flow approaches. Therefore, it is not presented as a specific building block of the architecture, although undoubtedly is hidden into the LVC. The same applies to data providers or infrastructure providers. Data providers are the Social Network APIs (e.g. Twitter Search and Stream open APIs), and the infrastructure providers are expressed into the lower infrastructure layer.
4.3 FP7 Project “Pheme”

A third example of instantiation of the MLi Hub reference architecture comes from the Pheme project. Pheme is an EU funded project under FP7 that runs parallel to MLi and shares two partners: The University of Sheffield and Atos. Pheme is addressing veracity challenges to identify and track rumours in social media across languages. Therefore, Pheme needed to develop sophisticated LT algorithms that run in real time over social media streams to identify the spread of potential rumours as well as historical data to develop models for veracity. The Pheme architecture can be seen in Figure 10.

As in the previous example, Figure 10 shows the reference architecture for Pheme. At first glance it is clear that Pheme follows closely the MLi Hub reference architecture. This has been done intentionally as common partners of the two projects where interested in collaborating in these architectural aspects. In more detail, the Pheme implementation shows the following:

- **IT Infrastructure:** Pheme implements the IT Infrastructure of the MLi Hub following the same building blocks. The Physical Infrastructure and Virtualization layer is not specified in the diagram further, but it uses virtualized servers and several installations for different use cases. Therefore, Pheme can be deployed in multiple environments by specific infrastructure providers. The main difference is in the storage layer, where Pheme decided to place their raw data and semantic knowledge base instead of placing them into the upper processing layer. This decision was done on purpose to separate storage to the processing framework, although conceptually is very similar to the approach followed by the MLi Hub. Pheme proposes NoSQL databases and indexed repositories for the raw data, while uses a semantic database (GraphDB) as Knowledge Base to reason over the data and propose interesting findings. Pheme needs to analyse streaming and batch data coming from social networks, and it proposes the usage of batch and streaming processing engines to do so.

- **The Language Value Chain in Pheme** proposes several steps since the collection of the data from social networks to its analysis and further usage. LT tools and

---

6 http://www.pheme.eu/
services and sophisticated algorithms for event detection, cross-language and cross-media linking, along with training of veracity models for rumour detection and classification are part also of the LVC. The results of the analysis are used by the use cases (digital journalism and health) and visualized in dedicated dashboards.
5. MLI HUB GOVERNANCE AND SUSTAINABILITY

As explained in the previous sections, the MLI Hub proposes a flexible reference architecture for LT/MT that might be landed in different implementations depending on the existing requirements and needs. In this sense, it is not very clear how to envisage a costed roadmap and sustainability plans, given that there could be multiple implementations. Therefore, this task has been performed in close collaboration with the work done in WP4 (LTI Cloud prototype) and WP7 (deliverable D7.7 – Final Report on MLI Governance and Sustainability). The costed roadmap and sustainability model for an implementation of the e-commerce scenario is therefore explained in detail in D7.7, while D2.3 provides an introduction and a summary pointing to that deliverable for more information. It is therefore based on the sustainability aspects of the LTI Cloud and provides a summary that is reported in more extension in WP4 and WP7 deliverables.

- Governance: As explained in section ¡Error! No se encuentra el origen de la referencia., the LTI Cloud is a prototypical LTC Broker that provides a marketplace connecting LT SaaS endpoints into one unified entry point for LT services. In order to provide the governance of the LTI Cloud, the following structures and procedures has been proposed:

  o Legal model: The plan is to leverage on top of the LT-Innovate⁷, and create a LTI Cloud Association, as an International non-profit organization, as the most appropriate structure to be adopted.
  o Governance: An initial structure of the governing bodies and the decision making process of the LTI Cloud Association has been proposed. The idea is to have a simple an agile structure to enable fast decisions and at the same time providing the necessary links to all stakeholders.
  o Membership: The LTI Cloud Association members would constitute the General Assembly (GA). Institutions, organizations, and also individuals can be members. There will be Core Members (with operational commitments and responsibilities towards the Association), and Associate Members (contributing with components to the LTI Cloud, but not to the operation of the Association).
  o Resources: The human resources of the LTI Cloud Association are expected to evolve in time depending of the success of the association. For the launching phase, it is expected to have a minimum of 6 people in the staff (1 Manager, 2 Business Developers, 1 Product Manager and System Operator).
  o Operational aspects, IPR and SLAs: The current version of the LTI Cloud does not host any LT services in itself, but offers access to services hosted by third parties. Currently IPRs are not an issue, as the different components hosted in the LTI Cloud will follow a SaaS model. However, it is expected that in the future data and component ownership may become an issue. The platform should therefore study and develop a sound IPR management plan and payment schema to deal with these issues. In the same way, the SLA aspects in the terms and conditions of the initial prototype have been shifted to the terms and conditions of the actual component and service providers. If, as it is expected, the platform evolves in the future as a hosting platform for LT services, then SLAs would need to be agreed and ensured by the platform. This would require extra requisites in terms of quality of service, support personnel and

⁷ http://www.lt-innovate.org/
contractual agreements with customers that will be studied as the platform evolves.

- Marketing and communications: An initial marketing and communication plan has been devised in D7.7 for the launching and initial steps of the LTI Cloud.

- Sustainability model: If a framework such as the LTI Cloud wants to operate and be successful, a sustainability model including the foreseen launching and operational costs, along with expected revenue streams have to be devised. In this sense, in collaboration with WP4 and WP7, an initial estimation of costs and potential revenue streams has been provided in D7.7. A summary of this can be seen below:

  - Costs: The main costs come from the salaries of the LTI Cloud staff, marketing campaigns and material, the underlying infrastructure needed for operation and development, and the cost related to IPR and SLA management. The costs have been divided in two periods: the launching period and the production period. Based on the estimations made, the main goal is to get a substantial amount of members to be able to cover the costs, especially for the production period. A strategy to achieve a target of more than 60 members during the launching phase and around 180 members during the production phase is therefore paramount.

  - Revenue streams: The approach chosen during the launching phase is a hybrid model combining the fee-based and the sponsored contributions. In order to achieve eventually self-sustainability of the Association, the following revenue streams are in our plans: member fees, additional sponsorship, a % of the API sales for the distribution of LT solutions, and consulting activities.

An initial cost estimation of the revenue and cost models, along with the associated risks have been provided in D7.7.
6. CONCLUSIONS

This document presented the refined reference architecture for the MLi Hub taking into account feedback from the rest of the consortium and especially from the implementation of the LTi Cloud prototype done in WP4. The design of the reference architecture takes into account several use cases, paying special attention to the fulfillment of e-commerce scenarios. The reference architecture describes the main building blocks that form the technical and service infrastructure. The architectural analysis has been made in the context of previous WP2 deliverables building on an iterative methodology based on the reference Enterprise Architecture, comprising multiple aspects of the overall design of the platform.

The results presented in this deliverable are in conjunction with the work done in WP7, where the business aspects and costed roadmap to build an instantiation to the MLi Hub are covered.
REFERENCES