DELIVERABLE

LEARNING
COMPASS

19-Feb-2017

O2/A1: Design of online services

Revision: 0.9.1, *Draft*

Dissemination Level: Confidential
(only for members of the consortium)

Erasmus+	The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY	Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την παραγωγή της παρούσας έκδοσης δεν συνιστά αποδοχή του περιεχομένου, το οποίο αντανακλά τις απόψεις μόνον των δημιουργών, και η Ευρωπαϊκή Επιτροπή δεν φέρει ουδεμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που εμπεριέχονται σε αυτό.

DOCUMENT INFORMATION

Project Information

Project name	Composing Lifelong Learning Opportunity Pathways through Standards-based Services
Project acronym	COMPASS
Project number	2015-1-EL01-KA203-014033
Project web site	www.learning-compass.eu

Document Identification

Document title	O2/A1: Design of online services
Document type	Intellectual Output
Filename	COMPASS_O2_A1_v0.9.1.odt
Current status	Draft
Current version	0.9.1
Contact person	Ioannis Voyatzis
Dissemination level	Confidential

Version history

Version	Contributor(s)	Contribution
0.1 | 10-Apr-2016	A. Koutoumanos (ELOT)	First draft
0.2 | 20-Jul-2016	All partners	Brainstorming and feedback on Personas and Scenarios towards description of User Stories
0.3 | 10-Sep-2016	A. Koutoumanos (ELOT)	Documentation of all partners' feedback
0.4 | 3-Oct-2016	A. Koutoumanos (ELOT)	Use case model and system's architecture
0.5| 13-Dec-2016	A. Koutoumanos (ELOT)	Changes after QA process
0.9| 19-Feb-2017	A. Koutoumanos (ELOT)	Update after final decisions for the implementation of the COMPASS RULO repository

THE CONSORTIUM

No	Partner	Country	Contact Person
P1.	**Technological Educational Institute of Athens** ***(Coordinator)***	Greece	**Ioannis Voyiatzis** voyageri@teiath.gr **Cleo Sgouropoulou** csgouro@teiath.gr
P2.	**Technische Universiteit Delft**	Netherlands	**Said Hamdioui** S.Hamdioui@tudelft.nl
P3.	**Université de Montpellier**	France	**Mariane Comte** Mariane.Comte@lirmm.fr
P4.	**National Quality Infrastructure System / Hellenic Standardization Organisation**	Greece	**Fragiskos Krokos** fdk@elot.gr **Anastasios Koutoumanos** anastasios.koutoumanos@gmail.com
P5.	**Consorzio Interuniversitario Nazionale Per L'informatica**	Italy	**Paolo Prinetto** paolo.prinetto@polito.it
P6.	**Brandenburgische Technische Universität Cottbus-Senftenberg**	Germany	**Heinrich Vierhaus** Heinrich.Vierhaus@b-tu.de
P7.	**Tallinna Tehnikaulikool**	Estonia	**Peeter Ellervee** peeter.ellervee@ttu.ee

CONTENTS

1 INTRODUCTION

1.1 Purpose

The main purpose of intellectual output “O2: Online Services” is to provide an evidence of the validity of the suggested metadata instruments and approach, along with a showcase of the effectiveness of the developed metadata structure for the efficient adoption of standards-based metadata representation of Lifelong Learning Opportunities.

In particular, the core objective of O2 is to transfer the systems architecture design and associated metadata application into a set of software components and services that can allow and facilitate the exploration of alternative lifelong learning opportunity pathways. In order to meet this goal, a software toolkit will be developed that will be readily available to interested organizations in order to develop their learning opportunity structure based on a competence-enhanced Metadata for Learning Opportunities (MLO).

Furthermore, an online repository of MLO-based structures will be setup, which will facilitate the "design patterns" template and will be regularly updated with descriptions and insight from the usage of standards in metadata development in the ICT domain as well as suggestions for transferring to other domains. Both the developed toolkit and the metadata repository will be adopted for the targeted ICT domain.

The specific objectives are:

- to illustrate how different and diverse software applications can be developed using the proposed metadata architecture and infrastructure; and
- to create awareness and prepare case studies and guidelines for other software developers that would like to connect and re-engineer existing systems and services or build new ones for compliant applications in any sector.

In order to effectively generate this output, and according to the project’s Description Of Work (DoW), COMPASS shall work towards the development of:

1. a software component for description of learning opportunities, to be made freely and publicly available to learning opportunity providers (**CS-1**),
2. a free service for lifelong learners to leverage information about learning opportunities for building desired pathways for personal development (**CS-2**).

The main activities of this intellectual output are:

- **O2/A1:** Design of online services
- **O2/A2:** Development of online services
- **O2/A3:** Delivery and support of online services

The purpose of this document, titled “Design of online services”, is to document the results of the first activity of the second intellectual output of the project (O2/A1).

1.2 Methodology

Both the COMPASS metadata schema as well as the COMPASS online services shall be adopted for the targeted stakeholders. The specific objectives are (a) to illustrate how different and diverse software applications can be developed using the proposed metadata architecture and infrastructure; and (b) to create awareness and prepare case studies and guidelines for other software developers that would like to connect and re-engineer existing systems and services or built new ones for COMPASS compliant applications in any sector.

1.3 Audience and License

This deliverable is targeted mainly for the software development team of the COMPASS consortium. Furthermore, since some of the information described in the deliverable could be useful for other teams and individuals, the deliverable is released as a public document, as an interoperability instrument for sharing and exchanging information about learning opportunities, combined with associated descriptions of expected learning outcomes and competences, within the envisioned federation of repositories and related software systems. Finally, the document's audience includes practitioners and researchers working on related projects, as well as potential future consortia, organisations or individuals interested in adapting or reusing the COMPASS metadata architecture, as a whole or reusing some of its constituents.

Any re-use, sharing and adaptation of the work described and presented in this document is allowed under the terms of the Creative Commons "Attribution-ShareAlike" license, CC BY-SA International 4.0. This license allows sharing, redistribution, copy, remix, trasnformation, and building upon this work for any purpose, as long as appropriate credit is given, a link to license is provided and any changes that were made are properly indicated. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

1.4 Terms, definitions, and abbreviations

1.4.1 Learning Opportunity

A chance to participate in education or training *(source: MLO).*

1.4.2 Learning Opportunity Provider

An agent (person or organisation) that provides learning opportunities *(source: MLO).*

1.4.3 Competence

Proven ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and in professional and personal development (*source: EQF*).

1.4.4 Learning Outcome

What a learner is expected to know, understand, or be able to do after successful completion of a process of learning (*source: shortened from ECTS Users Guide*).

NOTE: The full proper term is “intended learning outcome”, and unless it is clear from the context that the reference is to the actual outcome of learning in a person, “learning outcome” should be understood to mean “intended learning outcome” throughout this documentation.

1.4.5 LOC

Learning outcome or competence.

1.5 Structure

This report is structured in five chapters:

- Chapter 1 holds introductory information;
- Chapter 2 holds a short description of all the planned COMPASS outputs and their relation to the designed online services, along with insight about next work and expected evolution of the online services during the lifecycle of the COMPASS project and beyond;
- Chapter 3 presents the overall approach for software design and methodology that has been used by the COMPASS partners;
- Chapter 4 documents the identified personas and indicative scenarios of their expected use of the COMPASS online services;
- Chapter 5 presents the specific requirements from the COMPASS online services, modelled as use cases;
- Chapter 6 presents the overall system architecture of the COMPASS online services;
- Chapter 7 holds references to relevant sources of information.

2 COMPASS PROJECT LANDSCAPE

The design of the COMPASS online services and all related information presented in this document is currently (Jan 2017) a work in progress. The implementation and deployment of these services, anticipates for an agile process towards building maturity and adopting to the outcomes of the pilot usage, trialling, and feedback collected by direct and indirect users. The following table (Table 1), along with the associated graph (Figure 1), highlight the main milestones of this process, as adapted by the Project Management handbook of the COMPASS project.

Table 1: Intellectual Outputs, Activities and Milestones

Code	Activity	Type	Due Month
IO 1: Lifelong Learning Opportunity and Pathway models			
O1.A1	Analysis of existing LO schemes	Report	Feb 2016
O1.A2	Information model development for representing LO metadata	Report	Jul 2016
IO 2: Online Services			
O2.A1	Design of online services	Report	Feb 2018
O2.A2	Development of online services	Report	Feb 2017
O2.A3	Delivery and support of online services	Service-Product	Aug 2018
IO 3: Tested and populated online services			
O3.A1	Development of Pilot Testing methodology and evaluation plan	Report	Feb 2017
O3.A2	Population of the online service with the LO from the participating institutes	Service Product	Dec 2017
O3.A3	Internal evaluation of the online services	Report	Dec 2017
O3.A4	Public evaluation of the online services	Report	Jun 2018
IO 4: MOOC on standards-based LO descriptions and the online service			
O4.A1	Development of MOOC - Part 1: Standards-based LO descriptions	Service Product	Sep 2016
O4.A2	Development of MOOC - Part 2: Online service	Service Product	May 2017
IO 5: Recommendations for Policy and Standardization development			
O5.A1	Design and implementation of Policy Suggestions Report	Report	Jun 2018
O5.A2	Design and compilation of Technical Report to Standardization Bodies and Implementers	Report	Aug 2018

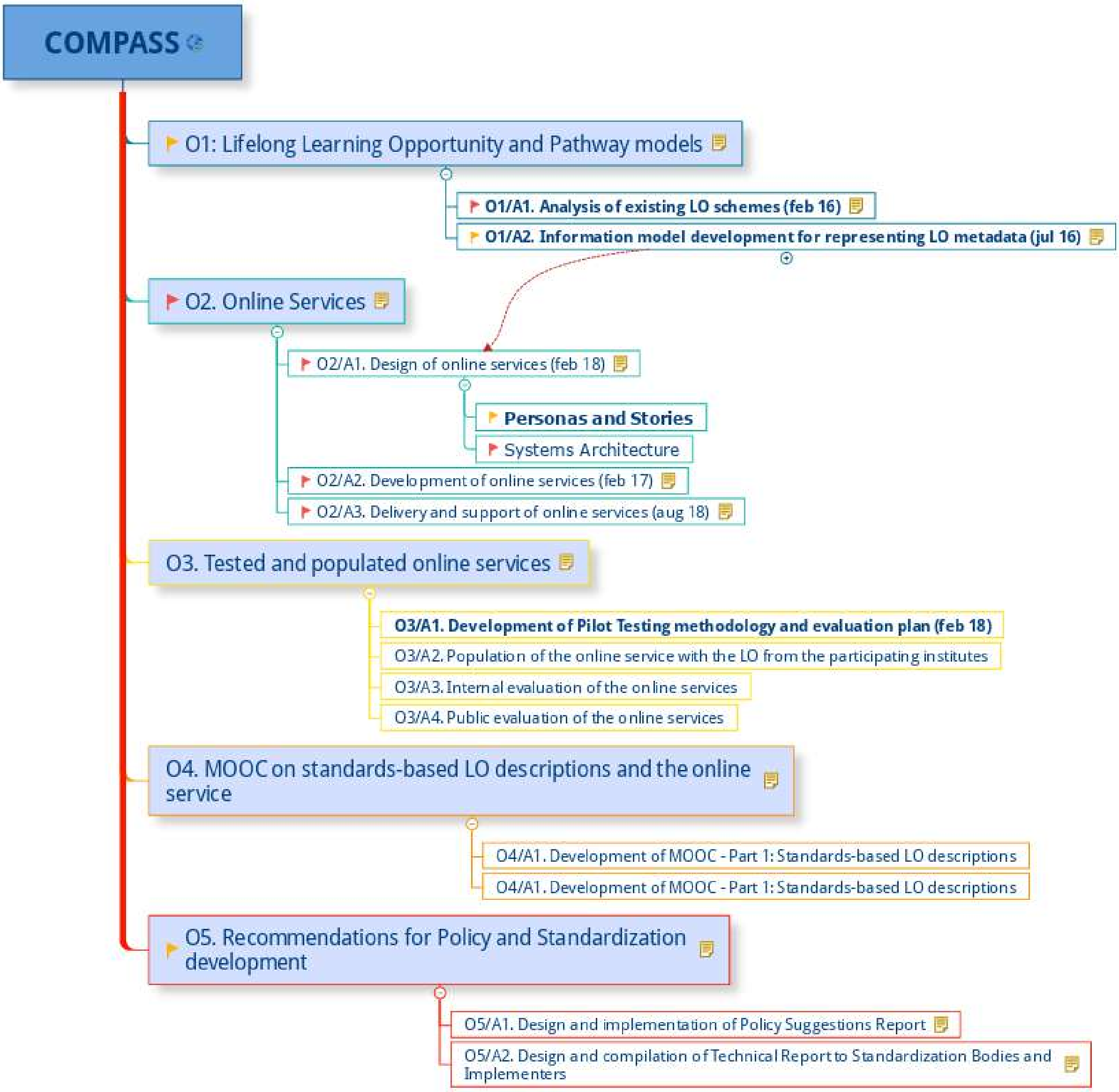

COMPASS
O1: Lifelong Learning Opportunity and Pathway models
O1/A1. Analysis of existing LO schemes (feb 16)
O1/A2. Information model development for representing LO metadata (jul 16)
O2. Online Services
O2/A1. Design of online services (feb 18)
Personas and Stories
Systems Architecture
O2/A2. Development of online services (feb 17)
O2/A3. Delivery and support of online services (aug 18)
O3. Tested and populated online services
O3/A1. Development of Pilot Testing methodology and evaluation plan (feb 18)
O3/A2. Population of the online service with the LO from the participating institutes
O3/A3. Internal evaluation of the online services
O3/A4. Public evaluation of the online services
O4. MOOC on standards-based LO descriptions and the online service
O4/A1. Development of MOOC - Part 1: Standards-based LO descriptions
O4/A1. Development of MOOC - Part 1: Standards-based LO descriptions
O5. Recommendations for Policy and Standardization development
O5/A1. Design and implementation of Policy Suggestions Report
O5/A2. Design and compilation of Technical Report to Standardization Bodies and Implementers

Figure 1: The COMPASS Intellectual Outputs

3 SOFTWARE DESIGN APPROACH AND METHODOLOGY

Before we can carry on with the description of the design of the COMPASS online services, it is interesting to describe the methodology and the tasks that have been reasoned essential to undertake in order to accurately document, validate and organise the requirements related to all anticipated stakeholders.

3.1 Architectural views of system’s design

According to the principles of the Unified Process for Software Engineering, the purpose of the software analysis and design is “to evolve a robust architecture for your system based on your requirements, to transform the requirements into a design, and to ensure that implementation environment issues are reflected in your design.”

The phases that are involved are the following:

1. identification of stakeholders;
2. capture of the user requirements and documentation of indicative user stories (scenarios), in order to validate the most important assumptions;
3. analysis of the essential use cases and design of the associated use case model using UML notation;
4. identification of the main software components; and
5. design of overall systems architecture and associated deployment model

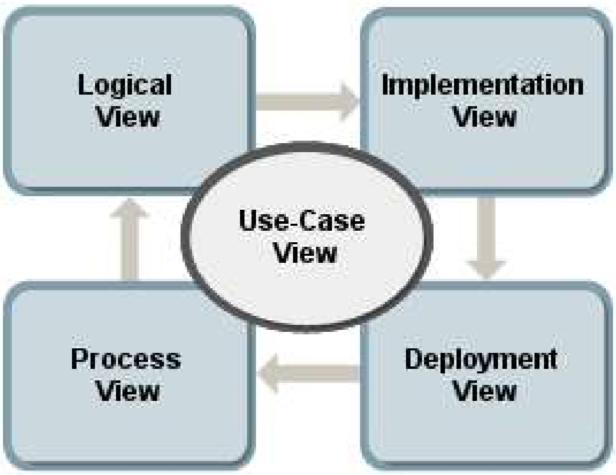


Figure 2: The “4+1” Views of Software Architecture

This software design follows the “4+1” framework [Kruchten 1995] that define a set of views of the software system’s architecture. The Four plus one (4 + 1) View Model describe the software architecture using five concurrent views, each of which addresses a specific set of concerns.

- the **Use Cases** view presents the users perception of the functionality provided by the online services.
- the **Logical** view describes the design's object model and is concerned with the functionality that the system provides to end-users. (sequence diagrams). Describes how the system is structured in terms of units of implementation. The elements are packages, classes, and interfaces. The relationship between elements shows dependencies, interface realizations, part-whole relationships, and so forth. Note: This view is mandatory when using the 4+1 Views of Software Architecture.
- the **Implementation** view presents the software's static organisation in the development environment, i.e. it describes how development artifacts are organized in the file system.
- the **Deployment** view describes the mapping of the software onto the hardware and is also considering how data will store physically. (data model).
- the **Process** view describes the design's concurrency and synchronisation aspects, i.e. it describes how the run-time system is structured as a set of elements that have run-time behavior and interactions. Run-time structure often bears little resemblance to the code structure.

This approach has been further adopted and has been further advanced and incorporated in the description of standard *"IEEE 1471 - Recommended Practice for Architecture Description of Software-Intensive Systems."*

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a product, taking account of the possibly conflicting requirements of the various user groups. Broadly speaking, software systems requirements engineering (RE) is the process of discovering that purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation (Nuseibeh et al, 2000). The project's life-cycle is not a one-way, a sequential design process (like the Waterfall model), but rather a more 'rational' and iterative one. Our approach towards RE has been based on the Rational Unified Process[1] (RUP), consisting of four phases:

1. Inception phase,
2. Elaboration phase,
3. Construction phase,
4. and Transition phase

These phases allow the process to be presented at a high level in a similar way to how a 'waterfall'-styled project might be presented, although in essence the key to the process is the iterations of different tasks that take place within all of the phases.

Requirements analysis is critical to the success of a development project. Requirements must be documented, actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. For the identification of user needs, the following activities have been carried out:

[1] http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process

- **Eliciting requirements:** the task of communicating with stakeholders and users to determine what their requirements are.
- **Analysing requirements:** determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.
- **Recording requirements:** Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

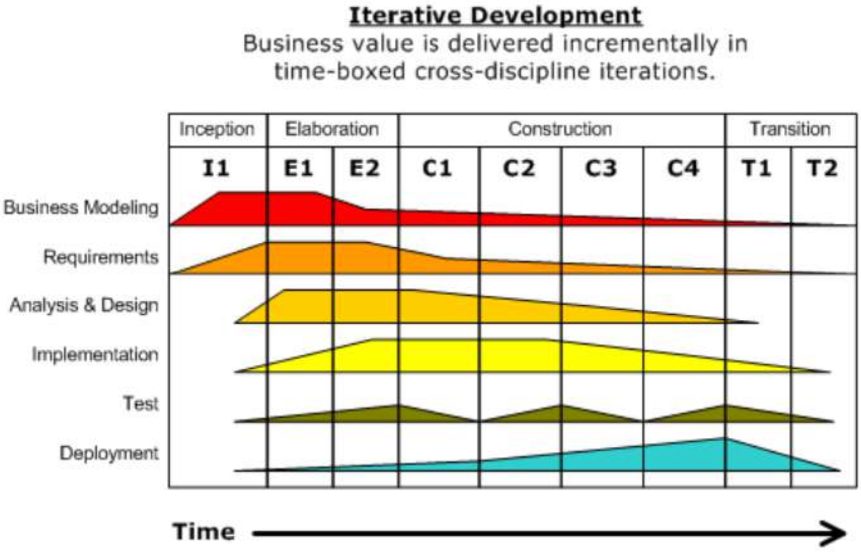


Figure 3: Phases and iterations

The main requirements analysis techniques used were agreed upon by the project partners and included the following:

- **Requirements documentation:** all identified requirements, suggestions and issues identified by project partners were documented and served as the basis for an internal live document, in the form of wiki pages, used as a communication and collaboration tool by all partners in order to reach a common consensus of the goals, objectives and procedures for the requirements validation.
- **Stakeholder engagement:** it was clear that strong engagement should be sought for the validation of identified requirements, with representatives from all main user- groups. In that sense, most partners organised events for getting input and feedback from all identified stakeholders, including learners, trainers, practitioners, representatives of competent organisations, policy makers, etc. from all participating countries. The wiki pages provided guidelines for alternative scenarios for these events, being mainly in three forms: (a) participatory workshops, (b) interviews (face-to-face or online) and (c) online survey.

This document reports on the results of these activities and presents the design of the COMPASS online services in order to address all identified requirements.

3.2 Requirements engineering: key concepts

The following paragaphs introduce the key concepts related ot requirements engineering, which are used throughout the next parts of the document.

3.2.1 (Software) System

A group of things or parts working together or connected in some way to form a whole. A software system is made up of software, hardware, and digital information, and provides its primary value by the execution of the software. A software system can be part of a larger software, hardware, business or social solution.

3.2.2 Stakeholders

Stakeholders are persons or organizations (legal entities such as companies, standards bodies) that have a valid interest in the system.

3.2.3 Requirements

What the software system must do to satisfy the stakeholders.

3.2.4 Story

A description of a way of using the system that is of value to a user or other stakeholder.

3.2.5 User

A stakeholder who interacts with the system to achieve its goals.

3.2.6 Use case

A use case is all the ways of using a system to achieve a particular goal for a particular user. In essence, a use case is a classifier which specifies behaviour of a system by describing a set of sequences of actions performed by the system to yield an observable result of some value to one or more users of the system. In other words, each use case describes a unit of complete and useful functionality that the system provides to its users. The detailed behaviour of a use case can be described by any means and techniques intended to define behaviours, in a separate diagram or textual document such as interactions, activities, state machines, pre- and post-conditions and natural language text (use case narrative).

3.2.7 Use Case 2.0

Coined up by Ivar Jacobson (Dec 2011), Use-Case 2.0 describes a scalable, agile practice that uses use-cases to capture a set of requirements and drive the incremental development of a system to fulfil them.

3.2.8 Actor

An actor defines a role that a user can play when interacting with the system. A user can either be an individual or another system. Actors have a name and a brief description, and they are associated to the use cases with which they interact. Cockburn (2000) distinguishes between primary and secondary actors. A primary actor is one having a goal requiring the assistance of the system. A secondary actor is one from which the system needs assistance to satisfy its goal.

3.2.9 Roles

Each actor defines a coherent set of roles users of the system can play (Rumbaugh et al., 1999). When an external entity interacts with the subject, it plays the role of a specific actor. That single physical entity may play several different roles, and a specific role may be played by single or multiple different instances.

3.2.10 Use case Diagram

A diagram showing a number of actors and use cases, and their relationships.

3.2.11 Use case Model

A model of all of the useful ways to use a system, and the value that it will provide. A use-case model is primarily made up of a set of actors and a set of use cases, and diagrams illustrating their relationships.

3.2.12 Use case Narrative

A description of a use case that tells the story of how the system and its actors work together to achieve a particular goal. It includes a sequence of actions (including variants) that a system and its actors can perform to achieve a goal.

3.2.13 Flow

A description of some full or partial path through a use-case narrative. There is always at least a basic flow, and there may be alternative flows.

3.2.14 Basic Flow

The description of the normal, expected path through the use case. This is the path taken by most of the users most of the time; it is the most important part of the use-case narrative.

3.2.15 Alternative Flow

Description of variant or optional behaviour as part of a use-case narrative. Alternative flows are defined relative to the use case's basic flow.

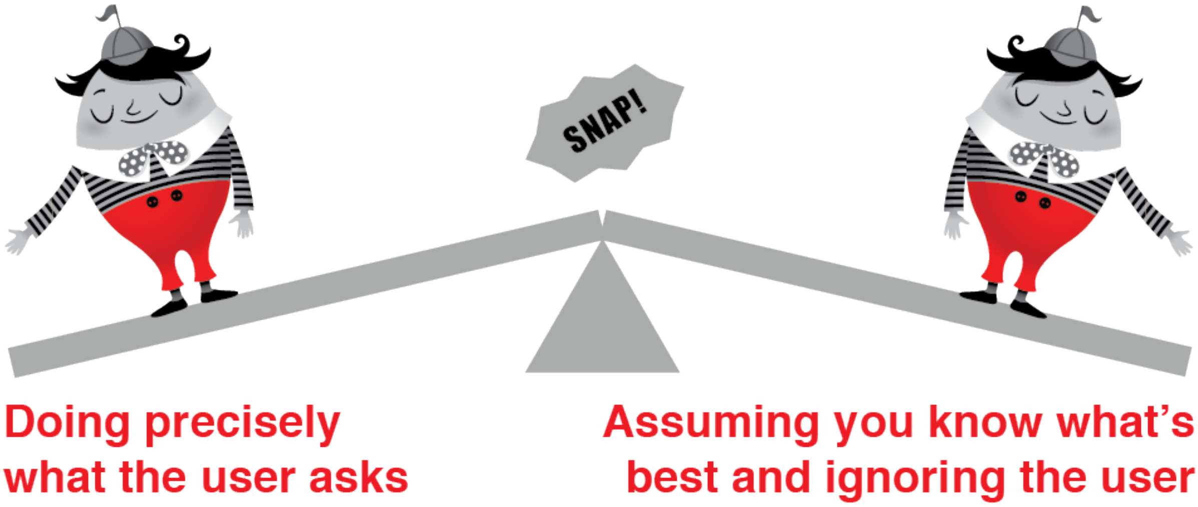
4 PERSONAS AND SCENARIOS

The first step of our approach towards the overall systems' design has been the identification of all interested parties (stakeholders) along with a description of their main expectations from the planned services. The outcomes of this work are documented in this chapter.

4.1 Personas and stories

We make use of the concepts of "personas" and "user stories". Software requirements is a communication problem: those who want software must communicate with those who build it. If either side dominates the output is poor:

- *Business dominates?* functionality and dates are mandated with little regard for reality or whether the developers understand the requirements.
- *Developers dominate?* technical jargon replaces the language of the business and developers lose the opportunity to learn from listening.


SNAP!
Doing precisely what the user asks
Assuming you know what's best and ignoring the user

Attribution: Alex Cowan

Figure 4: The twin anti-poles of design failure

This is exactly where user stories come to assist, as a basis for communication and discussion:

- **Personas** are fictional characters created to represent the different user types that might use a site, brand, or product in a similar way"[2]. Personas are useful in considering the goals, desires, and limitations of users in order to help to guide decisions about a service or product, such as features, interactions, and visual design of a website.

[2] https://en.wikipedia.org/wiki/Persona_(user_experience), http://www.usability.gov/how-to-and-tools/methods/personas.html

- **User stories** are part of an agile approach that helps shift the focus from writing about requirements to talking about them. The approach we have adapted for COMPASS includes a description of the task at hand for a specific persona, the actions and tools involved at the current situation and a discussion of the "COMPASS solution", i.e. an identification of the ways the COMPASS methodology and associated software services can facilitate interested parties to maximise the benefits from a systematic, standards-based approach to description of learning opportunities, learning outcomes and competences.

User stories support participatory design as opposed to empirical design:

- **Participatory design:** The users of the system become part of the team designing the behaviour of the system
- **Empirical design:** Designers of the new system make decisions by studying prospective users in typical situations

4.2 Personas

This is an extract of the Personas identified at the 2nd meeting of the project's partners.

1. Marko - Head of SP, TTU
2. Kadrin - Administrative staff at Office of International Relations, TTU
3. Daniel - PhD student
4. Marta - Engineer at NXP
5. Sonia - Computer Engineer IMEC, BE
6. Giorgio - QA unit, Rome University
7. Homer - Professor of Computer Engineering, TEI-A (same as Marko)
8. Juliet - International Relations, University of Montpellier
9. Hans - Student, TUB
10. Anna - Finished High School (Madrid)
11. Apo - Vocational Standard Office (EE)
12. Jane - International Relations Coordinator (Mariane)
13. Nicoals - 17 yo student
14. Michele - Nicolas father
15. Peter - (same as Homer and Marko)
16. Monique - Greek MoE, responsible for Ploigos
17. Laura - Professor of Computer Science, Turin
18. Francesca - 18 yo (Giuseppe)
19. Albert - MoE Policy Maker, responsible for funding
20. Teo - MSc candidate in Greece (Cleo)
21. Gregory - HR Dpt, CY
22. Bernard - Head of IT team for tool for University of Montpellier

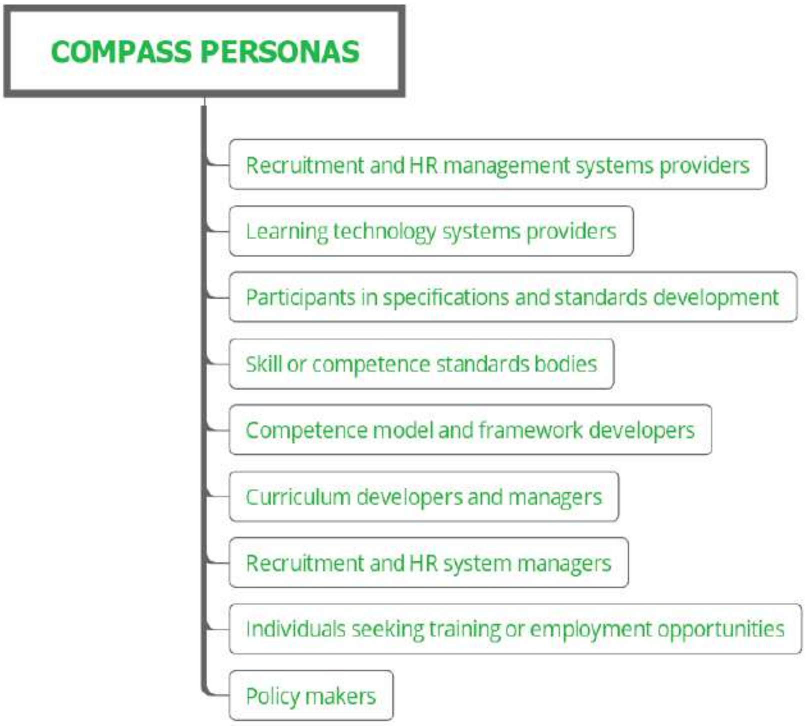

COMPASS PERSONAS
Recruitment and HR management systems providers
Learning technology systems providers
Participants in specifications and standards development
Skill or competence standards bodies
Competence model and framework developers
Curriculum developers and managers
Recruitment and HR system managers
Individuals seeking training or employment opportunities
Policy makers

Figure 5: Identified stakeholders groups

The following paragraphs provide indicative stories describing in narrative form the expected interest of the services from hypothetical users representing different stakeholder groups. Each story is a brief narrative, in the form of a scenario, that describes the hypothetical use of a system. In one or more paragraphs it:

- tells who is using the system and what they are trying to accomplish;
- provides a realistic, fictional account of a user's constraints: when and where they are working, why they are using the system, and what they need the system to do for them;
- describes any relevant aspects of the context in which the user is working with the system, including what information the user has on hand when beginning to use the system;
- indicates what the user regards as a successful outcome of using the system.

4.2.1 Bernard - IT team of tool for University of Montpellier

Bernard is responsible for the design of a new software system that shall be used for managing the descriptions of learning opportunities and their publishing on the online course catalogue of the University.

The MoE has also mandated that some of the information of the course description is provided in another system, for evaluation and QA process.

Furthermore, there is the need to share and exchange information about LO with foreign educational institutes with which the University of Montpellier has signed learning aggreements.

Last, but not least, course infromation must exist in Ploteus

Bernard tries to create an information model that will hold all the information, however it is very challenging to do so and he knows that his system will not be interoperable. He keeps searching for available solutions and discusses this online with colleagues abroad.

Until he finds what seems to be a perfect solution! The COMPASS approach and tool. He asks for a new community to be cofigured for his university on the central COMPASS server and asks the system administrator for small changes to the metadata Application Profile.

4.2.2 Daniel

Daniel is a PhD student at TU Delft. He has recently finished his MSc. He's doing his PhD at the Computer Engineering Lab. His research is about reliability of memory and needs background information about the topic. He speaks Ductch (native) and English.

4.2.3 Marta

Marta is an Engineer at NXP (NL). She is Spanish and also speaks English. She has been working as computer hardware engineer for two years. For new projects she needs new knowledge for relevant domains.

4.2.4 Marko

Marko is head of study programme at TTU (EE). He speaks english and estonian. As part of his work he advices exchange students about selection of courses at other universities is suitable He is professor of CS.

4.2.5 Kadrin

Kadri is administrative employee of the International Office at TTU. She helps students to look for other universities. She guides, suggests opportunities abroad. Also suggests foreign students for opportunities at home university.

4.2.6 Teo

Teo has completed his studies at Dpt. of Informatics and he decides that he wants to become a data scientist.

He searches on the web for the competences that a data scientist should have and finds that data science employs techniques and theories drawn from many fields within the broad areas of mathematics, statistics, operations research;

So Teo tries to identify which competences he has to acquire in order to be a data scientist. He ends up with a list of 4-5 competencies that he’s lacking;

He searches for universities that offer those competences.

After some search he finds two courses that offer the desirable knowledge, skills and competences in order to be a data scientist.

4.2.7 Mark

Mark is a first year PhD student at TU Delft in the Netherlands.

In this story we are going to use as an example of the *Learning pathway* of a PhD student in computer engineering.

4.2.7.1 Current situation

Mark recently finished his masters and is now doing his PhD at TU Delft at the Computer Engineering Lab. His research is about the reliability analysis and mitigation techniques of memories. He needs some background information, such as memories, reliability mechanisms, and mitigation techniques. Furthermore, for his PhD he also needs to fulfil a number of courses in the field and also more soft skill courses (presenting, technical writing, etc.). Therefore, he is trying to find courses, which will give him enough background information. He uses Google to find courses that will provide him the background information and also asks advice from his supervisor and other people at the university. The supervisor recommends him a summer school. The PhD council of the TU Delft also offers courses on soft skills for him to take.

4.2.7.2 The COMPASS way

Mark finds out about the new online service at http://learning-compass.eu. He decides to create a personal user account. He fills in information about his studies, background, and learning goals. Using the platform he is able to find some specialized courses in his field that will offer him background information. Also, he will get in touch with professors that work in this area. Finally, he also has a look at some soft skill courses and based on rankings, he is able to select remarkable (in terms of reviewing scores) on soft skills, such as presenting and technical writing.

4.2.8 Martha

Martha is a second year PhD student at TU Delft in the Netherlands.

In this story we are going to use as an example of the Learning pathway of a second year PhD student looking for expertise in possible future job qualification at industry.

4.2.8.1 Before COMPASS

Martha has completed her first graduate year at TU Delft and has gained the background knowledge on his research field. She is now interested to know which qualifications are needed to be a test engineer at the industry. She will use this information on choosing some elective courses during her remaining years of PhD studies at the university. She regularly looks into job advertising websites, job descriptions in companies and checks for needed skills of a test engineer.

4.2.8.2 The COMPASS way

Martha has already created a personal profile in the COMPASS platform. She enters all her skills inside the platform and now she is able to compare her profile with an existing test engineer profile in the COMPASS framework. In this way she can understand the lacking qualifications in her profile. In search for these qualifications, she can observe the universities, industrial training workshops, institutional courses which provide and give such skills. She can register to the courses by using the COMPASS platform in the appropriate period of the year and obtain the wanted skills.

4.2.9 Francesca

Francesca is an Italian girl (18 yr old) who has just acquired a Diploma from a Scientific High School in Rome, Italy. She is very good in maths and science, while she doesn't like history or philosophy. She is not very confident with ICT and speaks Italian and French.

She is interested in enrolling to a BsC career in Electronics Engineering. She is willing to move abroad but she has constraints on the language: speaking French, she can go to France, Belgium and Switzerland. She wants to search for the best course that fits her needs (in terms of learning outcome, job opportunities, location, entrance fees, possibility for exchange studies – Erasmus).

4.2.9.1 Before Compass

Francesca will deal with the following systems:

- *Social networks*: she will join group of abroad students and ask for their feedback
- *Google/Internet*: she will start by googling “Electronics Engineer in France” and learning about: professional figure, how it is recognized-employed, how studies are organized (Bologna process), how exams are graded, if attendance to courses is compulsory, how courses are taught, how lab activities are organized... She will repeat the same analysis for Belgium and Switzerland
- *Universities websites*: finally she will pick up some universities (also choising a city where to live) and crawl their website in search of information such as courses list, ECTS given, exams grading; she has to compare information "by hand".

4.2.9.2 The COMPASS way

Surely Francesca will save time and energy by easily compare Electronics Engineering courses through platform's filtering process. Would she receive generic information about how it is different to study in Italy and France of Belgium or Switzerland?

4.2.10 Marko

Marko is professor at the Department of Computer Engineering at Tallinn University of Technology and the head of the Computer Systems Engineering international master programme. In addition to the regular research and teaching activities, he also advises students selecting their courses at the home university and at the other universities during exchange studies. On top of that, he is also responsible for analysing and developing the study programme.

4.2.10.1 Before COMPASS, university students

Marko as the head of a study programme advises students to select courses for their exchange studies at a foreign university. That is, whether the courses a student would like to study abroad satisfies requirements of the study programme. For this, study outcomes and description of a course are needed to see how well the course will fit into one or another module of the curriculum. For the initial advise/decision, an online access to course descriptions is not so important but is essential for the final decision. It should be noted that the main problem is not to find and to access those descriptions but the different styles how the course is described at different universities.

A typical procedure advising the out-going students looks like as follows currently:

1. Marko receives either an e-mail or a table with the list of courses the student would like to study at another university.
2. To find out how well one or another course fulfils study programme requirements, the title of a course is not enough, and study outcomes and course descriptions are needed, at least. Typically the information can not be found at once online and some iterations with the student are needed to refine information. Especially because the way a course has been described, is different at different places. Also, Marko checks the list of courses against the performance of the student (what has been studied already).
3. When everything is OK, the application is approved.

4.2.10.2 The COMPASS way, university students

With the access to Compass tool/database, the second step will be simplified significantly. The following two main benefits can be outlined. First, because of the unified descriptions, comparison and matching of the courses with will be easier. Second, it will be much easier to suggest the student to take an additional course (or two) or to replace one of the initial selections with a more suitable for the goal of the study programme.

4.2.10.3 Before COMPASS, incoming students

For incoming students, Marko's advise is also sometimes asked. Typically a request from a potential exchange student is forwarded to him by the student exchange specialist of the university (Kadri). The information a student is looking for is which of the courses taught at TTÜ would be the best for his/her study programme. To make a good advise, Marko should know what the student has studied already and what are the expectations of the study programme. The overall procedure is very similar to the one used with outgoing students. The main difference is in the iterations of the second step – instead of looking for what is needed for the study programme he is supervising, he is now suggesting what may be useful for the student's studies. For that, sometimes direct contact, typically via e-mail, with the head of the study programme at the foreign university is needed.

4.2.10.4 The COMPASS way, incoming students

With the availability of Compass tool, the possible choices will be more visible and less guessing will be needed. Even more, Marko may find suggestions for wider are of topics because of enhanced ways for comparisons.

4.2.10.5 Before COMPASS, update of study-programme

Being the head of the study programme, one of the important tasks for Marko is to follow how up-to-date the study programme is. For that, analysis and comparison of the other existing study programmes is necessary but the difficulties currently are in the way different programmes are described at different universities. Of course, well-known example programmes from international organisations, e.g., IEEE or ACM, are useful but it is not easy to find information about interdisciplinary topics. This especially useful for smaller countries like Estonia where the need for one or another special knowledge is needed but the need for graduates is not large enough for a full study programme. In cooperation with the local industry and with the help of Compass tools, Marko can search for suitable module and/.or course examples from the other universities.

4.2.10.6 The COMPASS way, update of study-programme

This case has an important similarity with the case when advising outgoing students. Namely, it will be possible to advise the student to take a course or two related to the needed subject, especially when it will be useful for their current/future job. The university regulations in Estonia allow the students to take courses that are not directly related to their curricula, e.g., listing the courses in the free study module.

4.2.11 Kadri

Kadri is manager at the Mobility Centre of Office of Academic Affairs at Tallinn University of Technology. Her job is to advise and to coordinate student exchange – searching for information, consulting with heads of study programmes, organizing needed paperwork, etc.

4.2.11.1 Before COMPASS

As the coordinator of the student exchange, Kadri advises the student to look for suitable universities and to contact study advisor when needed.

4.2.11.2 The COMPASS way

The use of Compass tool will simplify her job by directing students directly to the possibility to search and compare courses, instead of finding contacts at one or another university. Of course, her main responsibility would be still handling the paperwork related to the study exchanges but very probably she can coordinate more students in future than today. This would be because of the possibility to search for the suitable courses by the students themselves. In addition, there will be less need to contact Marko or another head of the study programme to ask one or another detail regarding the suitability of one or another course in the first phases of setting up the study exchange plan.

4.2.12 Jane

Jane is a departmental International Relation coordinator at Polytech Montpellier. She is in charge of the validation of learning agreements for incoming and outgoing students of her department. Although she is familiar with communication technologies, she spends much time looking for up-to-date information, equivalent keywords, prerequisites, number of hours, credits and so on and so forth. With time, she can read programs in French, English, Italian, Spanish and Portuguese.

4.2.12.1 Before COMPASS

Jane receives applications from foreign students coming to the University within the frame of an exchange program and she should get from the transcripts of records what the level of the student is, and whether he/she has built a consistent study program (neither courses already validated in his/her home universities, nor courses where he/she lacks prerequisites). Therefore, she needs to check the « equivalence » of program / course / level / contents / ECTS credits / background knowledge, etc. that the applicant has already validated. She is in trouble finding references for level, duration, content, outcomes of the courses already validated by the applicants. Very often, she just finds some general keywords, not even in a language she can understand, so she needs to have an individual interview with each student and make a global oral evaluation of the student's knowledge in order to guide him/her in his/her choice of courses.

In the other way round, for her home students going to study in a partner university, she needs to check that their program is a coherent continuation of their training. In the same way, it is very difficult and time consuming for her to estimate whether the chosen courses fit with the students' previous knowledge without being redundant.

4.2.12.2 The COMPASS way

Jane can easily find course descriptions with meaningful information on the level, credits… through to the COMPASS platform. She can feed comparative data for the crouses provided by the partner universities of her university and reuse it the next years (for example, she can mark all the courses equivalent to the ones given in her Engineering School and it helps her checking the consistency of the pedagogical programs of the incoming and outgoing students). She can also benefit from other coordinators comparative marks and extend her knowledge of equivalent, or complementary, courses in various European Universities. In addition, given that the students can check these equivalences and complementarities beforehand, the learning agreements she receives are now much more consistent and she spends much less time revising them.

4.2.13 Nicolas

Nicolas is a French guy (17 yr old) who will finish his High School next year. He knows he wants to study in scientific field (maybe between computer science and bioinformatics) but he doesn't know which will be his professional opportunities and which knowledge is required before enroll (if any).

4.2.13.1 Before COMPASS

Nicolas will deal with the following systems:

1. *Open days*: University of his city invites High School students to visit it and students promote programs, opportunities (e.g., Erasmus, scholarships), learning outcomes -> through posters and report they highlight professional opportunities and data about percentage of workers after 1 yr from the Degree, average income, geographical distribution, type of employee and type of qualification.
2. *University website*: Nicolas will check the website of his nearest university (the one of his city and other two or three well-known universities in France) and compare "by hand" different BsC courses in the ICT area: he will compare courses thaught, statistics about exams passed and failed in the previous years, professional opportunities, language of the course, e-learning, opportunities for Erasmus or internships.

4.2.13.2 After COMPASS

1. *Open days:* will still be attended
2. *University website:* comparison of LO will be easier and faster through a filtering process among indicators

4.2.14 Michel

Michel is Nicolas's father. He did not go to the University so he is not familiar with technical terms.

4.2.14.1 Before COMPASS

Michel would like to figure out which types of job his son Nikolas will access though the different programs offered to help him choose a training. It is very difficult to manage his way through the Google searches and links to institutional web-sites, so he gives up...

4.2.14.2 The COMPASS way

Thanks to COMPASS, Michel and Nicolas can build a nice pathway for Nicolas. Nicolas is more focused on the course contents (whether he will be interested or not in the short term), while Michel feels only concerned by the job possibilities at the end of the pathway. Thanks to COMPASS, both gain confidence in the course program Nicolas is building and the career opportunities that will be accessible to him. In addition, thanks to the clear equivalence of programs given by the platform, Nicolas will even manage to convince his father that he can spend a semester of studies in a foreign university within the same pathway, so his big problem with English will be solved!

5 USE CASE MODEL

The subject of Use-Case 2.0 is the requirements, the system to be developed to meet the requirements, and the tests used to demonstrate that the system meets the requirements. At the heart of Use-Case 2.0 are the use case, the story and the use-case slice. These capture the requirements and drive the development of the system. The following figure shows how these concepts are related to each other. It also shows how changes and defects impact on the use of Use-Case 2.0[3]

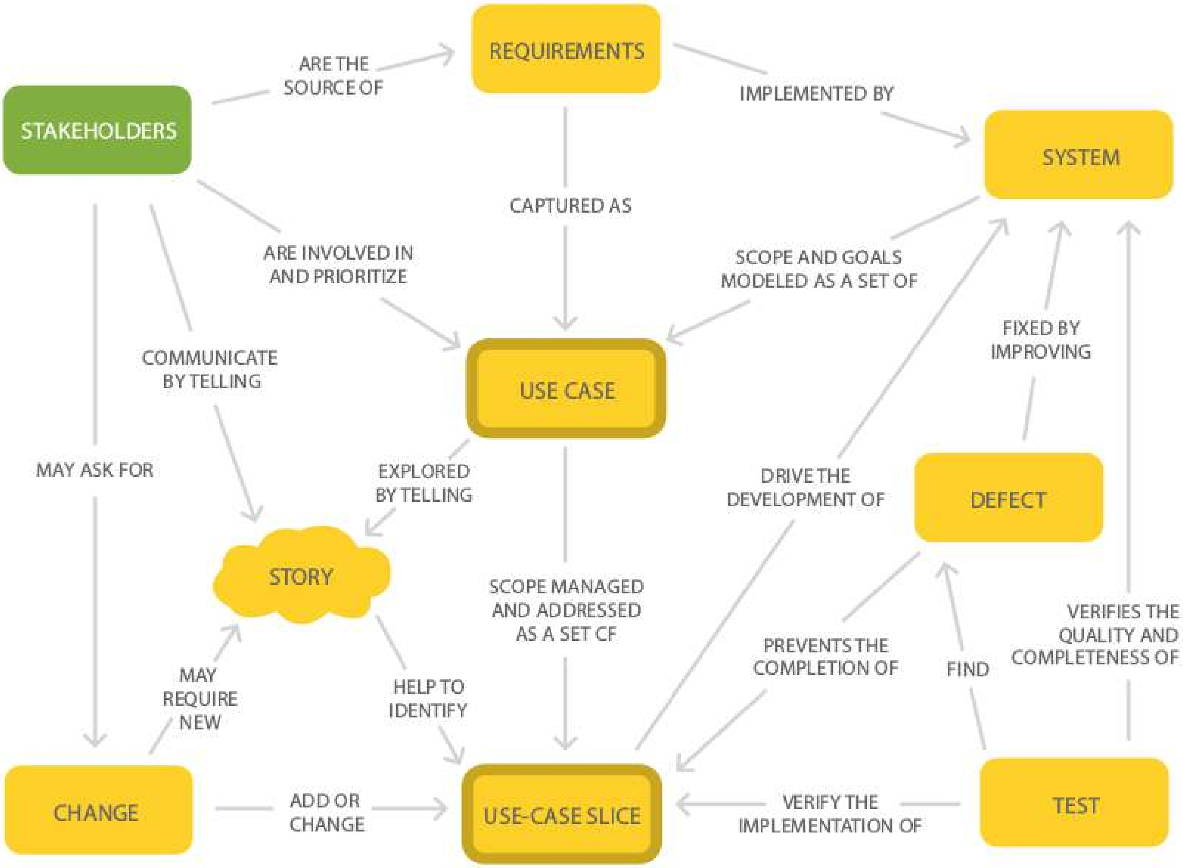


Figure 6: Key concepts of the Use-Case 2.0 approach

Telling stories bridges the gap between the stakeholders, the use cases and the use-case slices. It is how the stakeholders communicate their requirements and explore the use cases. Understanding the stories is also the mechanism for finding the right use-case slices to drive the implementation of the system.

[3] http://www.ivarjacobson.com/uploadedFiles/Pages/Knowledge_Centre/Resources/White_Paper/Resources/Use-Case%202_0_Jan11.pdf

A use case is all the ways of using a system to achieve a particular goal for a particular user. Taken together the set of all the use cases gives us all of the useful ways to use the system. These are described in the following two sections of the report.

5.1 Actors

The identified actors for the RULO[4]™ repository of are the following:

5.1.1 (Anonymous) Users

Anonymous users shall be able to browse through the contents of the repository. It is expected that a large audience of the services will happen to reach the repository with their web browsers through search engines (e.g. Google). After visiting the repository, interested audience should be able to browse through an adequate set of functionality that will without the need for setting up a user account.

5.1.2 Registered Users

Regular users of the repository could be prompted to register with the system. In order to do so, it should be evident that a set of functionality shall be available only to registered users.

5.1.3 Editors of RULOs

The repository shall provide specific support for the people who undertake the management and editing of the descriptions of RULOs.

5.1.4 Moderators

The repository shall provide specific support for moderators who overview the feedback from anonymous and registered users and coordinate the update of existing definitions of RULOs.

5.1.5 Other Systems

The repository shall be able to inter-communicate with other relevant systems, like for example qualification systems.

5.2 Resulting use cases

The essential use cases for the repository of RULOs are presented in the following diagram using the UML (Unified Modelling Language) notation [FOW03].

[4] Linked Lifelong Learning Opportunities Competences Learning Outcomes

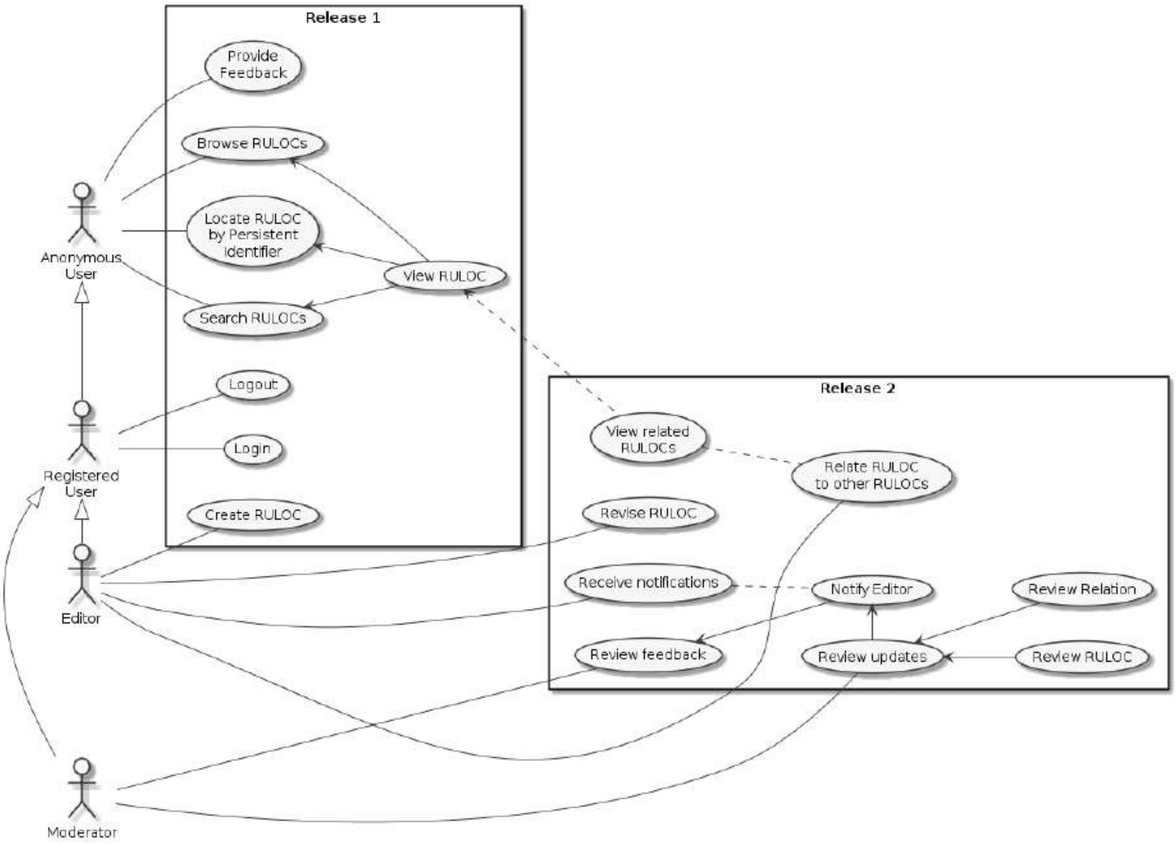


Figure 7: Use case model (Releases 1 & 2)

The following paragraphs hold the specification of the identified use cases, categorised by planned releases. The use cases are labelled in the form [UC x#], “x” marking the release (“a” for 1st version, “b” for 2nd, etc.) and # being a sequential integer.

5.2.1 1st Release

The use cases that shall be implemented for the 1st release of the system are the following.

5.2.1.1 UC a-01: Search RULOs

Summary:	Users shall be able to search the system for units of RULOs given some specific criteria or simply by going through the list of all available items in the repository.
Release:	Version 1
Actors involved:	All users (humans)
Preconditions:	-
Basic flow:	1. The user visits the repository in the web browser, through its URL or a link from another website. *2. The web server displays the main page with a (big) search field and an indication of available items in the repository.*

	3a. The user provides the search terms and hits the Search button; *or* 3b. The user presses directly the Search button (without any terms); *or* 3c. The user provides the search terms directly in the URL *(see 7a)*. *4.1 The system presents the search results in the form of an ordered list, containing the items that match the given search terms (if no search terms have been provided, the list contains all items in the repository). Each item is described by its title and some short descriptive details.* 4.2 The system also presents a *list of "facets" with indication of available "hits" (number of items) related to each facet.* *4.3 The system presents a "pager" indicating the total number of items and the number of pages that hold these items.* 5a. The user may select an item to view its detailed description (UC a-04). 5b. By pressing on any facet, the user can filter the search results accordingly. 5c. The user may use the "pager" to move to other pages of the results list.
Alternative flows:	6a. The user has the option to "refine" the search by altering the terms in the search field. 6b. The user can use the system's "search notation" to provide advanced search criteria, i.e. specify boolean criteria (and, or, not), search in specific fields (title, keyword, etc.), search for parts of a word, etc. 7a. The user is able to "store" the search, by copying the URL of the web browser. The full state of the search is stored, including any used facets and the current page of the search results list. 7b. Alternatively to 7a, the system may continuously update the search terms in the search field, according to the user's choices of facets, page, etc., so that the user is able to copy the content of the search fields in order to "store" the search.
Notes:	A next release of the system (Version 2) may include an Advanced Search form to allow for easy formulation of complex search queries.

5.2.1.2 UC a-02: Browse RULOs

Summary:	Users shall be able to browse for RULOs, by choosing specific levels and items of categorisation.
Release:	Version 1
Actors involved:	All users (humans)
Preconditions:	-

Basic flow:	1. The user visits the repository in the web browser, through its URL or a link from another website. 2. The user selects to browse using one (of possible more) method of categorisation, e.g. browse by topic (taxonomy-based), browse by provider (collection-based), browse by country or language, etc. *3. The system displays the list of available elements, according to the browse method, hinting on the number of items related to each element.* 4. The user selects one of the elements. 5.1 As soon as the user has selected a specific browsing element the system presents the related items, in the form of an ordered list. Each item is described by its title and some short descriptive details. 5.2 Additionally, any available browsing sub-elements are presented using a hierarchical ("tree-like") control of the user interface, with an indication of available number of items related to each sub-element. 5.3 *The system presents a "pager" indicating the total number of items and the number of pages that hold these items.* 6a. The user may select an item to view its detailed description (UC a-04). 6b. Alternatively the user may opt to browse through the sub-elements of the selected browsing method. 6c. The user may use the "pager" to move to other pages of the list of items of the selected browsing element.
Alternative flows:	7. The user is able to "store" the browsing status, by copying the URL of the web browser. The full state is stored, including any selected (sub-) elements and the current page of the results list.
Notes:	-

5.2.1.3 UC a-03: Locate RULO by Persistent Identifier

Summary:	Users shall be able to provide a Persistent Identifier that uniquely relates to a specific item of the repository.
Release:	Version 1
Actors involved:	All users (humans)
Preconditions:	-
Basic flow:	1. The user visits the repository in the web browser, through its URL or a link from another website. 2a. The user provides the Persistent Identifier (a mix of letters and numbers) in the Search field. 2b. Alternatively, the user provides the Persistent Identifier directly in the

	URL, using a specific encoding scheme. *3. The system verifies the validity of the Identifier and redirects the user to the page of the associated item (UC a-04).*
Alternative flows:	*3b. If the identifier is invalid, a descriptive message is presented to the user.* 1a. (version 2) The user can use a "short url" to identify a specific item. 1b. (version 2) The user can use a QR-code[5] to identify a specific item, possibly scanned by the camera of a smart device (phone, tablet, etc.), thereby obviating the need for the user to type the URL into the web browser.
Notes:	-

5.2.1.4 UC a-04: View RULO

Summary:	Users shall be able to view a web page with the detailed description of a RULO.
Release:	Version 1
Actors involved:	All users (humans)
Preconditions:	The user has identified a unit and opts to view its detailed description (through use cases UC a-01, UC a-02 or UC a-03).
Basic flow:	1. The browser of the user requests page with the description of a unit. *2.1 The system displays the description of the unit, including all the available information.* *2.2 A link is provided for any information which can lead to related units, including keywords, category, provider, etc. Each link redirects the browser of the user to a relevant page for searching (UC a-01) using pre-defined criteria (e.g. keyword), or browsing (UC a-02) related items (e.g. a specific category).*

[5] **QR code** (abbreviated from Quick Response **Code**) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached.

Alternative flows:	3a. The user may store the location of the unit for future reference by copying the URL, which should be encoded as a "Semantic URL[6]" (in a clean, human-readable fashion). 3b. Alternatively the user can choose to save a short version of URL referring to the page item's description page (short URL). 3c. Alternatively the user can choose to store the associated QR-code, which encodes the URL of the description page. 3d. Alternatively the user can choose to store the Persistent Identifier of the unit. *4a. If the requested unit has been withdrawn for some reason, an informational message is presented to the user, possibly providing details about the status of the unit (date of withdrawal, links to relevant units, etc.)* *4b. If a newer version of the requested unit exists, the system notifies the user accordingly.*
Notes:	Future versions of the system may allow for easy embedding details of the displayed item in other systems (similar to the notion of "tweeting" a news item).

5.2.1.5 UC a-05: Provide Feedback

Summary:	Users shall be able to provide feedback at any stage of their interaction with the system
Release:	Version 1
Actors involved:	All users (humans)
Preconditions:	-
Basic flow:	1. The user of the repository chooses to provide feedback, by pressing the associated button which is readily available to a specific and apparent location at every page of the repository. *2. The system provides a short input form for capturing the user's feedback.* 3. The user provides the feedback, along with some personal details for possible future contact. *4. The system stores the feedback and notifies the user accordingly.*

[6] Semantic URLs, also sometimes referred to as clean URLs, RESTful URLs, user-friendly URLs, or SEO-friendly URLs, are URLs intended to improve the usability and accessibility of a website or web service by being immediately and intuitively meaningful to non-expert users (http://en.wikipedia.org/wiki/Semantic_URL).

Alternative flows:	*2a. If the user is anonymous (i.e. not registered), the system presents a “challenge” in order to eliminate the possibiblity of non-human (robot-based) or spam feedback.* 3a. If the user is registered with the repository, the feedback form is pre-populated with some of the personal details from the user profile (name, e-mail).
Notes:	-

5.2.1.6 UC a-06: Register (Create account)

Summary:	Any user may opt to register with the repository, i.e. create a new account in order to be able to edit and submit new items to the repository.
Release:	Version 1
Actors involved:	All users (humans)
Preconditions:	-
Basic flow:	1. The user selects the 'Register' link. *2. The system presents a User registration form, asking for a minimum set of mandatory details, along with some optional information for the user’s personal profile:* • *e-mail* *(mandatory)* • password *(mandatory)* • *name* *(mandatory)* • *organisation* • *role* • *age* • *country* • *postal code* 3. The user fills the registration form and submits it to the system. *4. The system validates the user information and sends a verification e-mail to the user, with a hashed one-time-only link (URL) that must be used in order to validate the validity of the provided e-mail address.* 5. The user receives the verification e-mail and uses the link in order to certify the property of this e-mail. *6. The system creates a new user account and sends a confirmation e-mail with the account details to the user.*
Alternative flows:	4a. A set of checks ensures that all information is valid in terms of its expected type. In case of error, the user is notified accordingly and asked to provide valid information.

	4b. A special check is used in order to assure that the password provides enough security. 4c. In the special case where an e-mail has already been used, the user is notified accordingly along with a link for the password reset service (UC a-07).
Notes:	The possibility of making use of identification schemes based on OpenID should be investigated. This will allow users to be authenticated by certain co-operating sites or well-known social services (Twitter, LinkedIn, Google, Facebook, etc.)

5.2.1.7 UC a-07: Reset password

Summary:	Registered users may need to reset their password.
Release:	Version 1
Actors involved:	Registered user
Preconditions:	-
Basic flow:	1. The user selects the 'Reset Password' link. *2. The system asks for the e-mail address that has been used for the creation of the account.* 3. The user provides the e-mail address. *4. The system validates the provided information and sends an e-mail to the user, with a hashed one-time-only link (URL) that must be used in order to reset the password.* 5. The user receives the e-mail and uses the link in order to reset the password. *6. The system validates the link and presents a form where the user may enter a new password.* 7. The user enters a new password. *8. The user's profile is updated with the new password.*
Alternative flows:	4a. In the special case where the e-mail is not found in the system's database, the user is notified accordingly and a suggestion (link) to create a new account (UC a-06) is made. 8a. A special check is used in order to assure that the password provides enough security.
Notes:	-

5.2.1.8 UC a-08: Login

Summary:	Registered users can login to the system with their credentials
Release:	Version 1
Actors involved:	Registered users
Preconditions:	-
Basic flow:	1. The user selects the ‘Login’ link. *2. The system redirects to login page that is served through an encrypted HTTP channel (https) and asks for the e-mail address that has been used for the creation of the account along with the user’s password.* 3. The user provides the e-mail address and password. *4. The system validates the provided information creates a new session for the user.*
Alternative flows:	1a. Alternatively, the user selects to visit a page (directly entering a URL in the browser), which is not available for anonymous users. 2a. In this case the system provides a notification and redirects the user to the login page (Step 2). 3a. The user can select to “Stay connected”, in which case subsequent login processes are made automatically based on a cookie that is saved in the user’s browser (local storage). 4a. In case of invalid combination of email and password, the user is notified accordingly. After 3 invalid attempts, the user is presented with an extra challenge, in the form of CAPTCHA test (*"Completely Automated Public Turing test to tell Computers and Humans Apart").*
Notes:	See UC a-06, regarding the possibility of using an OpenID service of a trusted authority in order to login to the system.

5.2.1.9 UC a-09: Logout

Summary:	A user who has logged in the system can opt to logout in order to terminate the authenticated session.
Release:	Version 1
Actors involved:	Registered users
Preconditions:	The user should have logged in the system before opting to logout.
Basic flow:	1. The user selects the ‘Logout’ link. *2. The system presents a confirmation message.* 3. The user confirms the intention to logout.

	4. The system logs out the user (i.e. it terminates the session).
Alternative flows:	1a. After a specified amount of inactivity time the user’s session is automatically terminated and the user is considered to have logged out of the system, except if the user has opted to stay connected (UC a-08). 1b. After closing the user’s browser, the user’s session is automatically terminated and the user is considered to have logged out of the system, except if the user has opted to stay connected (UC a-08).
Notes:	-

5.2.1.10 UC a-10: Create a RULO

Summary:	A registered user can submit the definition of a new RULO to the repository.
Release:	Version 1
Actors involved:	Editors
Preconditions:	-
Basic flow:	1. The user selects the ‘Create new item’ link. *2. The system presents an input form, asking for a minimum set of mandatory details, along with some optional description for the item.* 3. The user fills the form and submits it to the system. *4. The system validates the submitted information and presents a summary asking for a confirmation from the user.* 5. The user confirms the validity of the information that is presented in the summary and selects the “Create item” link. *6. The system creates the new item and sends a confirmation e-mail with the item’s details to the user.*
Alternative flows:	4a. A set of real-time checks ensures that all information is valid in terms of its expected type. In case of error, the user is notified on-the-fly, with an indication next to the user interface control related to the erroneous information.
Notes:	Next versions of the system may allow for storing draft versions of the RULOs, i.e. unpublished information that the user may opt to complete and properly submit in a multi-staged manner. Moreover, the option for importing definitions of units of learning outcomes shall be investigated.

5.2.2 2nd Release

The second release of the system will focus mainly on the management of updates of items of the repository. To this end, it shall facilitate the tasks of users with advanced access rights that undertake the role of the **Moderator** of the system's content. Moreover, the new functionality that will be integrated in the 2nd release of the system shall allow the linking of each unit of RULO to other related units, which will allow for the creation of flat and hierarchical structures of inter-connected (units of) RULOs.

5.2.2.1 UC b-01: Relate a RULO resource to other resources

Summary:	An editor of a RULO shall be able to define relations to other RULOs.
Release:	Version 2
Actors involved:	Editors
Preconditions:	The user should have already created the RULO.
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project's lifetime, based on feedback from the usage of the 1st release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following tasks*: 1. A registered user can choose to "relate" an existing unit that he has published to one or more other RULOs 2. The system shall provide a mechanism for selecting other RULOs (at least a text field that shall allow for the input of a valid unique persistent identifier). 3. The system will also allow the selection of the "type" of relation from a predefined list of options (a closed vocabulary of relations). 4. Furthermore, the system shall allow the creation of "hierarchies" of units of learning outcomes, i.e. by special "parent-child" relationships.
Alternative flows:	5. The user shall be able to delete or otherwise change an existing relation (Error: Reference source not found).
Notes:	We need to aggree on the expected behaviour when a unit (u1) is linked to another unit (u2): shall the relation be evident in the description of u2? For the time being we suggest that u2 is updated only if the Moderator reviews and accepts that the relation is valid and should be visible in the descriptions of both units.

5.2.2.2 UC b-02: View related RULOs

Summary:	The description of a RULO shall include information for its relation with other RULOs.
Release:	Version 2
Actors involved:	All users (humans)
Preconditions:	Error: Reference source not found
Basic flow:	Extending *UC a-04: View RULO*: *2.2 A list of related RULOs is displayed, in the form of hyperlinked titles and other possible useful information (type of relation, date of relation, etc.)*
Alternative flows:	(not defined at this stage)
Notes:	▪ See notes in *UC a-04: View RULO*, regarding the “visibility” of a relationship on both related units of learning outcomes. ▪ An interesting functionality related to this use case is the ability to visualise relations in a graph-like manner.

5.2.2.3 UC b-03: Revise a RULO

Summary:	A user shall be able to revise the description of a RULO, provided the appropriate access rights.
Release:	Version 2
Actors involved:	Editors
Preconditions:	The user should have already created the RULO.
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project’s lifetime, based on feedback from the usage of the 1st release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following*: 1. A registered user can choose to “revise” an existing unit. 2. The system shall check whether the user has appropriate access rights and then reuse the form of Error: Reference source not found in order to allow the user to provide an updated description 3. The system should also ask for a short description of the update (revision log).
Alternative flows:	(not defined at this stage)
Notes:	We may need to implement a functionality that shall allow users to easily view the units that are under their “control”. Alternatively, this could be part of an “advanced” search filter, which could be called directly from a link on each registered user’s actions menu.

5.2.2.4 UC b-04: Manage updates

Summary:	Moderators shall be facilitated in their daily tasks of monitoring updates to the content of the repository.
Release:	Version 2
Actors involved:	Moderators
Preconditions:	*UC a-08: Login*
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project's lifetime, based on feedback from the usage of the 1st release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following tasks*: 1. Moderators shall have access to an "administration dashboard" with all recent changes to the content of the repository (i.e. new & updated RULOs, relations, etc.) 2. This information shall be displayed in the form of an intuitive list of items, which will be sortable, filterable and configurable to meet the needs of the moderator. 3. Through this list the moderator shall be able to follow the links for further actions.
Alternative flows:	(not defined at this stage)
Notes:	-

5.2.2.5 UC b-05: Moderate a RULO

Summary:	Moderators shall be able to review the description of a RULO and update it if needed.
Release:	Version 2
Actors involved:	Moderators
Preconditions:	*UC a-08: Login*
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project's lifetime, based on feedback from the usage of the 1st release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following tasks*: 1. Moderators can choose to edit an existing RULO. 2. The system shall reuse the functionality of *UC a-10: Create a RULO*. 3. A notification shall be sent with e-mail to the editor of the RULO.
Alternative flows:	(not defined at this stage)
Notes:	-

5.2.2.6 UC b-06: Moderate a relation

Summary:	Moderators shall be able to review the relation of a RULO with other RULOs.
Release:	Version 2
Actors involved:	Moderators
Preconditions:	*UC a-08: Login*
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project's lifetime, based on feedback from the usage of the 1^{st} release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following tasks*: 1a. Moderators can choose change the type of a relation. 1b. Moderators can choose to delete a relation. 1c. Moderators can choose to allow a relation to be visible on both ends. 1d. Moderators can choose to specify the type of the reverse direction of a relation. 2. A notification shall be sent with e-mail to the editors of both RULOs.
Alternative flows:	(not defined at this stage)
Notes:	-

5.2.2.7 UC b-07: Review feedback

Summary:	The Moderator's dashboard shall hold a list of all feedback by end-users of the repository.
Release:	Version 2
Actors involved:	Moderators
Preconditions:	*UC a-08: Login*
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project's lifetime, based on feedback from the usage of the 1^{st} release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following tasks*: 1. The system displays a list of all feedback by end-users that is related to specific content of the repository, in the form of a sorted table (date, username, title of related RULO, indication of status, etc.) 2. The moderator can act upon an item in the feedback-list (i.e. correct some description). 3. A notification shall be sent with e-mail to both the editor of the

	affected RULO and the user who has provided the feedback. 4. The status of the feedback item shall be changed (automatically or directly from the moderator) in order to indicate the change (i.e. resolved, closed, rejected, waiting for, etc.)
Alternative flows:	(not defined at this stage)
Notes:	-

5.2.2.8 UC b-08: Notify Editor

Summary:	Moderators shall be able to notify Editors of RULOs
Release:	Version 2
Actors involved:	Moderators
Preconditions:	*UC a-08: Login*
Basic flow:	*The detailed description for the basic flow of this use case will be provided later on during the project's lifetime, based on feedback from the usage of the 1st release and better insight of users needs and expected functionality. However, the system is expected to provide functionality for the following tasks*: 1. A Moderator shall be able to notify Editors for some reason related to the content that they have created in the repository. 2. The system shall display an input form, along with any contact details of the Editor. 3. The Moderator shall be able to use the form in order to provide their message to the Editor 4. The system shall include a Notification event in the Editor's user interface.
Alternative flows:	3a. The Moderator could choose to use e-mail for messaging the Editor.
Notes:	-

5.2.3 3rd Release

The third release of the system will focus mainly on its relation with other systems.

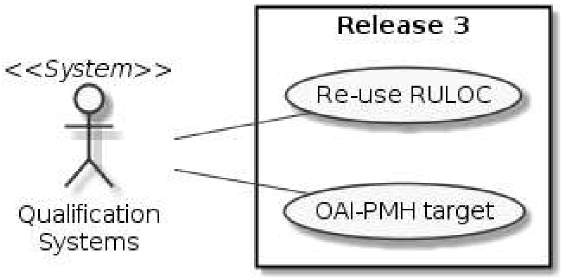


Figure 8: Use case model (Release 3)

5.3 Mockups and Design patterns

5.3.1 Pager

- http://ui-patterns.com/patterns/Pagination
- http://www.flickr.com/photos/morville/collections/72157603790650061/

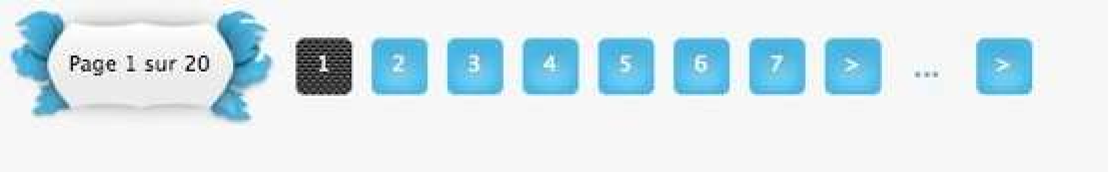
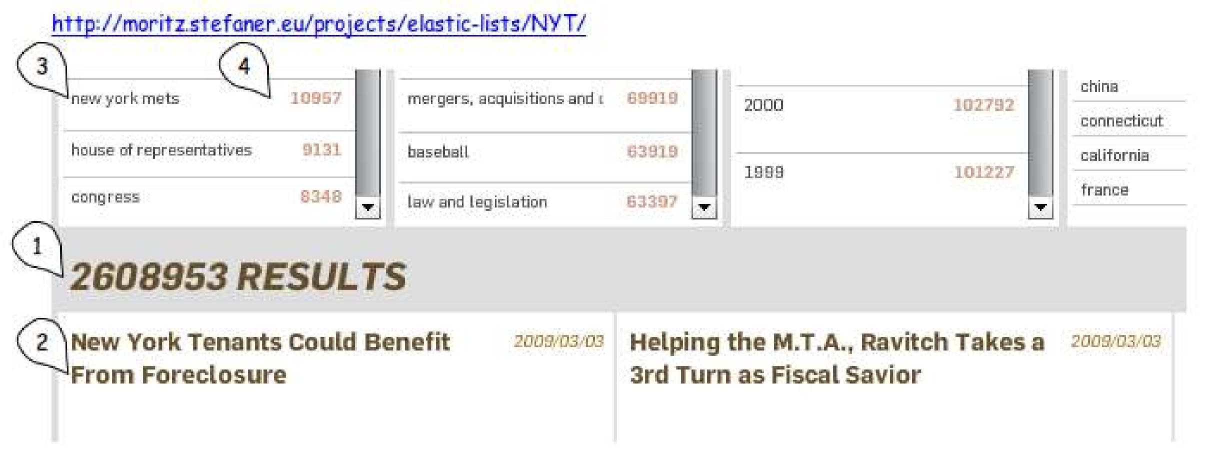


Figure 9: A simple pager

5.3.2 Faceted search


http://moritz.stefaner.eu/projects/elastic-lists/NYT/

new york mets 10957
house of representatives 9131
congress 8348

mergers, acquisitions and (69919
baseball 63919
law and legislation 63397

2000 102792
1999 101227

china
connecticut
california
france

2608953 RESULTS

New York Tenants Could Benefit From Foreclosure 2009/03/03

Helping the M.T.A., Ravitch Takes a 3rd Turn as Fiscal Savior 2009/03/03

Figure 10: Faceted search with filters

The image above presents the main user interface elements of faceted search with filters[7]:

- A total for all results is displayed [1] along with choices for filtering based on pre-defined categories [2] or facets [3].
- The sub-totals of results for each facet are also displayed [4].
- All filters are inclusive ("AND" logic / drill-down)

5.3.3 Behavioural patterns

The suggested approach for locating items (units of learning outcomes) in the repository, closely matches what is referred to as **"Exploratory Search"**. Quoting Wikipedia[8]:

> Exploratory search is a specialization of information exploration which represents the activities carried out by searchers who are either:
>
> a) unfamiliar with the domain of their goal (i.e. need to learn about the topic in order to understand how to achieve their goal)
>
> b) unsure about the ways to achieve their goals (either the technology or the process)
>
> c) or even unsure about their goals in the first place.

According to the documentation of DSpace[9]:

[7] http://www.uxmatters.com/mt/archives/2009/09/best-practices-for-designing-faceted-search-filters.php

[8] http://en.wikipedia.org/wiki/Exploratory_search

> In a standard search operation, a user specifies his complete query prior to launching the operation. If the results are not satisfactory, the user starts over again with a (slightly) altered query.
>
> In a faceted search, a user can modify the list of displayed search results by specifying additional "filters" that will be applied on the list of search results. In DSpace, a filter is a contain condition applied to specific facets.
>
> ...
>
> Using the standard search, a user would search for something like [wetland + "dc.author=Mitsch, William J" + dc.subject="water quality"]. With filtered search, they can start by searching for [wetland], and then filter the results by the other attributes, author and subject.

The following figures provide an overview of different approaches for searching and locating items in an online repository.

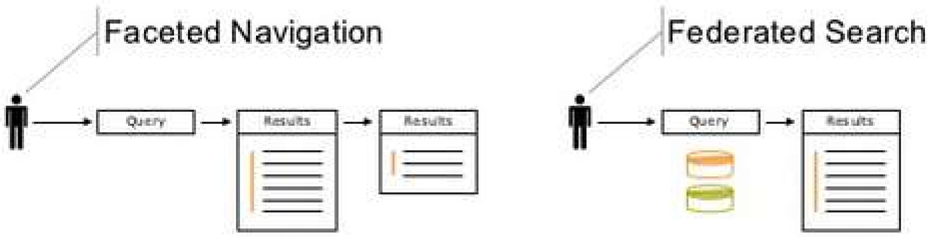


Figure 11: Faceted navigation (left) and Federated search (right)

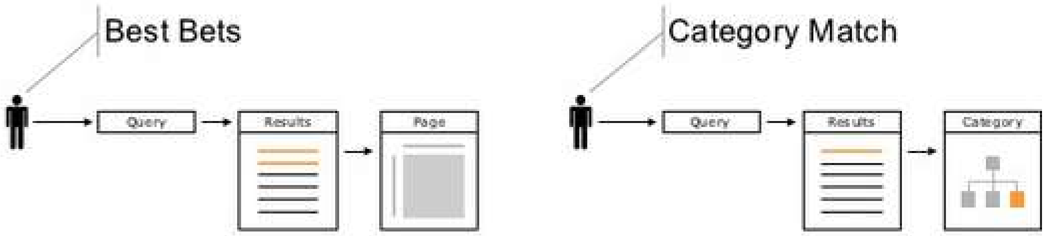


Figure 12: Searching by best bets (left) and Category match (right)

[9] https://wiki.duraspace.org/display/DSDOC3x/Discovery#Discovery-WhatisaSidebarFacet

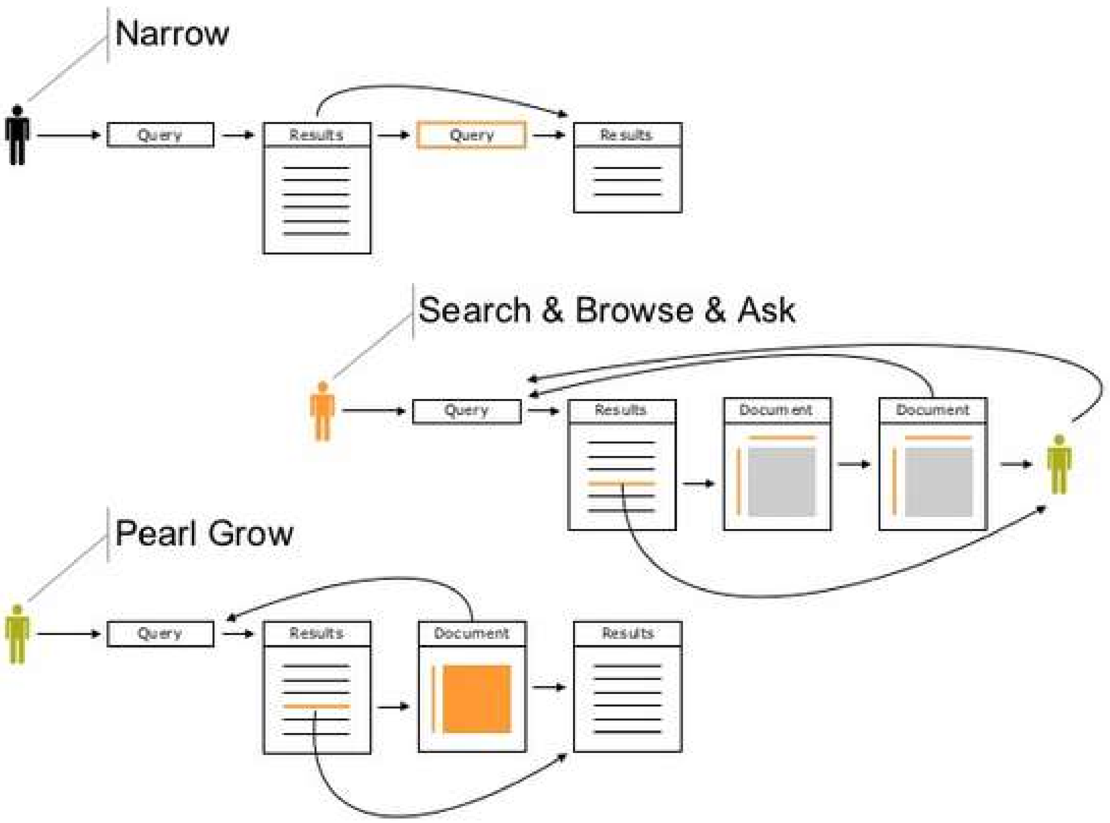


Figure 13: Searching by Narrowing results (top), Interactively (middle) and by "Pearl grow" (bottom)

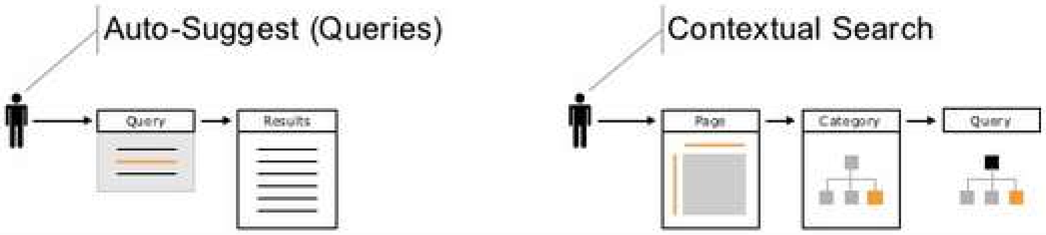


Figure 14: Searching by automatic suggestions (left) and by Context (right)

5.4 Non-functional requirements

The basic non-functional requirements are presented in the following table:

ID	Name	Comment
NF.01	Multi-lingual	The user can change the language of the interface.
NF.02	Usability	Design the user with usability in mind – users computer literacy can

		vary a lot, given the fact that the portal's topics are of global interest.
NF.03	Efficiency	The system should not exceed specific resource consumption for given load.
NF.04	Performance	Search results should appear timely and overall system's response time should be rated as "fast".
NF.05	Availability	The portal should have an 24x7 availability with minimum down-time and scheduled maintenance.
NF.06	Certification	The system should not include or make references to inappropriate content.
NF.07	Open Source	System design and coding should promote open source.
NF.08	Extensibility	Design should allow for development of extensions, plugins and connectors.
NF.09	Compatibility	The system should adopt standards and employ protocols for communicating with related external systems, tools and services.
NF.10	Scalability	The system should scale well, being able to maintain or even increase its level of performance with the addition of server resources (CPU power, memory, bandwidth, etc.)

6 SYSTEM ARCHITECTURE

The systems shall be deployed on cloud infrastructures in order to achieve efficient and effective operation at a minimum cost. The initial planning for the repository of RULOs has been to be based on Ariadne infrastructure components[10] and, more specifically, on:

i. the validation service that allows the validation of metadata instances against predefined application profiles;

ii. the registry service that allows the creation and management of catalogues/indexes with the providers of metadata records/instances;

iii. the harvester service that allows the periodical monitoring, fetching, aggregation and updating of metadata records from various distributed providers; and

iv. the repository service that allows the creation of a repositories that store the aggregated metadata records and facilitate the discovery, searching and retrieval of metadata records, both through an interactive, online interface at the user browser as well as through web-services that other online systems can use to directly integrate resources and metadata.

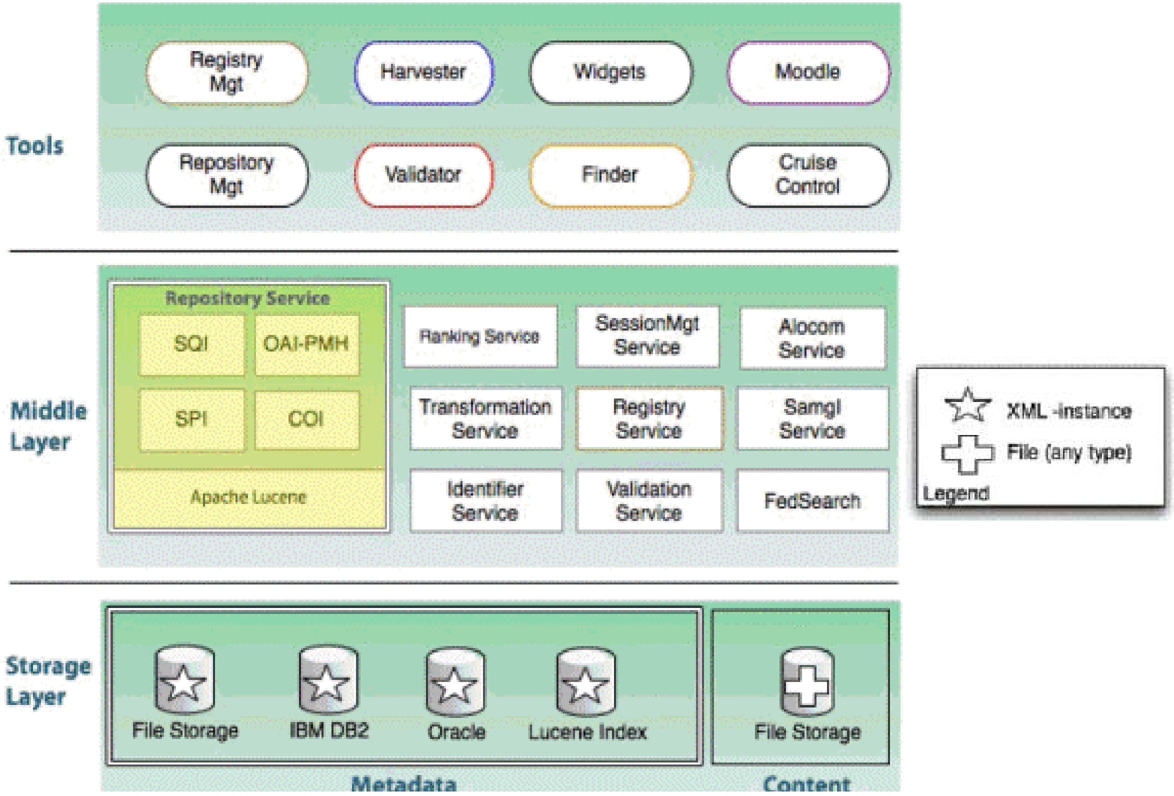
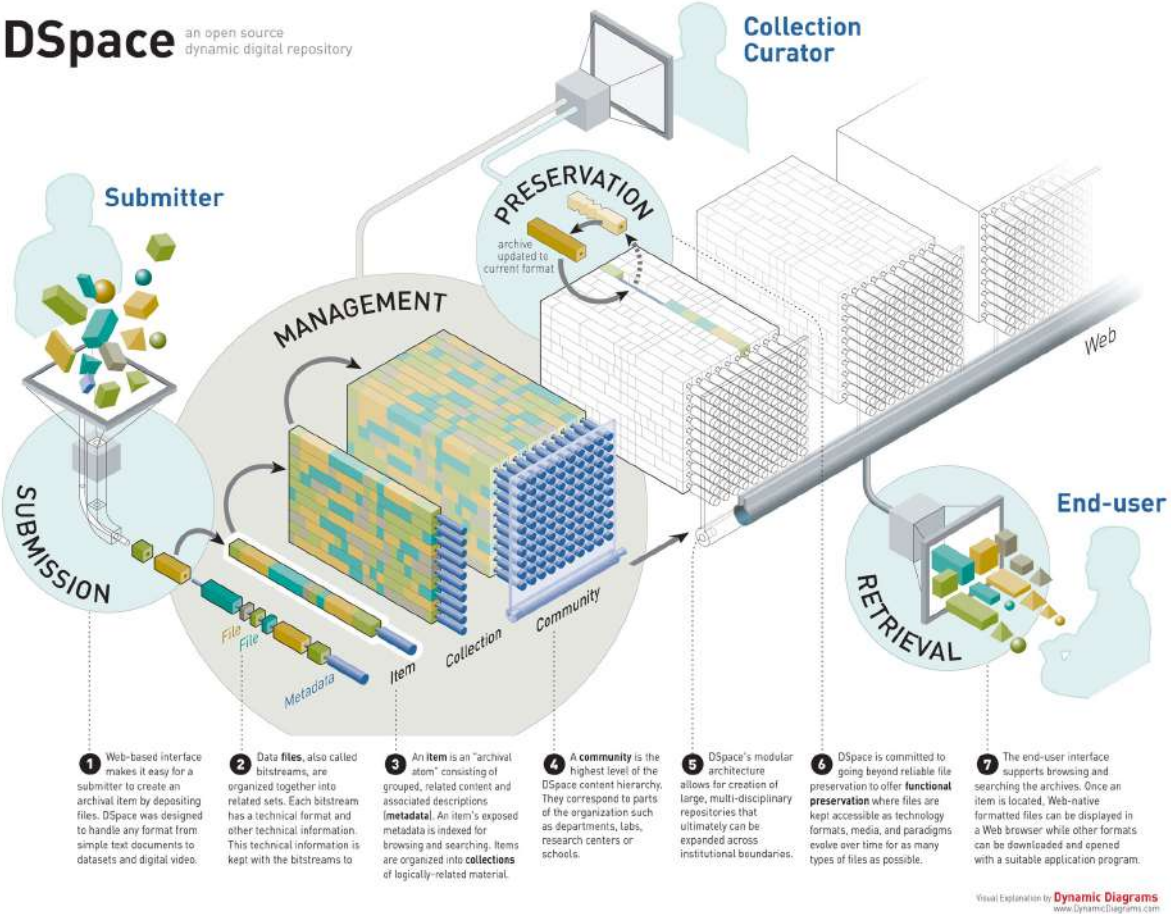


Figure 15: The ARIANDE infrastructure components

[10] http://www.ariadne-eu.org/

The updated design of the system is based on the usage of the DSpace open source software providing a turnkey repository application used by more than 1000+ organizations and institutions worldwide[11]. This software shall be configured and programmatically adapted in order to meet the specified requirements .The Ariadne infrastructure components shall thus be used alongside the DSpace installation, in order to reach an integrated set of services and functionality.



1 Web-based interface makes it easy for a submitter to create an archival item by depositing files. DSpace was designed to handle any format from simple text documents to datasets and digital video.

2 Data **files**, also called bitstreams, are organized together into related sets. Each bitstream has a technical format and other technical information. This technical information is kept with the bitstreams to

3 An **item** is an "archival atom" consisting of grouped, related content and associated descriptions (**metadata**). An item's exposed metadata is indexed for browsing and searching. Items are organized into **collections** of logically-related material.

4 A **community** is the highest level of the DSpace content hierarchy. They correspond to parts of the organization such as departments, labs, research centers or schools.

5 DSpace's modular architecture allows for creation of large, multi-disciplinary repositories that ultimately can be expanded across institutional boundaries.

6 DSpace is committed to going beyond reliable file preservation to offer **functional preservation** where files are kept accessible as technology formats, media, and paradigms evolve over time for as many types of files as possible.

7 The end-user interface supports browsing and searching the archives. Once an item is located, Web-native formatted files can be displayed in a Web browser while other formats can be downloaded and opened with a suitable application program.

Visual Explanation by Dynamic Diagrams
www.DynamicDiagrams.com

Figure 16: The DSpace architecture and workflow

In terms of interoperability, the repository shall allow for user registration using existing user accounts at OpenId providers. This will allow users with accounts in online services such as Google, Yahoo, Facebook, Flickr, Blogger.com, Wordpress.com, to easily create their account with each of the online services.

Furthermore, the repository of RULOs shall investigate the usage of the EPIC persistent identifiers scheme. EPIC, the European Persistent Identifiers Consortium, was founded in 2009 by a consortium of European partners in order to provide PID services for the European Research Community, based on the handle system[12], for the allocation and resolution of

[11] http://www.dspace.org/

persistent identifiers. The consortium signed a Memorandum of Understanding aiming to provide long term reliability for the PID services[13].

6.1 Deployment

The following diagram presents the schematic layout of the virtual machines (VMs) that shall be used for the deployment of the COMPASS online services. This deployment layout, or systems' architecture, has been designed in a way that allows for easy scaling in order to to handle increased or decreased usage demands. In fact, the systems' architecture accommodates both types of scaling: (a) Horizontal Scaling – scaling out and scaling in of VMs that are of the same type; and (b) Vertical Scaling – scaling up and scaling down.

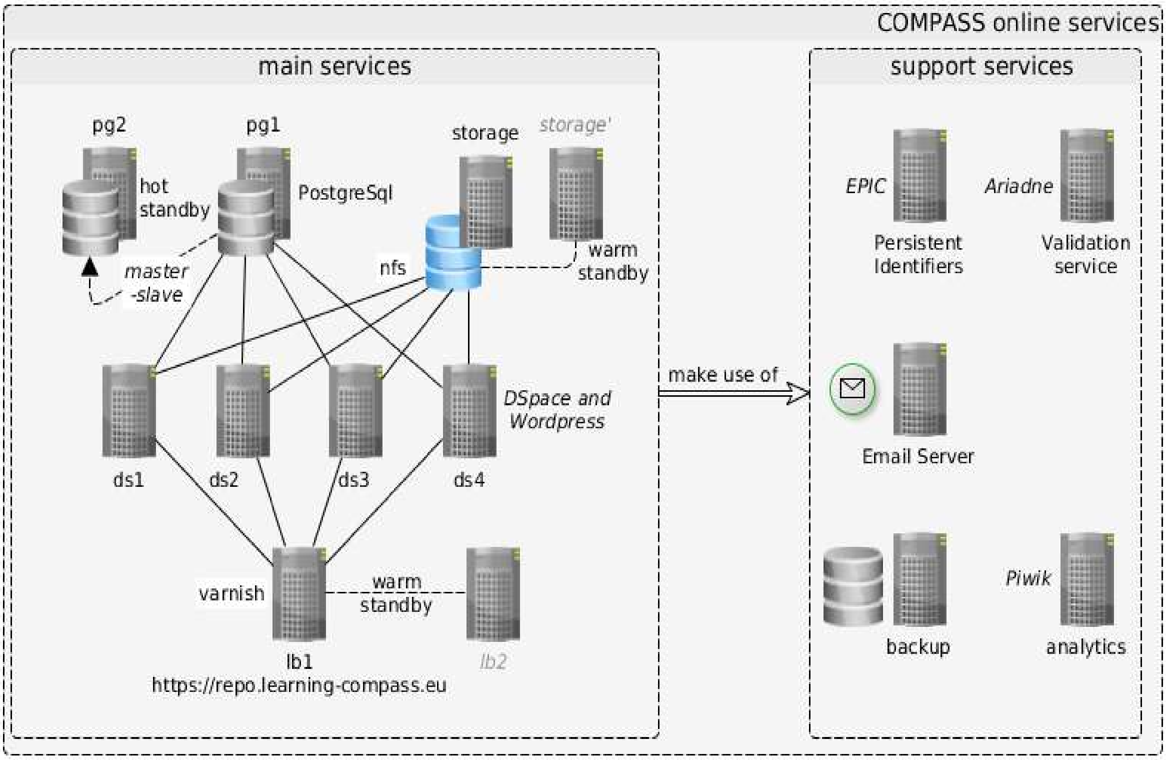


Figure 17: Virtual Machines of the COMPASS online services

The schematic layout of the systems' architecture diagram holds 15 VMs, an indicative number that could be lower of bigger, according to the user demand and performance requirements of the services. If, for example, an increase in user demand is noticed, the VMs that hold the dspace and wordpress installations can be increased (horizontal scaling from four to six or more). In another example, the VM that provides the central storage for the infrastructure, could be

[12] http://www.handle.net/

[13] http://www.pidconsortium.eu/

replaced with one of larger hard disk capacity and / or RAM memory in order to accommodate for greater demand of resources (vertical scaling).

7 REFERENCES

[1] Adolph, Bramble, Cockburn, and Pols: Patterns for Effective Use Cases, Addison Wesley, 2003.

[2] Adolph, Bramble, Cockburn, and Pols: Patterns for Writing Effective Use Cases, presented at the 19th Annual Conference on Object-Oriented Programming, Systems, Languages and Applications, 2004.

[3] Ambler, S.W.: The Object Primer 3rd Edition: Agile Model Driven Development with UML 2. Cambridge University Press, 2004

[4] Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide, Addison-Wesley Professional, 2005

[5] Fowler, Martin.: UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley. Published 2003.

[6] Tom Gilb: Principles of Software Engineering Management. Harlow, England: Addison Wesley Longman, 1988.

[7] Hull, E., Jackson, K. and Dick, J.: Requirements Engineering, Second Edition. Springer, 2005

[8] Jacobson , I., Spence , I., Bittner, K.: USE-CASE 2.0 The Definitive Guide, Ivar Jacobson Intl., 2011.

[9] Kroll, P. and Kruchten, P.: The Rational Unified Process Made Easy, Addison Wesley, 2003.

[10] Kroll, P. and MacIsaac, B.: Agility and Discipline Made Easy, Addison Wesley, 2005.

[11] Kruchten, P.: From Waterfall to Iterative Development - A Challenging Transition for Project Managers, The Rational Edge e-zine, 2000. available online (Jun 2014): http://download.boulder.ibm.com/ibmdl/pub/software/dw/rationaledge/dec00/FromWaterfalltoIterativeDevelopmentDec00.pdf

[12] Kruchten, Phillipe B.: The 4+1 View Model of Architecture, IEEE Software, vol. 12, no. 6, pp 42-50, Nov. 1995

[13] Object Management Group: Software Process Engineering Meta-Model, Version 2.0, 2008.

8 ABOUT COMPASS

COMPASS aims at supporting HEIs to implement the necessary reforms in line with the 2011 EU Modernisation Agenda and its priority areas through the improvement of quality and relevance of offered learning opportunities to current and emerging labour market needs, the enhancement of mobility, making LOs more visible and understandable for students that want to gain additional skills the strengthening of cross-border cooperation of HEIs in the definition of quality flexible learning pathways for their learners, the increase of social responsibility of HEIs through the transparent descriptions of their offerings, the implementation of sustainable infrastructure for all European HEIs and for the EU to leverage in the enhancement of existing or the creation of new related services.

COMPASS will work to:

- *promote the concept of flexible lifelong learning pathways within HEI providers and learners through the use of innovative guidance instruments;*
- *produce policy suggestions for fomenting their implementation by higher education institutions;*
- *recommend standard information models for enabling Learning Opportunity (LO) providers to engage in the well-structured description of their offers (based on the MLO standard and extending it with learning outcomes and competence information)*
- *develop online services comprising a software component for LO description, to be made freely and publicly available to LO providers, and a free tool for lifelong learners to leverage LO information for building desired pathways for personal development;*
- *leverage the consortium span to policy makers, HEIs, industry as well as its ICT-sector orientation, to design and implementation-wide pilot tests that will boost awareness, institutional commitment and national impact.*

The produced models and technology can be exploited by national and European services (e.g. Ploteus), making an impact on the quantity and quality of LO information at both national and European level. For this purpose a technical recommendation for ingesting LO descriptions into Ploteus, will be produced.