

# ICT ROMULUS

## REPORT

### D6.1 - Report on Rationale and Requirements of the demonstrator

---



**ROMULUS**

|                     |  |
|---------------------|--|
| <b>Editor</b>       | GESFOR   |
| <b>Contributors</b> | All partners                                   |
| <b>Reviewers</b>    | LIFERAY, DERI                                  |
| <b>Work Package</b> | WP6 – Vertical Demonstrators                   |
| <b>Task</b>         | T6.1 – Demonstrator requirements and rationale |
| <b>Date</b>         | 30/01/09                                       |
| <b>Release</b>      | 1.0  |
| <b>Status</b>       | Final  |
| <b>Distribution</b> | Confidential                                   |

## HISTORY

| Release | Date       | Remarks                                      | Status | Distribution |
|---------|------------|--|--------|--------------|
| 0.1     | 14-01-2009 | First draft version                          | Draft  | Confidential |
| 0.2     | 26-01-2009 | Filled the Musiteca and Scrooge description. | Draft  | Confidential |
| 0.3     | 27-01-2009 | Filled the conclusion                        | Draft  | Confidential |
| 0.4     | 29-01-2009 | Filled Cornelius project                     | Draft  | Confidential |
| 0.5     | 29-01-2009 | Liferay contribution                         | Draft  | Confidential |
| 0.6     | 30-01-2009 | DERI added contribution                      | Draft  | Confidential |
| 0.7     | 30-01-2009 | DERI revision                                | Draft  | Confidential |
| 0.8     | 30-01-2009 | Liferay revision                             | Draft  | Confidential |
| 1.0     | 30-01-2009 | Final version                                | Draft  | Public       |



## TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>1 INTRODUCTION.....</b>   | <b>4</b>  |
| <b>2 CRITERIA TO SELECT THE DEMONSTRATORS.....</b>                             | <b>4</b>  |
| <b>3 ROMULUS RELEASE PACKAGING.....</b>  | <b>4</b>  |
| 3.1 JavaTeam.....  | 5         |
| <b>4 VERTICAL DEMONSTRATORS.....</b>   | <b>5</b>  |
| 4.1 Musiteca.....  | 5         |
| 4.1.1 <i>Introduction</i> .....  | 5         |
| 4.1.2 <i>Romulus modules</i> .....   | 6         |
| 4.1.2.1 WP3 – ROMA Metaframework Architecture and Development Tools.....       | 7         |
| 4.1.2.2 WP4 – ROMULUS Data and Services Mashup for Enterprise Integration..... | 7         |
| 4.1.2.3 WP5 – Rich Client Web Technologies, security and reliability.....      | 8         |
| 4.2 Cornelius.....   | 8         |
| 4.2.1 <i>Introduction</i> .....  | 8         |
| 4.2.2 <i>Romulus modules</i> .....   | 8         |
| 4.2.2.1 WP3 – ROMA Metaframework Architecture and Development Tools.....       | 9         |
| 4.2.2.2 WP4 – ROMULUS Data and Services Mashup for Enterprise Integration..... | 9         |
| 4.2.2.3 WP5 – Rich Client Web Technologies, security and reliability.....      | 10        |
| 4.3 Scrooge.....   | 10        |
| 4.3.1 <i>Introduction</i> .....  | 10        |
| 4.3.2 <i>Romulus modules</i> .....   | 10        |
| 4.3.2.1 WP3 – ROMA Metaframework Architecture and Development Tools.....       | 11        |
| 4.3.2.2 WP4 – ROMULUS Data and Services Mashup for Enterprise Integration..... | 11        |
| 4.3.2.3 WP5 – Rich Client Web Technologies, security and reliability.....      | 11        |
| <b>5 CONCLUSIONS.....</b>  | <b>12</b> |

## 1 Introduction

This document describes the requirements and rationale of the vertical demonstrators of Romulus technologies. With the development of demonstrators that use the technologies provided by the Romulus project, it is intended to check its applicability in real contexts. The selection of the demonstrators has been performed taking into account the capabilities of Romulus technologies.

## 2 Criteria to select the demonstrators

The main criteria to define the demonstrators are:

- Completeness. Demonstrators should allow to evaluate the largest number of Romulus functionalities.
- Global vision. Although the demonstrators can be focused in any of the Romulus functionalities, the integration of the different components should be evaluated with demonstrators too, so that it would offer a global vision.
- Testing approach. Demonstrators are intended to test the Romulus technologies, tuning the technologies to each particular situation.
- Market orientation. Demonstrators should be close to the problems companies may have, and with a marketing vision.
- Training material. Demonstrators are a source of training material for new users of Romulus technologies.
- Added value. Demonstrators have to underline the new value added in Web application development by Romulus framework through the development of ROMA framework and Liferay portal and their integration.
- Application development performance. Web application development performance increases using the Romulus framework.

Based on the previously defined criteria, three demonstrators are going to be developed to evaluate the different functionalities that Romulus provides.

The goals of the demonstrators are:

- To offer a global vision of Romulus and test the integration of all its modules.
- To provide training resources for a first contact with Romulus.
- To test concerning the usability of Romulus toolset.

## 3 Romulus Release packaging

In order to provide an integrated and stable version of the different tools that integrate Romulus, it

has been decided to provide a collaborative portal for Java developers which provide these tools, so-called Java Team.

### **3.1 JavaTeam**

JavaTeam is a forge portal, a platform that allows users to make collaborative development and that also integrates Romulus tools for mashup development and security test in the portal.

The forge part of the portal contains the collaborative portlets of World of Liferay which have a standard configuration with similar functionalities to a social network. For example, the users can see the activity of the others and send messages to each other. It is targeted to developers and includes a portlet to follow the SVN activity of each user. Currently it is available at SourceForge. Besides the portal contains a semantic portlet which connects with Jira through a plug-in that must be installed previously in a Jira system, this portlet organizes the bugs in a semantic way.

Each user of the portal can access to the following Romulus tools:

- MyCocktail. A mashup builder that allows users to develop mashup easily with a graphic user interface. It is able to obtain data provided by external services as Flickr, Yahoo Services, Amazon Search, Google Services and a Roma Demo Blog Application.
- Wapiti. It allows the user to audit the security of the web applications.
- Roma demonstrator. An application developed with Roma Meta Framework to show the capabilities of the framework.

## **4 Vertical demonstrators**

Three projects are going to be developed to demonstrate the applicability of Romulus technologies: Musiteca, Cornelius, and Scrooge. These projects are described in more detail in the next sections.

### **4.1 Musiteca**

#### **4.1.1 Introduction**

The aim of this vertical demonstrator is to develop a musical content management application, where the user could watch or listen music pieces.

The vertical demonstrator will allow:

- Content authoring and cataloguing, based on a domain metadata model (Composition, Composer, Content, Content Segment).
- Content Search based on metadata.
- RSS publishing of new contents.

- Social Tagging of contents.
- Creation of communities which will be able to build portals with their contents.
- Content mashups (geolocalisation of Composers, graphical mashups for content filtering and publication, ...).
- Integration of Tevere Workflow (for translation, authorisation, ...).

The main rationale behind this demonstrator is that this is well-known domain for GESFOR and UPM, who have collaborated in several European and National Research projects, related to musical digital libraries. This good understanding of the domain allows to compare Romulus solutions with the previously used options (custom development or based on a content management system). The main limitation of the previous approaches were:

- productivity of the developments. Development of wizard and search facilities has supposed several PM years
- efficiency of the solution. The selected domain is demanding in term of relationships between entities and queries through their relationships. In the case of Content Management Systems, such as Alfresco, it is not easy to modify existing search facilities for metadata. The development of a custom facility is easier, but its maintenance is not cost-effective, since adding new facilities require to understand and modify the system.
- Integration in a portlet environment. The usage of a portal like Liferay and a CMS like Alfresco requires its integration, which is difficult to address mainly because of the integration of user and session management systems.

The demonstrator will evaluate if these problems are overcome thanks to the usage of Romulus technology and will provide a solution which could be easily adapted to customer needs, focusing on the description of the customer metadata domain.

## 4.1.2 Romulus modules

This section describes the Romulus modules that will be used in the development of the demonstrator to achieve the desired features.

#### 4.1.2.1 WP3 – ROMA Metaframework Architecture and Development Tools

| Technology                               | Description of use   |
|--|--|
| <b>Romulus Methodology</b>               | The demonstrator will follow Romulus methodology in order to build the resulting software product.   |
| <b>Roma Metaframework</b>                | <p>Domain Driven Design, the Musiteca model is based in the concept of POJOs.</p> <p>Portlet Development, the Musiteca tools will be integrated in portals that support the JSR 286 specification of Java.</p> <p>Decouple Development, Musiteca will integrate the latest breaking technologies without modifying the domain and application logic because they are really decoupled.</p> <p>Workflow functionalities, Musiteca will use the workflow engine to provide collaborative capabilities to enrich musical contents.</p> <p>JSP View, flexibility design of the web view, including jQuery tools.</p> <p>MVC development, model-view-controller independent of a concrete implementation.</p> |
| <b>Romulus RAD Development tool</b>      | The IDE Plugins for Romulus (Eclipse and Netbeans) will be used to increase productivity in the application development.   |
| <b>Semantic Powered Development Tool</b> | Based on the Semantic Aspect, metadata will be exposed using RDFa along with the Music Ontology.   |

Table 1: Modules from WP3 which will be used in the development of Musiteca

#### 4.1.2.2 WP4 – ROMULUS Data and Services Mashup for Enterprise Integration

| Technology                         | Description of use  |
|------------------------------------|---|
| <b>Mashup builder (MyCocktail)</b> | This tool will be used to build the web services mashups, for example for integrating information of Composers.                                     |
| <b>Portal mashups</b>              | Integration with other applications, such as music players, lyrics search or other information sources could be achieved using widget technologies. |
| <b>Semantic Web Pipes</b>          | In conjunction with MyCoctail, it can be used to build mashups if the data from Musiteca will be available in RDF.                                  |
| <b>Sindice Widgets/SEAP</b>        | Similar as SWP, can be used to create mashups but slightly more sophisticated and demanding.  |

Table 2: Modules from WP4 which will be used in the development of Musiteca

### 4.1.2.3 WP5 – Rich Client Web Technologies, security and reliability

| Technology  | Description of use   |
|---|--|
| <b>Emerging Portal Technologies prototype</b>       | Scripting technologies could be used to simplify user input validation.<br><br>When the application is deployed as a portlet, the categorisation can be navigated using other portlets and interportlet communication. |
| <b>Testing and Security prototype (Wapiti)</b>      | Wapiti tool will be used to check the security of the application  |
| <b>Testing and Security prototype (ATP4Romulus)</b> | ATP4Romulus will ensure the correctness of the demonstrator generating structural, functional, and performance test cases.   |

*Table 3: Modules from WP5 which will be used in the development of Musiteca*

## 4.2 Cornelius

### 4.2.1 Introduction

Cornelius is a vertical demonstrator for project management. The aim of Cornelius is to provide projects, activities and resources management, with a user-friendly application. The target users of Cornelius are Project managers and company's employees.

The main features of the demonstrator are:

- Management of information of hours-worked by employees over an activity
- Count of project's real and forecast costs
- Report extraction about hours-worked, project's cost or forecast resources effort
- Automatic mail-alerting system
- Information hiding policy based on three different profiles: administrators, superuser and users

The main rationale behind this rationale is the business opportunity of a web project management application which can be adapted to every customer and integrated with their internal systems.

### 4.2.2 Romulus modules

This section shows which Romulus modules are used in the development of Cornelius.

#### 4.2.2.1 WP3 – ROMA Metaframework Architecture and Development Tools

| Technology                               | Description of use   |
|--|--|
| <b>Roma Metaframework</b>                | <p>Domain Driven Design: Cornelius business model is based on the concept of POJOs.</p> <p>PersistenceAspect module: Cornelius use Persistence Aspect to manage data into a database.</p> <p>ViewAspect module: Cornelius layout use Roma Framework View aspect to show information</p> <p>Admin and Users modules: used for login operation, profile management</p> <p>Messaging module: used in mailing system</p> <p>Chart module: used for creating project's charts</p> <p>Cornelius could be deployed as a portlet, so it can be integrated with Liferay Portal.</p> |
| <b>Romulus RAD Development tool</b>      | The IDE Plugins for Romulus (Eclipse and Netbeans) will be used to increase productivity in the application development.   |
| <b>Semantic Powered Development Tool</b> | Future integration will be provided with the SemanticAspect in order to export semantic information about projects (such as milestones in iCal format or resources information using DOAP)   |

*Table 4: Modules from WP3 which will be used in the development of Cornelius*

#### 4.2.2.2 WP4 – ROMULUS Data and Services Mashup for Enterprise Integration

| Technology                           | Description of use   |
|--------------------------------------|--|
| <b>Enterprise Integration Mashup</b> | <p>The Cornelius application will be enriched with a set of Use Cases that make use of Enterprise Integration Mashup capabilities.</p> <p>In particular, the personal data of the human resources involved in the projects and the customer data will be collected by different sources and mashed up, in order to be used inside the application. Moreover, the allocation of resources will be performed taking into account their current calendar, which is provided by other applications available in the enterprise.</p> <p>Every integration will be created by using the Enterprise Integration Mashup approach and the tools developed as result of WP4.</p> |

*Table 5: Modules from WP4 which will be used in the development of Cornelius*

### 4.2.2.3 WP5 – Rich Client Web Technologies, security and reliability

| Technology                                   | Description of use   |
|--|--|
| Emerging Portal Technologies prototype       | Based on pure client side components, visualization of data could be customized.   |
| Testing and Security prototype (Wapiti)      | Wapiti tool will be used to check the security of the application  |
| Testing and Security prototype (ATP4Romulus) | ATP4Romulus will ensure the correctness of the demonstrator generating structural, functional, and performance test cases. |

*Table 6: Modules from WP5 which will be used in the development of Cornelius*

## 4.3 Scrooge

### 4.3.1 Introduction

Scrooge is a vertical demonstrator on the personal finance area. The goal of Scrooge is providing personal and enterprise financial management, helping users to control their expenditures. The demonstrator is oriented to be part of the electronic bank services which banks provides to the users.

The services provided by this demonstrator are:

- Personal categorization of credit cars, bills, etc., using folksonomies.
- Categorisation of expenses, based on its data or social tagging.
- Reporting of expenses in graphical way, according to their categorisation.
- User logging
- Comparison of expenses of one user and the 'social network'.

The rationale of this demonstrator is the growing interest in this applications (Wesabe, Mint, ...) by banking entities, and the good position in this market by several companies of the consortium. In addition, this application has been selected in order to provide evaluate how Romulus technology can fit in a very interactive application, designed mainly with client technology.

### 4.3.2 Romulus modules

This section describes the Romulus modules which will be involved in the development of Scrooge in other to achieve the features proposed.

#### 4.3.2.1 WP3 – ROMA Metaframework Architecture and Development Tools

| Technology                          | Description of use  |
|-------------------------------------|---|
| <b>Romulus Methodology</b>          | The demonstrator will follow Romulus methodology in order to build the resulting software product.  |
| <b>Roma Metaframework</b>           | <p>Domain Driven Design, Roma Metaframework allows to build the web application with little effort using POJOs.</p> <p>Decouple Development, Scrooge will integrate the latest breaking technologies without modifying the domain and application logic because they are really decoupled.</p> <p>JSP View, flexibility design of the web view, including jQuery tools.</p> <p>MVC development, model-view-controller independent of a concrete implementation.</p> |
| <b>Romulus RAD Development tool</b> | The IDE Plugins for Romulus (Eclipse and Netbeans) will be used to increase productivity in the application development.  |

Table 7: Modules from WP3 which will be used in the development of Scrooge

#### 4.3.2.2 WP4 – ROMULUS Data and Services Mashup for Enterprise Integration

| Technology                         | Description of use   |
|------------------------------------|--|
| <b>Mashup builder (MyCocktail)</b> | It will be used to build mashups like statistics diagrams.   |
| <b>Portal Mashup</b>               | <p>They will be used to integrate additional features such as a calculator, a currency converter or to display financial news.</p> <p>The will be used to provided different views of information.</p> |

Table 8: Modules from WP4 which will be used in the development of Scrooge

#### 4.3.2.3 WP5 – Rich Client Web Technologies, security and reliability

| Technology  | Description of use   |
|---|--|
| <b>Emerging Portal Technologies prototype</b>       | They will be used to allow users to calculate new values based on existing data.   |
| <b>Testing and Security prototype (Wapiti)</b>      | Wapiti will ensure the project is secure.  |
| <b>Testing and Security prototype (ATP4Romulus)</b> | ATP4Romulus will ensure the correctness of the demonstrator generating structural, functional, and performance test cases. |

Table 9: Modules from WP5 which will be used in the development of Scrooge

## 5 Conclusions

This document defines the vertical demonstrators in order to evaluate Romulus technologies and their potential. The application of them into real contexts will be useful to detect errors, weaknesses and improvement opportunities before the correction requires too much effort or its negative impact is significant. Three demonstrators have been described in order to cover of all Romulus modules, promoting the evaluation and the feedback of each module and also the benefits of using each one in real contexts.