



Deliverable Number: D26.3
Factsheet Number: #1
*Software prototype Name: “Innovation
Ecosystem Platform – Core Components”*

Document Owner:	Martino Maggio (ENG), Davide Storelli (ENG), Calogero Crapanzano (ENG)
Contributors:	H. Boyé (Hardis), Miguel Loitxate Cid (Tecnalia)
Dissemination:	Public
Contributing to:	WP2
Date:	04/10/2012
Revision:	1.0

List of Abbreviations

BPM	Business Process Management
BPMN	Business Process Management Notation
CAS	Central Authentication Server
CMS	Content Management System
CRUD	Create Read Update Delete
DMS	Document Management System
ESB	Enterprise Service Bus
GUI	Graphical User Interface
IEP	Innovation Ecosystem Platform
IPMS	Ideation Process Management System
OSGi	Open Service Gateway initiative
RDF	Resource Description Framework
REST	Representational State Transfer
SPARQL	Simple Protocol and RDF Query Language
SSO	Single Sign On
URL	Uniform Resource Locator
USDL	Unified Service Description Language

Table of Contents

1	AVAILABILITY AND CONTACTS.....	4
2	ARCHITECTURE AND FUNCTIONALITIES.....	5
	TECHNICAL INFORMATION	9
2.1	TECHNICAL DETAILS	9
2.1.1	<i>Single Sing On (SSO)</i>	9
2.1.2	<i>Ecosystem Front-end</i>	9
2.1.3	<i>Ideation Process Management System (IPMS)</i>	9
2.1.4	<i>Enterprise Service Bus (ESB)</i>	9
2.1.5	<i>USDL Repository</i>	10
3	LICENSING.....	11
3.1	SERVICE LICENSE	11
3.2	THIRD PARTY LICENSES	11
4	TECHNICAL MANUAL	12
4.1	SINGLE SIGN ON (SSO)	12
4.2	ECOSYSTEM FRONT-END	12
4.3	IDEATION PROCESS MANAGEMENT SYSTEM (IPMS)	13
4.4	ENTERPRISE SERVICE BUS (ESB)	13
4.5	USDL REPOSITORY	13
4.6	SOURCE CODE.....	14
5	USER MANUAL.....	15
5.1	ACCESS.....	15
5.2	USER MANUAL	15
5.2.1	<i>Operational/Ecosystem-governance Process Management System</i>	15
5.2.2	<i>Ideation Process Management System</i>	22
5.2.3	<i>USDL Repository</i>	28
6	FUTURE PLANS.....	30
7	REFERENCES	31

1 Availability and Contacts

This table is described how to reach the prototype and the contact person in case of questions.

Version	1.0
Availability	
Accompanying specification and design document	MSEE D26.1 – Innovation Ecosystem platform specifications and architecture
Source control	http://engrep.eng.it/svn/msee/SP2/WP26/D26.3
Contact person	Refer to owners of this document

2 Architecture and Functionalities

The Innovation Ecosystem Platform (IEP) is a complex environment composed by several integrated pieces of software. The Architectural components of the first prototype, depicted in Figure 1, will be presented in this chapter. These components, called Core Components of the IEP, includes all the functionalities that are directly related to management of the processes and to the integration issues.

The main component is built on top of a web-portal server (Liferay) working on top of an application server (Apache Tomcat). In the portal server are integrated all the services compatible with standard JSR-168. Other services incompatible with this environment reside in other environments.

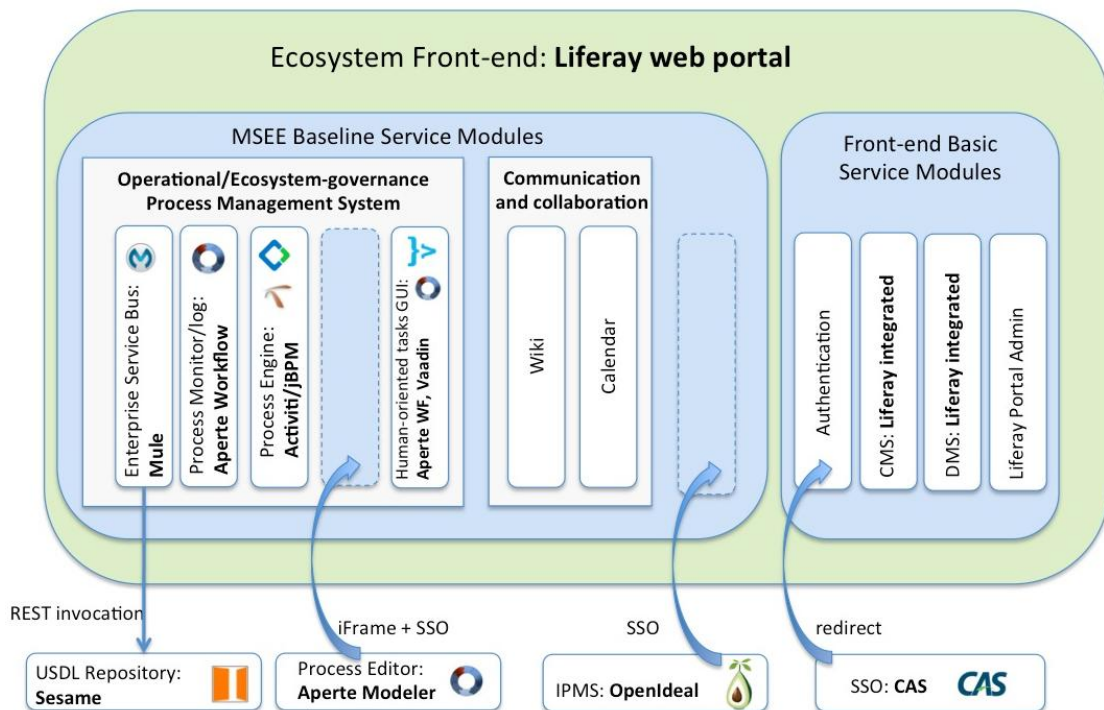


Figure 1. Innovation Ecosystem Platform Core Components.

The Innovation Ecosystem Platform Architecture consists of the following components categories:

- MSEE Baseline Service Modules,
- Front-end Basic Service Modules,
- External App Modules.

MSEE Baseline Service Modules are composed in turn by:

- Modules related to the Operational/Ecosystem-governance process management system:
 - Process Editor,
 - Human-oriented tasks GUI,
 - Process Engine,
 - Enterprise Service Bus,
 - Process Monitor/log,
- Communication and collaboration modules, like shared calendars, wiki, forums;

Front-end Basic Service Modules implement basic functionalities like authentication, content management and document management. These modules are:

- Authentication module,
- Content Management System Module,
- Document Management System Module,
- Portal Administration

Finally, some of the components are externally deployed and integrated with the Front-end through different techniques, including authentication integration (through SSO, graphical integration or simple link/redirect). Such external components are:

- The Idea Process Management System,
- USDL Repository
- the Process Editor
- the SSO server

The **Ecosystem Front-end** is implemented using the enterprise open source portal Liferay [5]. The portal is the centralized interface through which ecosystem users can access to the functionalities of the Innovation Ecosystem Platform. These functionalities are available from the portal in the form of portlets, iframe or links, and are integrated with the portal through a Single Sign On system that allows users to switch from one functionality to another in a simple and transparent way. Users log in the portal and can access to all functionalities enabled in respect of their role.

The **Operational/Ecosystem-governance Process Management System** is implemented using the open source Aperte Workflow Platform [2]. This software solution is a combination of leading open source technologies: Liferay [5] and Vaadin [6] for the user interface, OSGi framework [7] for the plugin support, Hibernate/JPA [8] for persistence, Activiti [9] for BPM engine implementation and Mule ESB [4] for enterprise service bus functionalities.

The entire process management system is composed by:

- **Process Editor:** implemented using the Aperte Modeler web application. Aperte modeler is a BPMN2.0 process editor integrated with Aperte Workflow. It is based on Signavio Core Components [10] open source library, and it is tailored to match specific needs of Aperte Workflow runtime engine. With this editor it is possible to manage and model BPMN2.0 processes, to define automatic and human tasks. For human tasks it is possible to design the user interface using a graphical editor, and to assign human tasks to users or roles. For automatic tasks it is possible to link the task with a Mule ESB flow, allowing in this way the interaction between the process and the external environment. When a process is modeled and all information about its execution behavior are defined, it is possible to publish it in the process management engine, making it available to ecosystem users.
- **Human-oriented tasks GUI:** implemented by Aperte Workflow Activities portlets. This is a graphical interface accessible through the Ecosystem Front-end that allows users to create new processes, search existing processes, browse task queues and handle process tasks. Users can start processes according to their roles and accessing rules defined in the process editor. In the task queue users can see new tasks to be handled, or recent tasks already completed. They also can trace custom task queues defined in the process editor. After starting a new process, users are asked to fill some information in a form that was created in the process editor.
- **Process Engine:** this is the engine used by the process management system for executing processes created using the process editor. Aperte Workflow, the platform used to implement the process framework, supports currently two BPM engines:

jBPM [11] and Activiti [9]. The process engine implementation used in the MSEE Ecosystem-governance Process Management System is Activiti, because of its more mature compatibility with the BPMN2.0 notation. The process engine is integrated within the process management framework, and the deployment consist of the installation of a OSGi bundle containing the process definition. Moreover this operation is done automatically by the process editor.

- **Enterprise Service Bus:** this is the module which support the enterprise integration, and it is implemented using an embedded version of Mule ESB [4]. Mule ESB is integrated in Aperte Workflow platform by means of an OSGi bundle, containing an instance of the Mule server. An automatic task defined with the process modeler is linked to a Mule Flow, also deployed as a bundle. This flow is executed within the Mule server instance integrated in the platform. In the MSEE Ecosystem-governance Process Management System there are two instances of the Mule server, one integrated in the process framework, and one instance standalone. This distinction allows to reduce the complexity and workload of automatic tasks and externalize them in other environments (like other server). In this way, simple automatic processes are executed within the process framework, whereas more complex processes are executed in the external Mule instance. This separation allows also modularity and scalability: it is possible to add other Mule server instances, and use new Mule versions without the need to change the bundle installed in the framework. It is also possible to use other ESB external implementations.
- **Process Monitor/log:** this module allows the monitoring of the state of a process. Aperte Workflow provides two functionalities in the Liferay Control Panel: Aperte Workflow Process Definitions and Aperte Workflow Process Instances. The first functionality allows administrators to take trace of all process definitions, with their description, version and state (enabled/disabled). It is also possible to disable/enable a process definition. If a process is disable users can't create new instances of the process. The second functionality allows administrators to take trace of all process instances, both active and inactive. For each instance it is possible to see the creation date, the creator, the current task and owner. It is also possible to see a graphical process map that shows the BPMN2.0 process, with the information of the current state (completed tasks are highlighted with a different color). It is also possible to see the process history, whit details about each user that participated to the process end about completed tasks.

Communication and collaboration modules are inherited from Liferay Portal basic applications. These modules are blogs, forum, wiki or calendar. With Blog Portlet ecosystem users can run their own blog service. With Message Board Portlet it is possible to create forums where ecosystem users can exchange their views. Wiki Portlet allows creation of contents in the Wiki collaboration style. Calendar Portlet allows community users to have a common calendar with tasks list, where users can create, manage and search for events.

Front-end basic service modules implements the basic functionalities for the portal:

- **Authentication:** this module allows the integration of the authentication mechanism of all the applications accessible from the Ecosystem Front-end. It allows users to access to each application without the need to repeated login. this module is implemented using the CAS Central Authentication Service [1]. The CAS server is responsible of user authentication, and communicate with the front-end (Liferay) by means of an integrated and configurable Liferay CAS module. The integration with applications realized with other technologies is granted by the large number of CAS client libraries (Java, .Net, PHP, Perl and others). Specifically, the integration with the

Idea Process Management System (developed in PHP) is implemented using the CAS PHP client library phpCAS [12] in conjunction with a Drupal [14] module for CAS integration [13].

- **CMS:** the module for the content management provides the basic function of publishing content in the portal. The implementation of this modules uses the Content Management core Liferay Portlet. It allows to manage structured and unstructured content to be published in a web site.
- **DMS:** the module for the document management provides functionalities to upload/download documents in the portal, to organize/search documents and to manage access permissions to documents. The implementation of this module is provided by the Liferay Document Management Portlets.
- **Portal Administration:** the portal administration module provides administrative functionalities like users, profiles, groups and organizations management, monitoring operations, communication and collaboration supervision, and so on. The module is implemented by the Liferay Control Panel.

The **Idea Process Management System** exposes all the functionalities for the idea management, from idea creation, to idea enhancement with comments and evaluations, to idea workflow management (new ideas, draft ideas, accepted ideas, refused ideas, realized ideas). It also exposes functionalities for categorize ideas, search existing ideas, see most commented or latest ideas and so on. This module is implemented by a customization of the open source Open ideal [3] web application. This is a Drupal based product, integrated in the fornt-end using the authentication module, and in specifically with the library and module described before (phpCAS + Drupal CAS module).

The **USDL Repository** is implemented through the Sesame2 [15] RDF Store, which allows the management of RDF models describing both (in)tangible assets and services that expose such assets. Descriptions are encapsulated in RDF Graphs to guarantee easy management of their lifecycle. Operations on resources is enabled by a web-application that provides a GUI for executing CRUD operations and SPARQL 1.1 queries. The same operations are exposed through a RESTful architecture for programmatic management of statements and RDF Graphs.

Technical Information

2.1 Technical details

2.1.1 Single Sign On (SSO)

Nature	Web Application
Programming Language	Java
Development Framework	JDK 1.6, Windows XP/2003/2008 server Linux (not tested)
Additional libraries	-
Application Server	Apache Tomcat 6.0.35
Database	MySQL 5.1.36

2.1.2 Ecosystem Front-end

Nature	Web Portal
Programming Language	Java
Development Framework	JDK 1.6, Windows XP/2003/2008 server Linux (not tested)
Additional libraries	-
Application Server	Apache Tomcat 7.0.27
Database	MySQL 5.1.36

2.1.3 Ideation Process Management System (IPMS)

Nature	Web Application
Programming Language	PHP
Development Framework	PHP 5.3.0, Windows XP/2003/2008 server Linux (not tested)
Additional libraries	-
Application Server	Apache HTTP Server 2.2.11
Database	MySQL 5.1.36

2.1.4 Enterprise Service Bus (ESB)

Nature	Standalone Application
--------	------------------------

Programming Language	Java
Development Framework	JDK 1.6, Windows XP/2003/2008 server Linux (not tested)
Additional libraries	-
Application Server	-
Database	-

2.1.5 USDL Repository

Nature	Web Application
Programming Language	Java
Development Framework	Java 5 or newer, Windows XP/2003/2008 server Linux (not tested)
Additional libraries	-
Application Server	Apache Tomcat 7.0.27
Database	-

3 Licensing

3.1 Service license

The license is to be determined. The IEP is currently in a prototype state and not intended for public use, but a physical running instance is managed and provided by Engineering. Technologies used are available open source or similar licenses allowing usage of the software without extra fees or limitations.

3.2 Third party licenses

The Single Sign On is implemented using the CAS (Central Authentication Service) authentication system. CAS is released under the Apache License, Version 2.0.

The Innovation Ecosystem Platform Portal is implemented using the Aperte Workflow software, that is released under the GNU LGPLv3 license.

The Idea Management System is implemented using the Open ideaL web application that is an open source product licensed under the GNU GPLv2 License.

The Enterprise Service Bus is implemented using the Mule ESB Community Edition, that is licensed under the Common Public Attribution License (CPAL). Sesame 2 is available under a BSD-style license. Moreover it includes software developed by the Apache Software Foundation

Third party software	Licence
CAS (Central Authentication Service)	Apache License, Version 2.0.
Aperte Workflow	GNU GPLv3 License
Open ideaL	GNU GPLv2 License
Mule ESB Community Edition	Common Public Attribution License (CPAL)
Sesame2	BSD-style license

4 Technical Manual

4.1 Single Sign On (SSO)

The installer can be downloaded from this address:

<http://engrep.eng.it/msee/iep/bin/sso.zip>.

Extract the file in a folder of choice that we call *%sso_root%*. Now this folder contains:

- The folder *sso*

Edit the file:

%sso_root%/sso/apache-tomcat-6.0.35/webapps/cas/WEB-INF/deployerConfigContext.xml

and change, if needed, these properties:

- *bean datasource -> property url*: set the url of the MySQL RDBMS containing the *iep* database (see IEP Portal installation),
- *bean datasource -> property username*: insert the password for the *iep* database
- *bean datasource -> property password*: insert the username for the *iep* database.

Edit the file:

%sso_root%/sso/apache-tomcat-6.0.35/webapps/cas/WEB-

INF/view/jsp/default/ui/casLogoutView.jsp

and insert the URL of the IEP Portal (change only hostname and port).

Start the Single Sign On executing the script *startup.sh* or *startup.bat* in the folder

%sso_root%/sso/tomcat-7.0.27/bin

4.2 Ecosystem Front-end

The installer can be downloaded from this address:

http://engrep.eng.it/msee/iep/bin/iep_portal.zip

Extract the file in a folder of choice that we call *%iep_portal_root%*. Now this folder contains:

- The folder *iep_portal*
- The *iep.sql* script
- The *iep_aperte.sql* script

run the scripts in a MySQL RDBMS, creating in this way the *iep* and *iep_aperte* databases.

Edit the file *%iep_portal_root%/iep_portal/portal-setup-wizard.properties*, and change these properties:

- *jdbc.default.username*: insert the username for the *iep* database
- *jdbc.default.password*: insert the password for the *iep* database
- *jdbc.default.url*: set the url of the MySQL RDBMS containing the *iep* database
- *liferay.home*: insert the path *%iep_portal_root%/iep_portal*.

Start the IEP portal using the script *startup.sh* or *startup.bat* in the folder *%iep_portal_root%/iep_portal/tomcat-7.0.27/bin*.

The default port of the application server containing the IEP Portal is 8080. Open a browser and go to <http://localhost:8080>.

Log in using administrator credentials [admin@msee.eu/msee].

Go to Control Panel -> Portal Settings -> Authentication -> CAS, and:

- set the Login URL of the SSO (changing the host and the port),
- set the Logout URL of the SSO, (changing the host and the port)
- set the Server Name of the SSO (changing the host and the port),
- set the Server URL (changing the host and the port),
- set the No Such User Redirect URL (changing the host and the port),
- Enable the SSO authentication checking the Enabled checkbox.

Go to Manage → Page → Idea Management, and change the URL of the Idea Management System (only hostname, port and context).

4.3 Ideation Process Management System (IPMS)

The installer can be downloaded from this address:

<http://engrep.eng.it/msee/iep/bin/ipms.zip>

Extract the file in a folder of choice that we call *%ipms_root%*. Now this folder contains:

- The folder *ipms*
- The *ipms.sql* script.

Run the scripts in a MySQL RDBMS, creating in this way the *ipms* database.

Edit the file:

%ipms_root%/ipms/sites/default/settings.php, and change, if needed, these properties:

- "*\$databases -> default -> default -> database*": set the *ipms* database name,
- "*\$databases -> default -> default -> username*": set the *ipms* database username,
- "*\$databases -> default -> default -> password*": set the *ipms* database password,
- "*\$databases -> default -> default -> host*": set the *ipms* database hostname,
- "*\$databases -> default -> default -> port*": set the *ipms* database port,

Copy the *%ipms_root%/ims* folder in the *www* folder of the HTTP Server, open the browser and go to <http://localhost/ipep> (here the port 80 is the default HTTP port).

Log in using administrator credentials [admin/admin] and go to *Configuration -> People -> CAS Settings*.

Change, if needed, these properties:

- *Hostname*: the hostname of the SSO,
- *Port*: the port number of the SSO,
- *URI*: the uri of the SSO

Go to *Redirection* section and check the "*Check with the CAS server to see if the user is already logged in?*" checkbox.

4.4 Enterprise Service Bus (ESB)

The installer can be downloaded from this address:

<http://engrep.eng.it/msee/iep/bin/esb.zip>.

Extract the file in a folder of choice that we call *%esb_root%*. Now this folder contains:

- The folder *esb*.

Set the *MULE_HOME* environment variable to point to the location of the ESB installation:

%esb_root%/esb/mule-standalone-3.3.0.

Start the ESB using the script *mule.bat* in the folder

%esb_root%/esb/mule-standalone-3.3.0/bin.

4.5 USDL Repository

The installer can be downloaded from this address:

http://engrep.eng.it/msee/iep/bin/usdl_repo.zip.

Extract the file in a folder of choice that we call *%usdl_repo_root%*. Now this folder contains:

- The file *openrdf-sesame.war*.
- The file *openrdf-workbench.war*.

Then install these web applications in a Tomcat application server. For more information about Sesame2 installation please refer to:

<http://www.openrdf.org/doc/sesame2/2.3.2/users/userguide.html#chapter-server-install>

4.6 Source code

In the following table we indicate how to access to source code of each prototype module.

Module	Component	URL
Single Sign On	Customized version of the Cas Server 3.5.0	http://engrep.eng.it/svn/msee/SP2/WP26/D26.3/cas-server-webapp/
IEP Front-end	Liferay Portal 6.1.1	http://www.liferay.com/downloads/liferay-portal/available-releases
	Aperte Workflow	https://github.com/bluesoft-rnd/aperte-workflow-core
	Customized mule-esb-plugin (to install in place of the provided version by Aperte Workflow)	http://engrep.eng.it/svn/msee/SP2/WP26/D26.3/mule-esb-plugin/
	Eng-Brochure-theme	http://engrep.eng.it/svn/msee/SP2/WP26/D26.3/Eng-Brochure-theme/
Idea Management System	OpenideaL	http://drupal.org/project/idea
	CAS Drupal module	http://drupal.org/project/cas
	IdeaManagement-theme	http://engrep.eng.it/svn/msee/SP2/WP26/D26.3/IdeaManagement-theme/
Enterprise Service Bus	Mule ESB Community Edition	http://xircles.codehaus.org/projects/mule/repos
USDL Repository	Sesame2	http://repo.aduna-software.org/svn/org.openrdf/sesame/

5 User Manual

5.1 Access

The prototype installation is not yet accessible.

5.2 User Manual

5.2.1 Operational/Ecosystem-governance Process Management System

The principal functionalities of the process management system are:

- creating a new process,
- executing an existing process.

The creation of a process involves process modeling using BPMN2.0 notation, and definition of human and automatic tasks.

Lets create a simple process called *msee01-simple-process*.

To create a new process log in the front-end portal and go to section “Process Modeler”, like in Figure 2.

In the process modeler, processes can be organized within folders.

To create a new process folder:

1. From the explorer menu select New -> Folder
2. When asked about the directory name, type in *msee-processes* and press OK.
3. Navigate to the folder double clicking its icon.

Now to create a simple process within this folder:

1. From the Explorer menu select New > Business Process Diagram (BPMN 2.0) for Aperte Workflow,
2. Now new tab or window with process Editor is going to be opened,
3. Save the diagram by clicking the save icon from the Editor top menu, and provide the process diagram name *msee01-simple-process* and a description,
4. From the Editor Shape Repository palette drag and drop Start Event element,
5. Double-click the Start Event element and name it *start*.
6. From the Editor Shape Repository palette drag and drop Task element, and name it *user input*.
7. Connect *start* element with *user input* task.
8. Name the Sequence Flow between these two elements *to user input*. The process should look like in Figure 3:



Figure 2. Process Modeler.

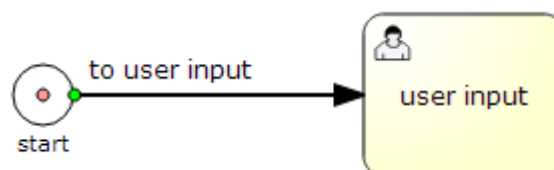


Figure 3. msee01-simple-process, step 1

9. Add others two tasks, and a final event, connect them end provide names, like in Figure 4.

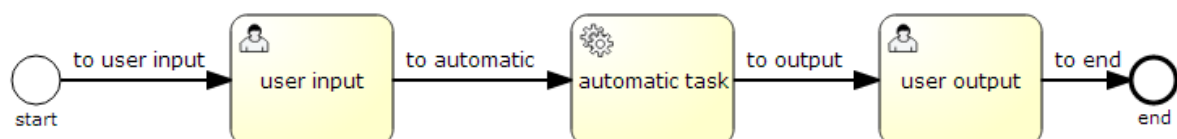


Figure 4. msee01-simple-process, step 2.

10. To tag a task as “automatic”, select the task, open the Aperte Step Editor by clicking on the icon showed in Figure 5, and select the “MuleStep” task type like in Figure 6.

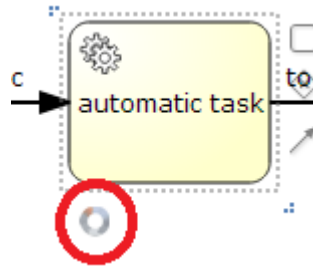


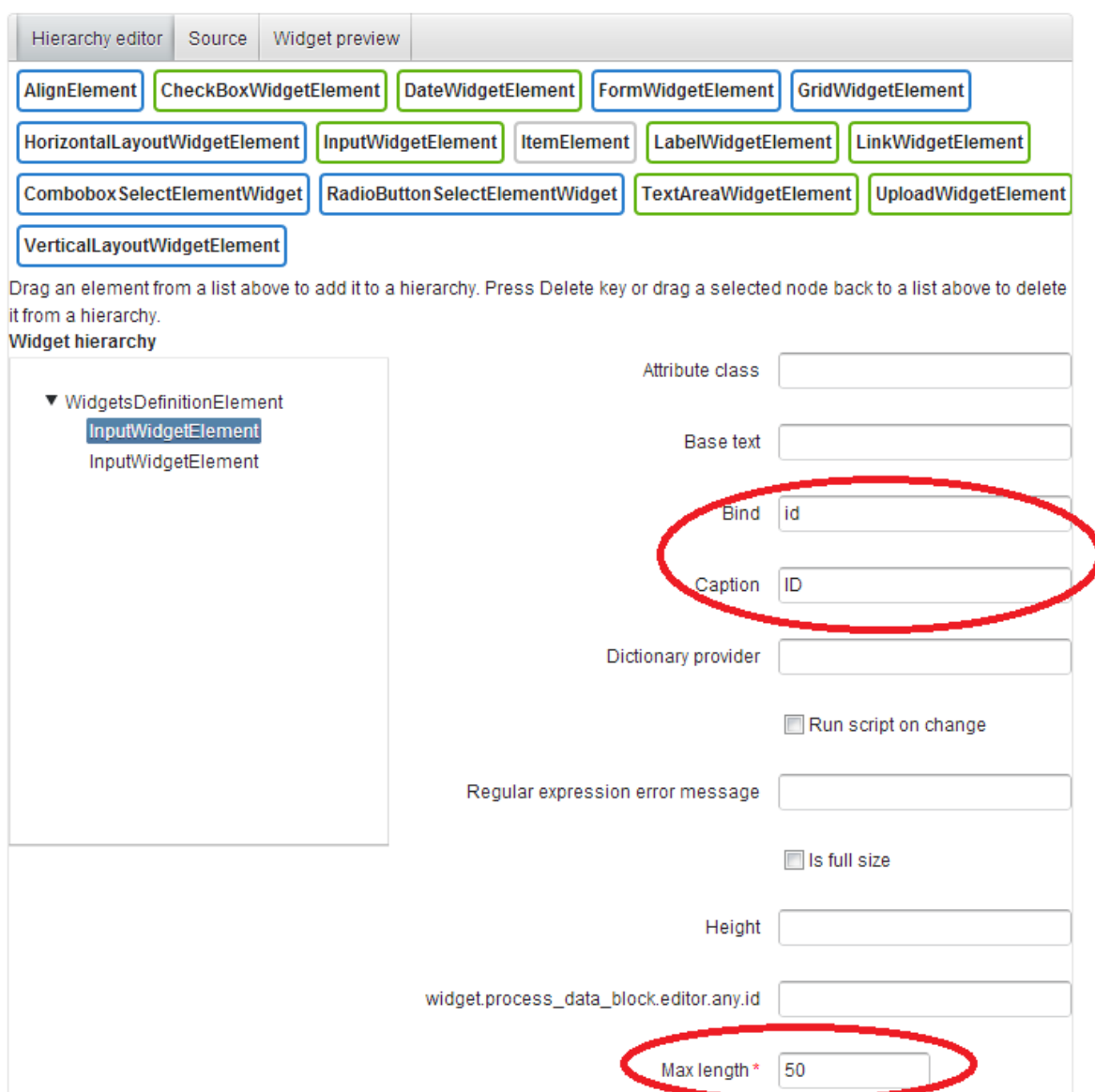
Figure 5. Open Aperte Step Editor

Figure 6. Automatic task (MuleStep)

11. Make sure to save the process diagram.

To define the user interface for human task:

1. Open the Aperte Step Editor on the *user input* task,
2. Select the *User* task type,
3. Switch to *User Interface Definition* tab,
4. Drag new *Vertical Layout* widget to the widget hierarchy tree,
5. Drag new *ProcessData* Block to the Vertical Layout added before,
6. Click on the *ProcessData* Block widget and navigate to the Attributes tab,
7. Provide values for the following attributes: Caption and Comment,
8. Add to the *ProcessData* Block an *inputWidgetElement* and fill these attributes (see Figure 7):
 - a. Bind: *id*
 - b. Caption: *ID*
 - c. Max length: *50*
 - d. Visible: *checked*
9. Add another *inputWidgetElement* and fill these attributes:
 - a. Bind: *staffDetail*
 - b. Caption: *Staff Detail*
 - c. Max length: *300*
 - d. Visible: *checked*
 - e. Is Read Only: *checked*



Drag an element from a list above to add it to a hierarchy. Press Delete key or drag a selected node back to a list above to delete it from a hierarchy.

Widget hierarchy

- ▼ WidgetsDefinitionElement
 - InputWidgetElement**
 - InputWidgetElement

Attribute class:

Base text:

Bind:

Caption:

Dictionary provider:

☐ Run script on change

Regular expression error message:

☐ Is full size

Height:

widget.process_data_block.editor.any.id:

Max length *:

Figure 7. Create new inputWidgetElement.

10. To define permissions on user form switch to the *Permissions* tab in the *ProcessData* widget, and select roles for all permissions. The *.** role name stands for all roles. In this example select *.** role for all permissions.
11. In the *State Definition* tab it is possible to insert a description and commentary.
12. To save the created user interface press the editor Save button.
13. Open the Aperte Step Editor on the *user output* task and create the same elements like in the user input task. But in the first *inputWidgetElement* check the attribute *Is Read Only*.

To define automatic task:

- Open the Aperte Step Editor on the *automatic* task,
- Fill the *destinationEndpointUrl* attribute with the URL of the deployed mule bundle that has to execute this task. In this case insert *vm://searchStaff*.
- Press the editor Save button.

An automatic task is so a link to a deployed mule step bundle. To create and deploy a new mule step bundle follow the following steps. First you will need installed Apache Maven.

1. Create a maven project to build the bundle containing the mule step, with a structure like this:

```

searchStaffTask
  +-- pom.xml
  +-- src
      +-- main
          +-- resources
              +-- flow.xml

```

Figure 8. Automatic task Maven project structure.

2. Create a pom.xml file with the content in Figure 9.
3. In the *src/main/resources* create a new file named *flow.xml* with the content in Figure 10:
4. In the console window, enter main directory of the project and type *mvn package*
5. This will build our bundle. You should find it as *target/searchStaffTask-0.0.1-SNAPSHOT.jar*
6. To deploy the bundle log in the front-end, go to the section “Aperte Workflow Plugin Manager” in the Control Panel.
7. Select the *Upload new or updated plugin* button and upload the bundle file.

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>eu.msee.task</groupId>
  <artifactId>searchStaffTask</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>bundle</packaging>
  <name>searchStaffTask</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <muleVersion>3.2.1</muleVersion>
  </properties>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>2.2</version>
        <configuration>
          <useDefaultManifestFile>true</useDefaultManifestFile>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.felix</groupId>
        <artifactId>maven-bundle-plugin</artifactId>
        <extensions>true</extensions>
        <version>2.1.0</version>
        <configuration>
          <instructions>
            <Bundle-SymbolicName>eu.msee.task</Bundle-SymbolicName>
            <Bundle-Version>0.0.1-SNAPSHOT</Bundle-Version>
            <Bundle-Description>eu.msee.task.searchStaffTask</Bundle-Description>
            <Mule-Config-Files>/flow.xml</Mule-Config-Files>
          </instructions>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>

```

Figure 9. Automatic task - pom.xml file.

To define process execution permissions and human task assignment:

1. Select *user input* task, and open the Aperte Step Editor.
2. Go to the *Assignment* tab.

3. Set the Assignee attribute value *#{initiator}*, in this way the task user input is assigned to the user that initiated the process.
4. Do the same operations with the *user output* task.

At the end save the process end do this last steps to deploy the process:

1. In the “Attributes” section of the modeler set these properties:
 - a. Name: msee01-simple-process
 - b. Aperte process filename: msee01-simple-process
 - c. Manifest: Bundle-Name: eu.msee.processes.simple01
 - d. Manifest: Bundle-Description: first msee example process
 - e. Manifest: ProcessTool-Process-Deployment: msee01-simple-process
2. Select the *to automatic* arrow and open the Aperte Action Editor like in Figure 11, and fill these attributes:
 - a. Priority: 0
 - b. Label: *Search Staff*
3. Select the *to end* arrow and open the Aperte Action Editor, and fill these attributes:
 - a. Priority: 0
 - b. Label: *End Process*
4. Save the process,
5. Click the *deploy* icon showed in Figure 12.

```
<?xml version="1.0" encoding="UTF-8"?>

<mule xmlns="http://www.mulesoft.org/schema/mule/core"
      xmlns:script="http://www.mulesoft.org/schema/mule/scripting"
      xmlns:file="http://www.mulesoft.org/schema/mule/file"
      xmlns:http="http://www.mulesoft.org/schema/mule/http"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:json="http://www.mulesoft.org/schema/mule/json"
      xsi:schemaLocation="
        http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd
        http://www.mulesoft.org/schema/mule/scripting http://www.mulesoft.org/schema/mule/scripting/current/mule-scripting.xsd
        http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd
        http://www.mulesoft.org/schema/mule/json http://www.mulesoft.org/schema/mule/json/current/mule-json.xsd
      ">

  <flow name="searchStaff" >
    <inbound-endpoint address="vm://searchStaff" exchange-pattern="request-response"/>
    <processor-chain>
      <script:component>
        <script:script engine="groovy">
          try {
            def map = [:];
            def staffMap = [:];

            staffMap.put("001", "name: John, surname: Smith, profile: Project Manager");
            staffMap.put("002", "name: Matthew, surname: Shah, profile: Software Architect");
            staffMap.put("003", "name: Joseph, surname: McKenzie, profile: Solution Developer");

            for (attr in payload.getProcessAttributes()) {
              map.put(attr.key, attr.toString());
            }

            String id = map["id"];
            String staffDetail = staffMap[id];
            payload.setSimpleAttribute("staffDetail", staffDetail);
            return payload
          }
          catch (Exception e) {
            e.printStackTrace();
            throw e;
          }
        </script:script>
      </script:component>
    </processor-chain>
  </flow>
</mule>
```

Figure 10. Automatic task - flow.xml file.

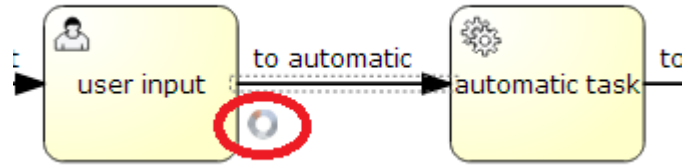


Figure 11. Aperte Action Editor.



Figure 12. Deploy button.

To execute the new process *msee01-simple-process* log in the front-end portal and go to section “Process Management”. In the New Process section, it is possible to start a new *msee01-simple-process* by clicking the Start link (see Figure 13).

Once started the process, you can see the user form of the *user input* task created before, with the *ID* input field. Insert *001* in this field and press the *Search Staff* button (see Figure 14).

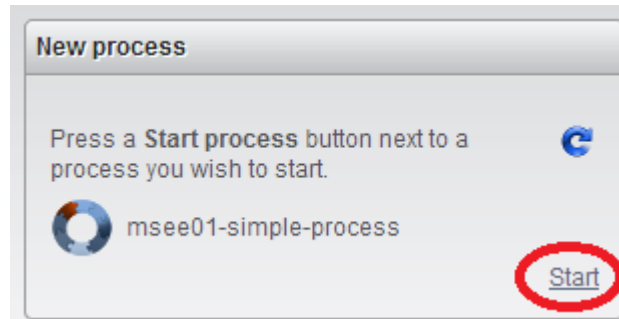


Figure 13. Start new process.

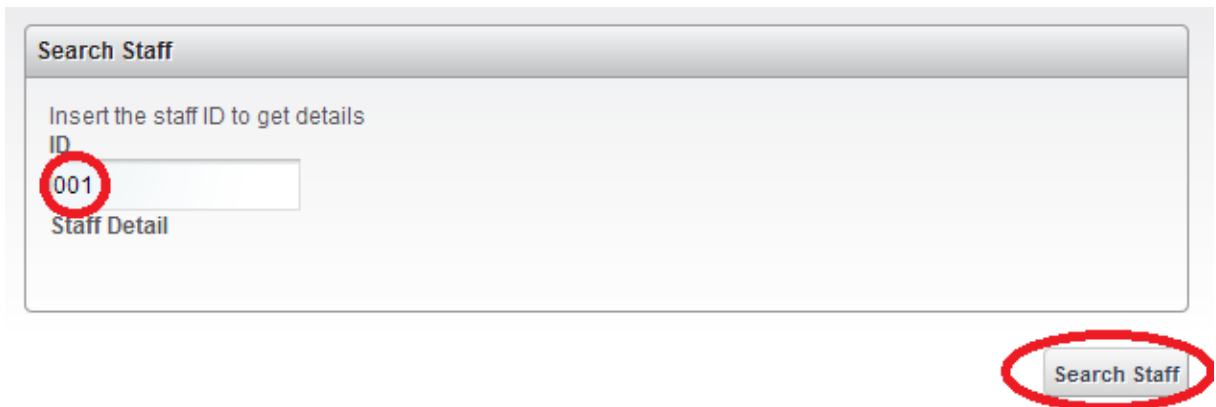


Figure 14. User input task.

After you have clicked the button, the system execute the automatic task, and then shows the *user output* human task, that contains the result of the process in the Staff Detail field (see Figure 15).

ID
001
Staff Detail
name: John, surname: Smith, profile: Project Manager

End Process

Figure 15. User output task.

Finally click the End Process button to end the process.

5.2.2 Ideation Process Management System

To access to the idea management functions log in the front-end portal and go to section “Idea Management”.

Now, to let’s follow the process activity diagram in Figure 16.

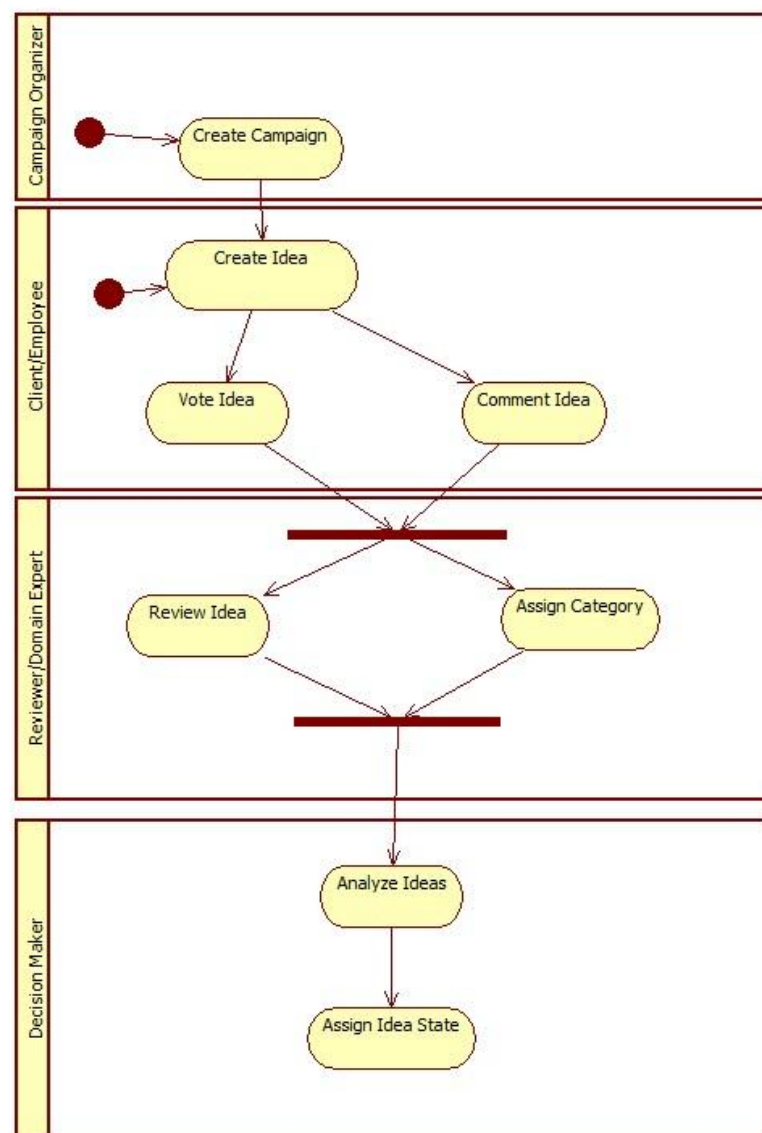


Figure 16. Idea Management Process.

To create a new campaign (or challenge) click on the *Challenge Management* menu link like in Figure 17.

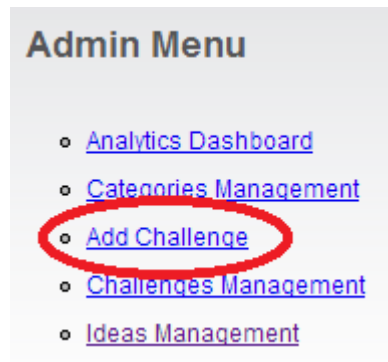
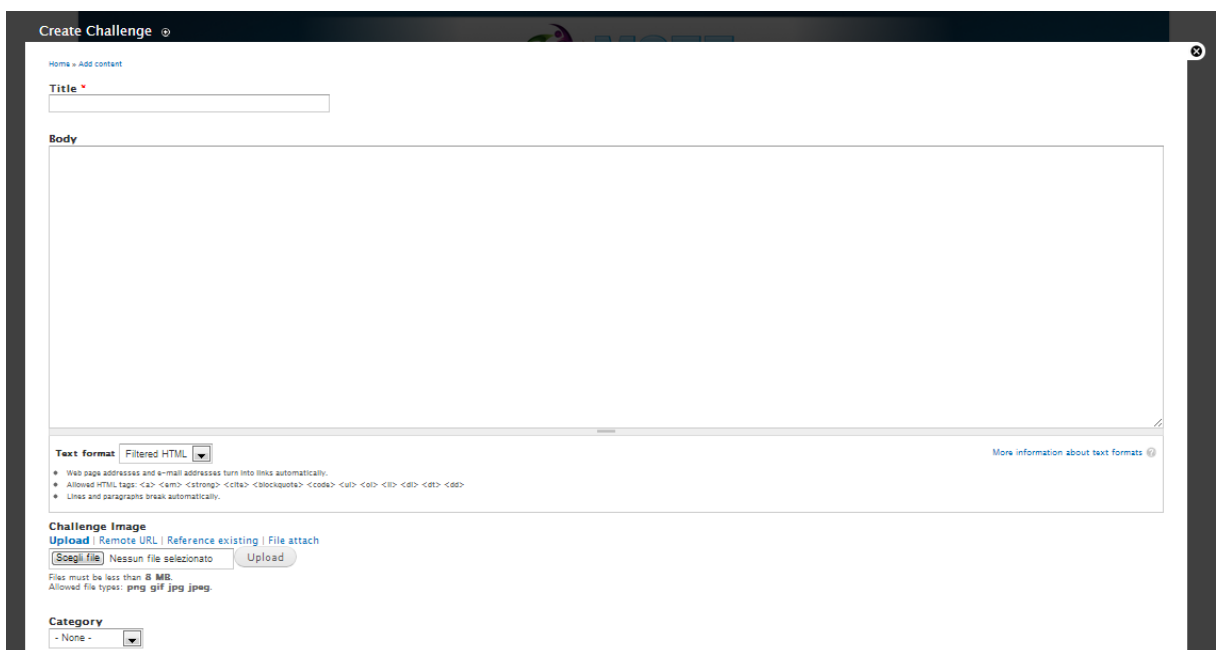


Figure 17. Add Challenge menu link.

You will see the form depicted in Figure 18.



The image shows a 'Create Challenge' form. It includes a 'Title' field, a large 'Body' text area, a 'Text format' dropdown set to 'Filtered HTML', and a 'Challenge Image' section with 'Upload', 'Remote URL', and 'Reference existing' options. Below the image section is a 'Category' dropdown menu.

Figure 18. Create Challenge form.

Here you have to insert the challenge title, and optionally a description of the challenge. It is also possible to define the category of this challenge, selecting one of the existing categories, and to select an image for promoting the challenge. Let's name the challenge *First MSEE Challenge*. Once the challenge is create it will be visible in the *Challenges* page.

Now it is possible to create an idea in response to the new created challenge.

To create the new idea click the *Add Idea* menu link, like in Figure 19.

Idea Management

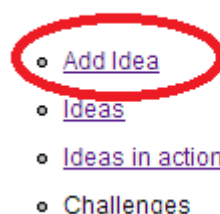


Figure 19. Add Idea menu link.

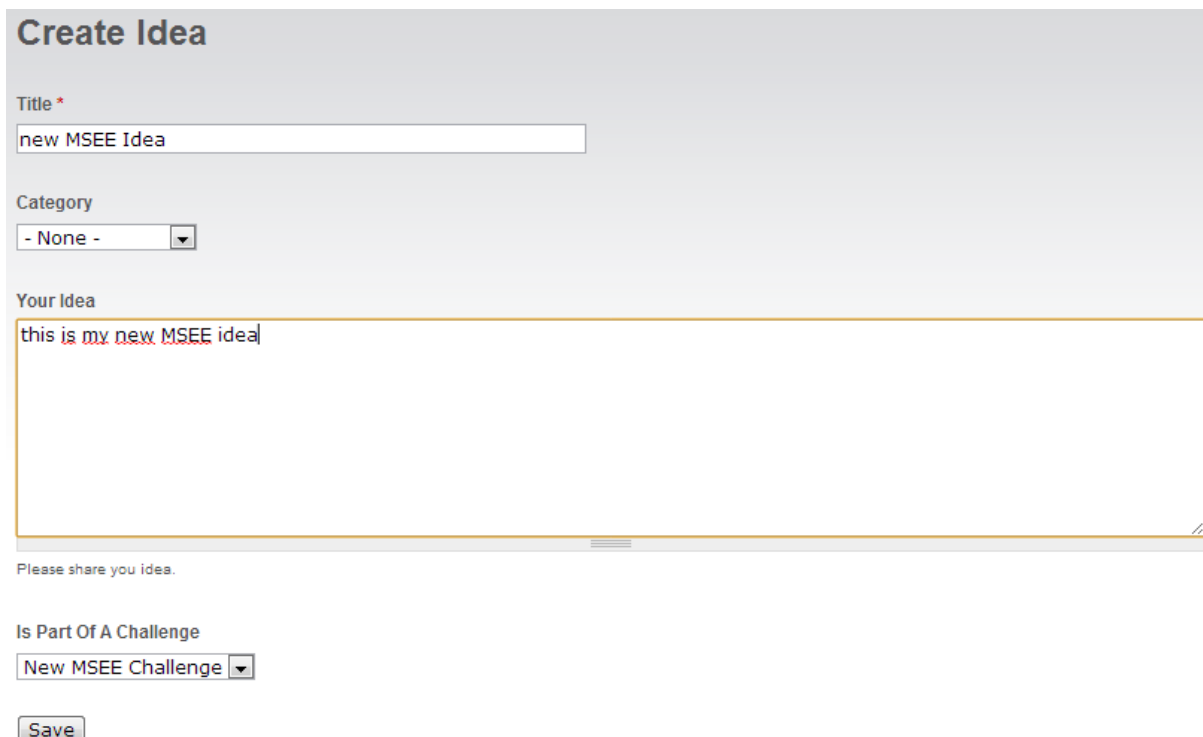
You will see the form depicted in Figure 20.

Here you have to insert the title of the new idea, the content of the idea, and select the challenge *First MSEE Challenge*. To save the new idea press the *Save* button.

Now it is possible to:

- Vote the idea,
- Comment the idea

To vote the idea search the new created idea in ideas list, by clicking the Latest Ideas tab in the Ideas page, like in Figure 21. Then click on the + icon. The score of the voted idea will be incremented by one point.



Create Idea

Title *

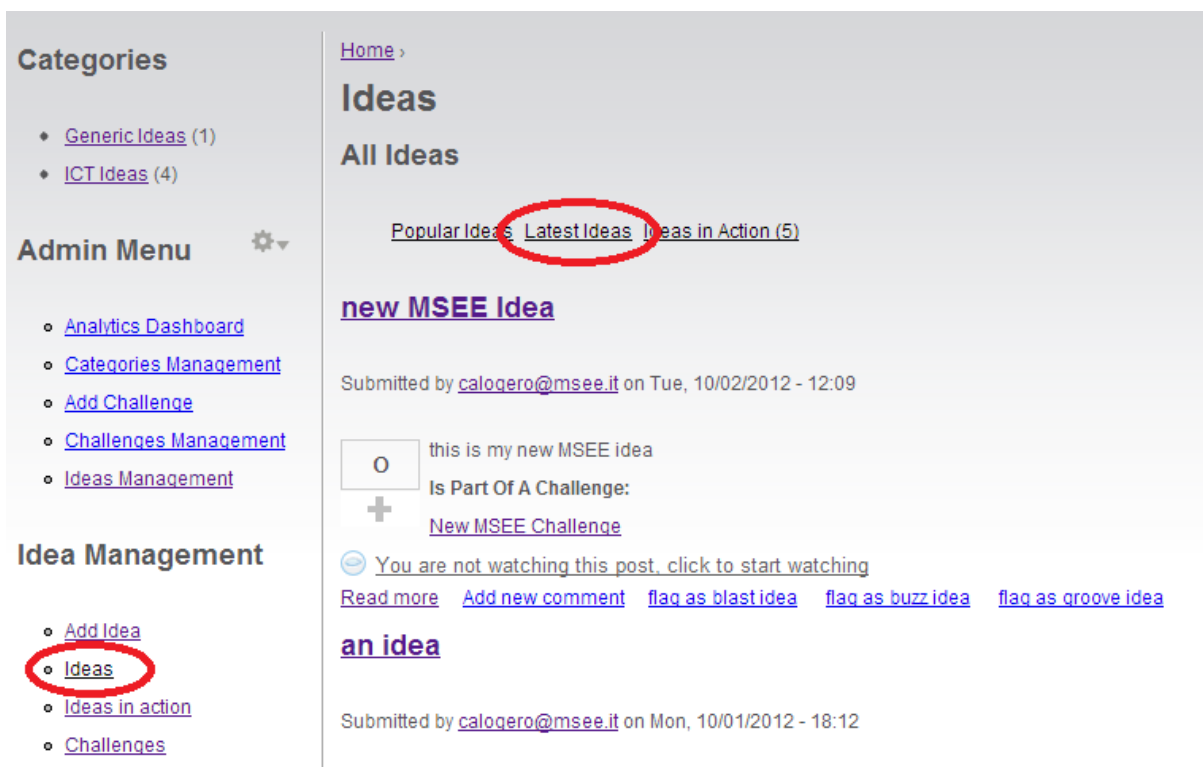
Category

Your Idea

Please share you idea.

Is Part Of A Challenge

Figure 20. Create Idea form.



Categories

- [Generic Ideas](#) (1)
- [ICT Ideas](#) (4)

Admin Menu

- [Analytics Dashboard](#)
- [Categories Management](#)
- [Add Challenge](#)
- [Challenges Management](#)
- [Ideas Management](#)

Idea Management

- [Add Idea](#)
- [Ideas](#)
- [Ideas in action](#)
- [Challenges](#)

[Home](#) >

Ideas

All Ideas


[Popular Ideas](#) [Latest Ideas](#) [Ideas in Action \(5\)](#)

[new MSEE Idea](#)

Submitted by [calogero@msee.it](#) on Tue, 10/02/2012 - 12:09

this is my new MSEE idea

Is Part Of A Challenge:
 [New MSEE Challenge](#)

 You are not watching this post, [click to start watching](#)

[Read more](#) [Add new comment](#) [flag as blast idea](#) [flag as buzz idea](#) [flag as groove idea](#)

[an idea](#)

Submitted by [calogero@msee.it](#) on Mon, 10/01/2012 - 18:12

Figure 21. Search idea.

To create a comment of the new idea, click on the comment link like in Figure 22, and fill the comment area like in Figure 23. To save the new comment push the *Save* button.

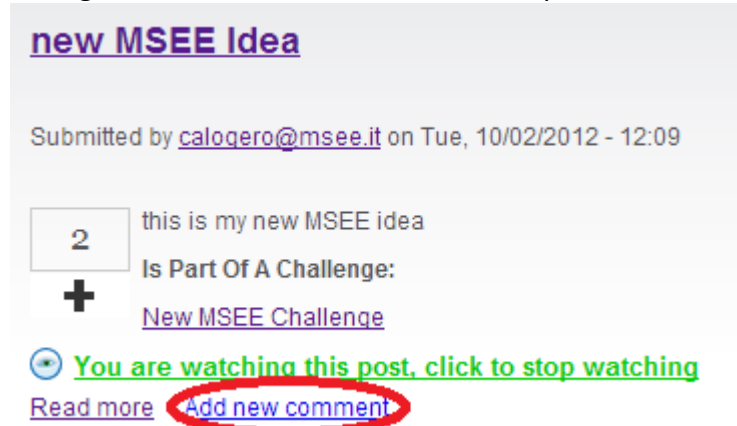
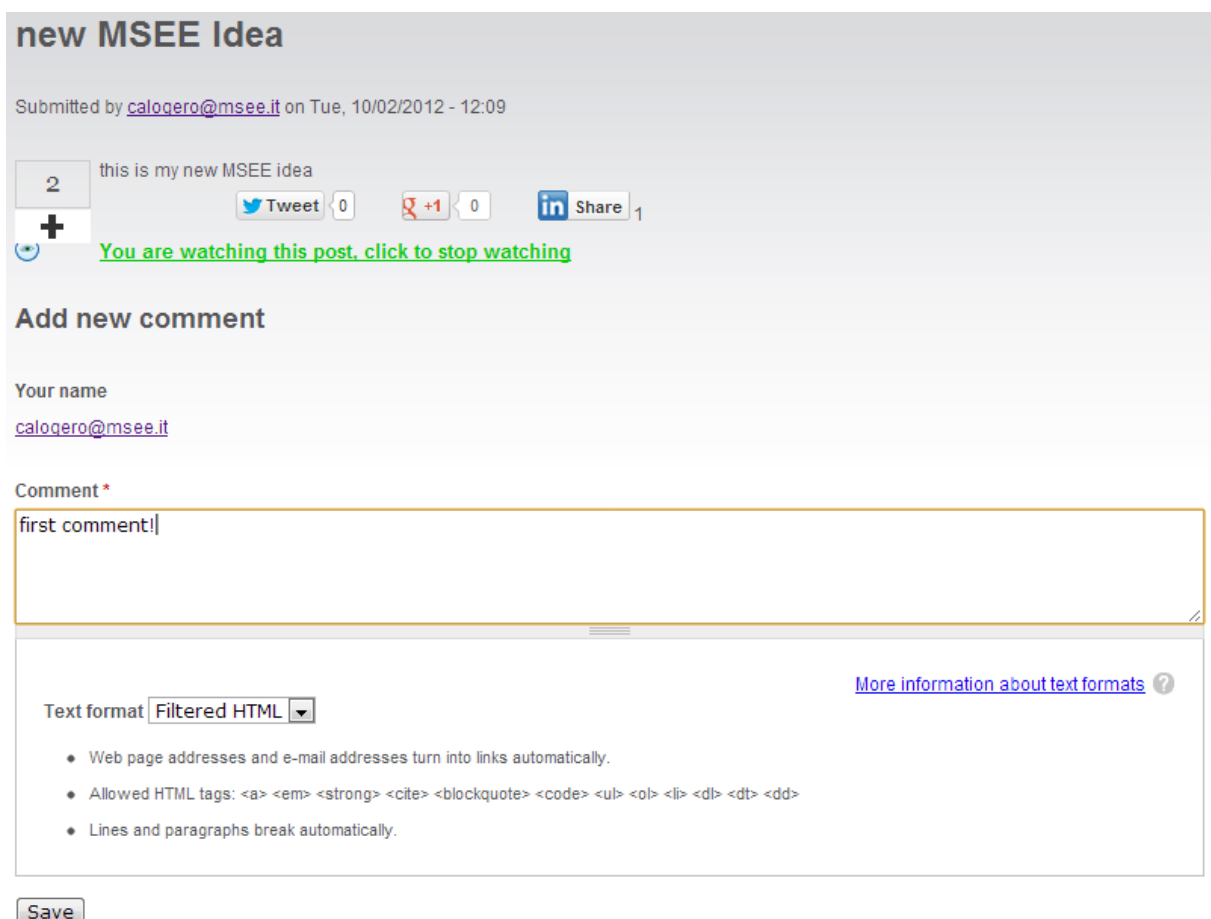


Figure 22. Idea Comment link.

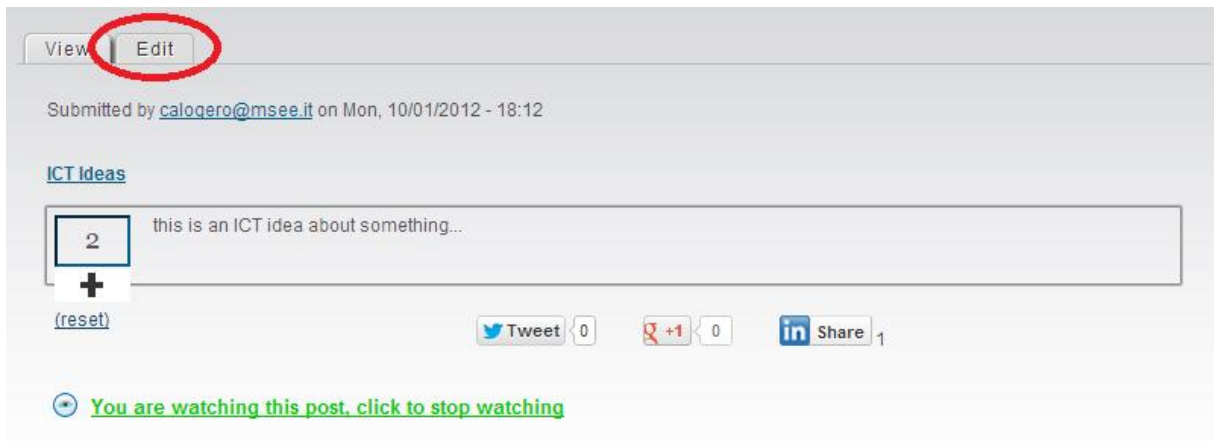


The screenshot shows the 'new MSEE Idea' page. At the top, it says 'Submitted by caloqero@msee.it on Tue, 10/02/2012 - 12:09'. Below this, there is a box with the number '2' and a plus sign, followed by the text 'this is my new MSEE idea'. To the right of this box, there are social media sharing buttons for Twitter, Google+, and LinkedIn. Below this, there is a green link that says 'You are watching this post, click to stop watching'. Below this, there is a section titled 'Add new comment'. In this section, there is a text input field with the text 'first comment!'. Below the input field, there is a 'Text format' dropdown menu set to 'Filtered HTML'. To the right of the dropdown, there is a link that says 'More information about text formats'. Below the dropdown, there are three bullet points: 'Web page addresses and e-mail addresses turn into links automatically.', 'Allowed HTML tags: <a> <cite> <blockquote> <code> <dl> <dt> <dd>', and 'Lines and paragraphs break automatically.'. At the bottom of the form, there is a 'Save' button.

Figure 23. Idea Comment form.

A reviewer can do some operations on existing ideas. It can review ideas or assign a workflow state to it. Idea possible states are *Under Review*, *Reviewed*, *Coming Soon*, *Launched* and *Next Time*.

To review an idea select it, go to the edit tab like in Figure 24, and select the *Revision Information* tab like in Figure 25. To save the review click the *Save* button, than you will see the idea revision page like in Figure 26.





Submitted by calogero@msee.it on Mon, 10/01/2012 - 18:12

ICT Ideas

2

this is an ICT idea about something...

(reset)

 Tweet 0
  +1 0
  Share 1


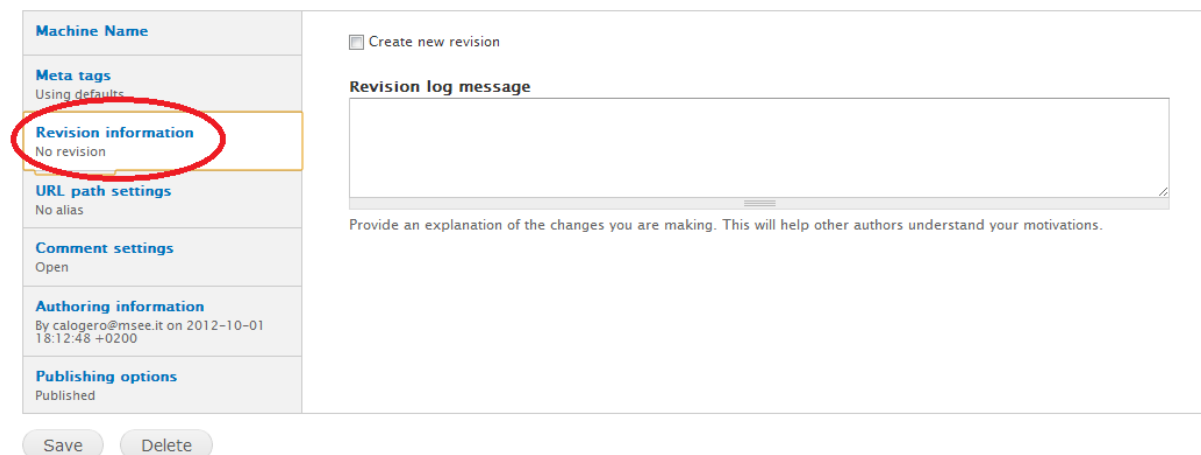
 You are watching this post, click to stop watching

Figure 24. Edit idea.



Machine Name

Meta tags
Using defaults

Revision information
No revision

URL path settings
No alias

Comment settings
Open

Authoring information
By calogero@msee.it on 2012-10-01 18:12:48 +0200

Publishing options
Published

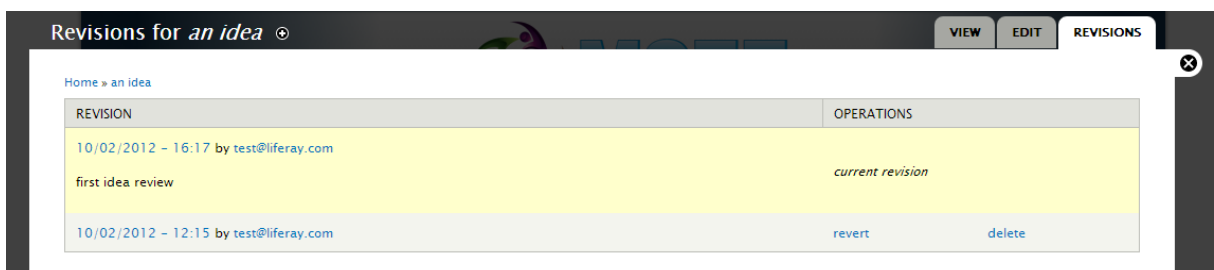
☐ Create new revision

Revision log message

Provide an explanation of the changes you are making. This will help other authors understand your motivations.

Save Delete

Figure 25. Review tab.



Revisions for <i>an idea</i>		VIEW	EDIT	REVISIONS
Home » <i>an idea</i>				
REVISION	OPERATIONS			
10/02/2012 - 16:17 by test@liferay.com				
first idea review	<i>current revision</i>			
10/02/2012 - 12:15 by test@liferay.com	revert	delete		

Figure 26. Idea Revision page.

To assign a category to an idea, edit the idea and select the category in the *Category* field, like in Figure 27.

Edit Idea new MSEE Idea +

[Home](#) » [new MSEE Idea](#)

Title *

new MSEE Idea

Category

- None -

Figure 27. Assign Category.

To analyze ideas, click the *Analytics Dashboard* menu link, then you will see the dashboard represented in Figure 28.

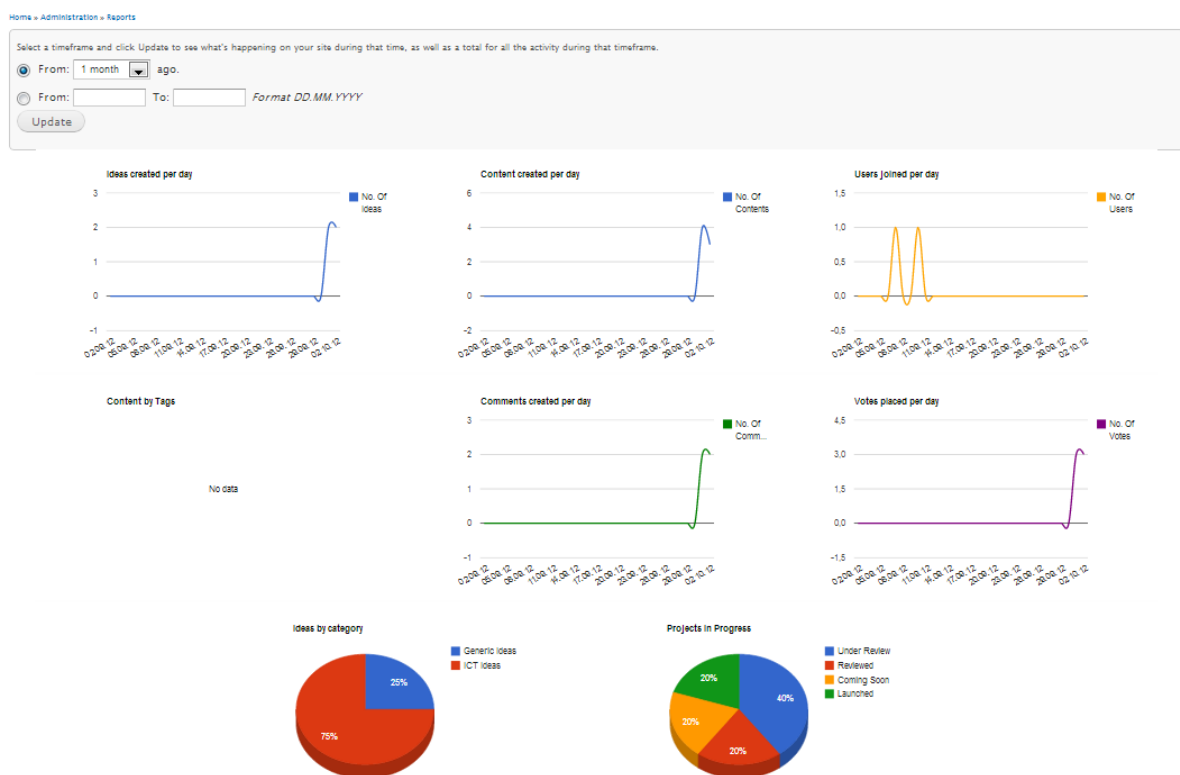


Figure 28. Analytics Dashboard.

To assign the state of an idea, click the *Ideas Management* menu link. The idea management page look like in Figure 29.

Title contains

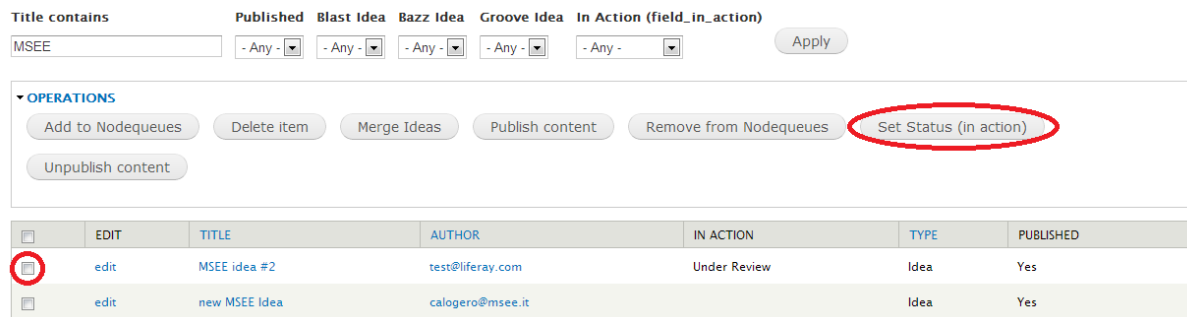
Published **Blast Idea** **Bazz Idea** **Groove Idea** **In Action (field_in_action)**

OPERATIONS

	EDIT	TITLE	AUTHOR	IN ACTION	TYPE	PUBLISHED
<input type="checkbox"/>	edit	MSEE idea #2	test@liferay.com	Under Review	Idea	Yes
<input type="checkbox"/>	edit	new MSEE Idea	calogero@msee.it		Idea	Yes

Figure 29. Ideas Management page.

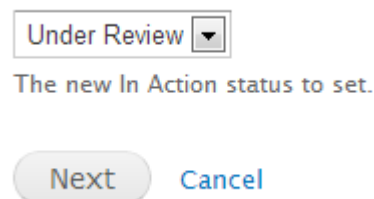
In this page it is possible to search and sort ideas, and to do some operations on ideas. To modify the state of an idea, select the idea and then click the *Set Status* button (see Figure 30).



The screenshot shows a search and filter interface at the top with tabs for 'Published', 'Blast Idea', 'Bazz Idea', 'Groove Idea', and 'In Action (field_in_action)'. Below these are dropdown menus for 'Title contains' (set to 'MSEE') and filters for each tab (all set to '- Any -'). An 'Apply' button is to the right. Below the filters is a section titled 'OPERATIONS' containing buttons: 'Add to Nodequeues', 'Delete item', 'Merge Ideas', 'Publish content', 'Remove from Nodequeues', 'Set Status (in action)' (circled in red), and 'Unpublish content'. At the bottom is a table with columns: EDIT, TITLE, AUTHOR, IN ACTION, TYPE, and PUBLISHED. The first row shows 'MSEE idea #2' by 'test@liferay.com' with status 'Under Review'. The second row shows 'new MSEE Idea' by 'calogero@msee.it' with status 'Idea'. A red circle highlights the 'edit' link in the first row.

Figure 30. Set idea state.

You will see the page represented in Figure 31.



The screenshot shows a 'Change State form' with a dropdown menu currently set to 'Under Review'. Below the dropdown is the text 'The new In Action status to set.' At the bottom are two buttons: 'Next' and 'Cancel'.

Figure 31. Change State form.

Select a new state and press the Next button, then press the *Confirm* button.

5.2.3 USDL Repository

The USDL Repository is currently implemented as an instance of the Sesame2 [15] RDF Store. Sesame provides two complementary ways to manage the resources. The first is a web-application (OpenRDF Workbench), which allows the user to explore the repositories, execute CRUD operations on their resources and execute SPARQL queries. A user manual for such web-application is out of scope for this document. A good user manual can be found at: <http://www.jenitennison.com/blog/node/153>.

The second and most important way to interact with the Sesame repository is through its HTTP-based communication protocol for programmatic access to the resources. A clear and exhaustive description of such protocol, based on SPARQL 1.1 Graph Store and SPARQL 1.1. Update endpoint is provided at <http://www.openrdf.org/doc/sesame2/system/ch08.html>. Both are based on the REST architectural style.

Here we provide a brief introduction to the way these protocols should be used to manage RDF descriptions of tangibles and intangibles and of the related services (LinkedUSDL models). According to what is described in MSEE D22.5, indeed, both in/tangible assets and (LinkedUSDL) models of the services that are based on such assets are represented as RDF statements.

We chose to manage both descriptions as named graph because such descriptions represent a coherent and cohesive set of statements that should be seen as an atomic entity in order to allow easy retrieval and CRUD operations.

Creation of a new description

Each description is stored as a named graph. The creation of a new graph is achieved through the SPARQL 1.1 Update endpoint provided by Sesame. The POST method is used to perform an update described in a SPARQL 1.1 Update string.

```
POST /openrdf-sesame/repositories/<repo_name>/statements HTTP/1.1
Host: localhost
Content-Type: application/ rdf+xml;charset=UTF-8
update= INSERT DATA { GRAPH <graph_name> { .. RDF payload .. } }
```

Fetching all statements from a description

Retrieving all the statements from a named graph is done through the SPARQL 1.1 Graph Store protocol. In particular the GET method is used.

```
GET /openrdf-sesame/repositories/<repo_name>/rdf-graphs/<graph_name> HTTP/1.1
Host: localhost
Accept: application/rdf+xml
```

Add statements to a description

Descriptions can be extended by adding new statements through the POST method of the SPARQL 1.1 Graph Store protocol.

```
POST /openrdf-sesame/repositories/<repo_name>/rdf-graphs/<graph_name> HTTP/1.1
Host: localhost
Content-Type: application/ rdf+xml;charset=UTF-8
[RDF/XML ENCODED RDF DATA]
```

Replace statements of a description

Statements of a descriptions can be replaced by a new set of statements through the PUT method of the SPARQL 1.1 Graph Store protocol.

```
PUT /openrdf-sesame/repositories/<repo_name>/rdf-graphs/<graph_name> HTTP/1.1
Host: localhost
Content-Type: application/ rdf+xml;charset=UTF-8
[RDF/XML ENCODED RDF DATA]
```

Delete a description

A descriptions can be deleted through the DELETE method of the SPARQL 1.1 Graph Store protocol.

```
DELETE /openrdf-sesame/repositories/<repo_name>/rdf-graphs/<graph_name> HTTP/1.1
Host: localhost
```

Query

Evaluation of queries on the repository is done *by sending requests to: <SESAME_URL>/repositories/<ID>.* This resource represents a SPARQL query endpoint. Both GET and POST methods are supported¹.

```
GET /openrdf-sesame/repositories/<repo_name>?query=<sparql_query> HTTP/1.1
Host: localhost
Accept: application/sparql-results+xml, */*;q=0.5
```

```
POST /openrdf-sesame/repositories/<repo_name> HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Accept: application/rdf+xml, */*;q=0.5
query=<sparql_query>
```

¹ <http://www.openrdf.org/doc/sesame2/system/ch08.html>

6 Future Plans

The Innovation Ecosystem Platform (IEP) is a complex environment composed by several integrated pieces of software. These components of this first prototype, called Core Components of the IEP, include all the functionalities that are directly related to management of the processes and to the integration issues. In particular the IEP Core Components allow the execution and management of the three types of services that characterize an ecosystem: MSE Innovation, MSE Governance and VME Operations. The IEP is developed as a web-portal that integrates functionalities to directly manage such processes and leverage virtualized and servitized (in)tangible assets stored in a RDF repository.

In the second cycle of MSEE the Core Components will be evolved according to the refined necessities of the MSEE Use Cases (where the Core Components will be tested) and will be complemented by other components that will allow the management of demand and offer of (in)tangible assets as services and by a dedicated tool to help enterprises assess their maturity level and manage the change process.

7 References

- [1] CAS: <http://www.jasig.org/cas>
- [2] Aperte Workflow: <http://www.aperteworkflow.org/>
- [3] Open Ideal: <http://www.openidealapp.com/>
- [4] Mule ESB: <http://www.mulesoft.org/>
- [5] Liferay: <http://www.liferay.com/>
- [6] Vaadin: <https://vaadin.com/home>
- [7] OSGi: <http://www.osgi.org/>
- [8] Hibernate: <http://www.hibernate.org/>
- [9] Activiti: <http://activiti.org/>
- [10] Signavio Core Components: <http://code.google.com/p/signavio-core-components/>
- [11] Jbpm: <http://www.jboss.org/jbpm>
- [12] phpCAS: <https://wiki.jasig.org/display/CASC/phpCAS>
- [13] CAS (Drupal): <http://drupal.org/project/cas>
- [14] Drupal: <http://drupal.org/>
- [15] <http://www.openrdf.org/>