


Project ID 284860	MSEE – Manufacturing Services Ecosystem	 MSEE Manufacturing Service Ecosystem
Date: 11/07/2013	Deliverable D43.2 – Service Delivery Infrastructure specifications and architecture – M21	



*D43.2 –
Service Delivery Infrastructure specifications
and architecture –
M21*

Document Owner:	José María García (UIBK), Iker Larizgoitia (UIBK), Ioan Toma (UIBK)
Contributors:	Martino Maggio (ENG), Davide Storelli (ENG), Giovanni Castella (SOFTECO), Enrico Morten (SOFTECO)
Dissemination:	Public
Contributing to:	WP 43
Date:	11/07/2013
Revision:	3.0

VERSION HISTORY

NUMBER	DATE	NOTES AND COMMENTS
0.5	01/06/2013	FIRST DRAFT
1.0	01/07/2013	CONTRIBUTIONS ADDED READY FOR INTERNAL REVIEW
2.0	05/07/2013	UPDATED TO ADDRESS PEER REVIEW COMMENTS.
3.0	11/07/2013	ROADMAP SECTION ADDED

DELIVERABLE PEER REVIEW SUMMARY

ID	Comments	Addressed (✓) Answered (A)
1	Add more scenario based descriptions in Section 4.2.7.2	Added a description to the screenshots, based on an scenario from the MSEE use cases.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	5
1 INTRODUCTION	6
1.1 <i>CONTEXT AND PURPOSE OF THE DELIVERABLE</i>	6
1.2 <i>RELATION TO OTHER WORK PACKAGES AND DELIVERABLES</i>	6
1.3 <i>STRUCTURE OF THE DOCUMENT</i>	6
2 SERVICE DELIVERY PLATFORM (SDP) SERVICE MODEL	7
3 REQUIREMENTS DEFINITION FOR MSEE SERVICE DELIVERY PLATFORM	9
4 SERVICE DELIVERY PLATFORM ARCHITECTURE	10
4.1 <i>GLOBAL COMPONENT VIEW</i>	10
4.2 <i>COMPONENT AND BEHAVIORAL VIEWS OF EACH MODULE</i>	11
4.2.1 <i>Repository</i>	11
4.2.2 <i>Registry</i>	11
4.2.2.1 <i>Component diagram</i>	11
4.2.2.2 <i>Sequence diagrams</i>	12
4.2.3 <i>Discovery</i>	13
4.2.3.1 <i>Component diagram</i>	13
4.2.3.2 <i>Sequence diagram</i>	14
4.2.4 <i>Ranking</i>	15
4.2.4.1 <i>Component diagram</i>	15
4.2.4.2 <i>Sequence diagram</i>	15
4.2.5 <i>Invocation</i>	16
4.2.5.1 <i>Component diagram</i>	16
4.2.5.2 <i>Sequence diagrams</i>	17
4.2.6 <i>Monitoring</i>	17
4.2.6.1 <i>Component diagram</i>	18
4.2.6.2 <i>Sequence diagram</i>	18
4.2.7 <i>Service Delivery Platform Dashboard</i>	19
4.2.7.1 <i>Component diagram</i>	19
4.2.7.2 <i>Screenshots by functionality</i>	20
5 SERVICE DELIVERY PLATFORM INTEGRATION	24
5.1 <i>INTEGRATION WITH THE SERVICE DEVELOPMENT PLATFORM</i>	24
5.2 <i>INTEGRATION WITH THE MOBILE PLATFORM</i>	24
5.3 <i>INTEGRATION WITH THE INNOVATION ECOSYSTEM PLATFORM</i>	25
5.4 <i>INTEGRATION WITH THE FEDERATED SINGLE SIGN ON UTILITY SERVICE</i>	25
5.5 <i>INTEGRATION WITH THE PIR - PROCESS INDICATOR REGISTRY UTILITY SERVICE</i>	25
6 ROADMAP	26
7 CONCLUSIONS	27

Figures

Figure 1.- Semantic Service Description simplified example showing which components exploit the different parts.....	8
Figure 2.- Service Delivery Platform component view	10
Figure 3.- Repository component diagram	11
Figure 4.- Registry component diagram	12
Figure 5.- Sequence diagram for a WSDL service registration.....	12
Figure 6.- Sequence diagram for a REST service registration	13
Figure 7.- Discovery component diagram	14
Figure 8.- Sequence diagram for service discovery	14
Figure 9.- Ranking component diagram	15
Figure 10.- Ranking sequence diagram	16
Figure 11.- Invocation component diagram	16
Figure 12.- Invocation sequence diagrams for WSDL and RESTful services	17
Figure 13.- Monitoring component diagram	18
Figure 14.- Monitoring sequence diagram	18
Figure 15.- Dashboard component diagram	19
Figure 16.- SDP Dashboard.....	20
Figure 17.- Login to the dashboard	20
Figure 18.- Registration dashboard	21
Figure 19.- Successful registration through the dashboard	21
Figure 20.- Discovery dashboard	21
Figure 21.- Discovery category	22
Figure 22.- Discovery category result	22
Figure 23.- Invocation dashboard.....	23
Figure 24.- Invocation result	23
Figure 25.- Integration of the Service Delivery Platform.....	24

Executive Summary

The transition from product-centric offerings to a value-proposition based on product-service bundles is a hard challenge for European manufacturing enterprises, which are in need of models, methods and tools to face the servitization process through a structured approach in order to mitigate the risks and maximize the opportunities. Sub Project 4 (SP4) of the MSEE project aims at designing and developing an integrated IT system, which will support the servitization process through IT tools, specifically to develop, operate and govern the information technology parts of the target service system.

WP4.3 aims at designing the MSEE Service Delivery Platform (SDP). Rather than being a monolithic platform, the MSEE SDP will be realized as a set of decoupled, standalone services, which offer core service usage tasks (such as discovery, ranking, grounding, invocation, etc.). The MSEE Service Delivery Platform will work on top of lightweight semantic descriptions, and will be able to deal with different service technologies, i.e. traditional SOAP-based services as well as RESTful services. This deliverable D43.2 updates the previous version of the SDP specifications and architecture regarding introduced changes with respect to D43.1.

After an introduction to the scope and objectives of the deliverable given in Section 1, Section 2 describes the Service Model that will be used in MSEE SDP. The model is inherited from work on light weight service descriptions, such as MSM.

Section 3 is about how the functional and non-functional requirements of the MSEE SDP have been fulfilled and extended after the feedback received during the last period.

Section 4 describes the MSEE SDP design. It provides at this stage of the project the final architecture description, an overview of each one of the modules present in the architecture and a more detailed specification of the different modules needed in the MSEE SDP, including components and sequence diagrams of all the relevant modules.

Section 5 elaborates on one important point for all the platforms to be developed as part of MSEE, namely the integration points. The MSEE SDP is going to actively communicate and provide its services to the MSEE Generic Service Development Platform, the MSEE Generic Mobile Business Platform and the MSEE Innovation Ecosystem Platform. In this section the integration strategy is outlined.

Finally, the document is closed presenting a roadmap for the remaining tasks for the final prototype implementation, and then discussing some conclusions about the outcome of the work reflected in this deliverable.

1 Introduction

1.1 Context and Purpose of the Deliverable

WP43 is designing and developing the MSEE Service Delivery Infrastructure that supports various service related tasks such as discovery, ranking, invocation and monitoring, for both WSDL and RESTful services. The MSEE Service Delivery Infrastructure is realized as the MSEE Service Delivery Platform (SDP), a non-monolithic platform composed of decoupled services supporting the service tasks mentioned before. The first version of the SDP specification, architecture and implementation was delivered as part of D43.1, and D43.3 respectively. Based on the feedback received from using the SDP platform as part of the various use case, the current deliverable defines the second version of the specification and architecture of the MSEE Service Delivery Platform.

This deliverable focuses on the requirements of the Service Delivery Platform, as well as the design of the functionality of each module of it. The initial SDP requirements were revisited in the light of feedback received during the past nine month and new requirements were added.

The result of this deliverable is a set of requirements and design guidelines that update and complement the first specification. They serve as a guide for the development of the second and last prototype of the SDP. Special attention is given also to the integration of the other platforms to be developed in MSEE (e.g. the Service Development Platform, the Mobile Platform and the Innovative Ecosystems Platform).

1.2 Relation to other Work Packages and Deliverables

As the SDP platform will be interconnected to other platforms developed in MSEE, the work done in other Work Packages and Deliverables are to be taken into account. Especially for WP43, the work done in WP42 - *MSEE Generic Service Development Platform* and WP44 - *MSEE Generic Mobile Business Platform* is of special importance from the technical point of view. In regard to the requirements, MSEE is use case driven, therefore the work of WP5.2 reflected in D52.2 – *User Requirements Analysis for Virtual Factories & Enterprises in MSEE*, focused on user requirements, has also been considered.

1.3 Structure of the Document

This deliverable is structured as follows. Section 2 revisits the semantic model used in the SDP platform. Section 3 focuses on the fulfilment of the requirements of the SDP platform. Section 4 contains the actual design, specification and architecture of the revised SDP as a whole, and each of its modules. Section 5 presents the integration points with other platforms in MSEE. Section 6 discusses the roadmap that will be followed for the final prototype implementation. Finally Section 7 concludes the deliverable.

2 Service Delivery Platform (SDP) Service Model

As it was defined in the previous deliverable of the design of the Delivery Platform (D43.1) the underlying semantic model is based on MSM, which has allowed us to model both REST and WSDL based services. The model has not been changed for the actual release, it has however be combined with additional information covering aspects needed for the use cases and also for the monitoring information (see Figure 1 for an example).

We can summarize the use of semantics by the different components in the following way:

- **Discovery:** the discovery module take primarily advantage of the semantic descriptions to enable search by categories and properties. The MSM semantic models define the categories the service belongs to, which can be exploited during discovery. The MSM is versatile enough to add additional properties to the service from other vocabularies, which can be used as well for the discovery.
- **Registration:** registration of services is done based on the descriptions of the interfaces, either WSDL with annotations (SAWSDL) or REST descriptions with annotations (hREST). From this annotations an MSM model is automatically created to be used in the Delivery Platform.
- **Monitoring & Ranking:** quality of service attributes are modelled as an ontology as well, and when monitoring is enabled, the measurements are attached to the registered services descriptions to enrich their semantic descriptions, enabling the usage of them in both ranking and discovery.
- **Invocation:** the invocation through the Delivery Platform using the semantic descriptions enables the separation of the actual information from the underlying invocation technology (in our case REST or WSDL). The model defines the correspondent schemas for lifting and lowering and how to call the actual service.

```

@prefix hr: <http://www.wsmo.org/ns/hrests#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#> .
@prefix ex: <http://cms-wg.sti2.org/TR/d12/v0.1/20081202/xslt/example.xhtml#> .
@prefix msee:<http://www.msee-ip.eu/properties#> .
@prefix indesit:<http://www.indesit.it#> .
@prefix msm:<http://cms-wg.sti2.org/ns/minimal-service-model#> .
@prefix msee-qos:<http://www.msee-ip.eu/qos#> .

indesit:CarefreeService a msm:Service ;
  rdfs:isDefinedBy <http://www.msee-ip.eu/cws/api.xhtml#> ;
  rdfs:label "Indesit Carefree REST Service" ;

  msee:description "Provides data on customers washing machines." ;
  msee:requirements "REST client" ;
  msee:icon <http://www.msee-ip.eu/images/Rest_icon.jpg> ;
  msee:access_point <http://www.msee-ip.eu/cws> ;

  sawsdl:modelReference <http://msee.sti2.at/categories#business> ;
  sawsdl:modelReference <http://msee.sti2.at/categories#Rest> ;
  sawsdl:modelReference <http://msee.sti2.at/categories#Indesit> ;

  msee-qos:hasQoSParameter [
    a msee-qos:QoSParameter ;
    msee-qos:hasQoSValue "620.0" ;
    msee-qos:hasQoSType msee-qos:UnavailableTime ;
    msee-qos:hasDateTime "2013-06-26T14:44:38+0200^^xsd:dateTime" ;
  ] ;

  msm:hasOperation indesit:OperationGetWashingMachines .

indesit:OperationGetWashingMachines a msm:Operation;
  hr:hasMethod "GET" ;
  hr:hasAddress "http://www.msee-ip.eu/cws/{customerid}"^^hr:URITemplate ;

  msm:hasInput [
    a msm:MessageContent ;
    sawsdl:modelReference <http://www.msee-ip.eu/cws/carefree.owl#Customer> ;
    sawsdl:loweringSchemaMapping <http://www.msee-ip.eu/cws/customer.xsparql>
  ] ;
  msm:hasOutput [ a msm:MessageContent ] .

```

Annotated description (REGISTRATION)

Additional properties (DISCOVERY)

Categories (DISCOVERY)

QoS Parameters (MONITORING & RANKING)

Interface description (INVOCATION)

Figure 1.- Semantic Service Description simplified example showing which components exploit the different parts.

3 Requirements definition for MSEE Service Delivery Platform

This section discusses the fulfilment of the original set of requirements identified in deliverable D43.1. Table 1.- Requirements fulfilment matrix for SDP Table 1 presents a fulfilment matrix for each requirement according to the current implementation of the SDP. Noteworthy, we have added a new non-functional requirement that was not originally identified in D43.1, but we had to account for specifying the architecture of the final version of the SDP, which is described in this deliverable. This new requirement is described as follows:

Req.8: The MSEE Service Delivery Platform MUST support both WSDL-based and REST-based services.

For a full description of the rest of the requirements see deliverable D43.1.

Table 1.- Requirements fulfilment matrix for SDP

<i>Requirement</i>	<i>Importance</i>	<i>Fulfilment</i>
<i>Func. Req. 1: Store semantic description of services</i>	Required	Yes
<i>Func. Req. 2: Retrieve semantic description of services</i>	Required	Yes
<i>Func. Req. 3: Delete semantic description of services</i>	Required	Yes
<i>Func. Req. 4: Update semantic description of services</i>	Required	Yes
<i>Func. Req. 5: Store ontologies for description of services</i>	Required	Yes
<i>Func. Req. 6: Retrieve ontologies for description of services</i>	Required	Yes
<i>Func. Req. 7: Delete ontologies for description of services</i>	Required	Yes
<i>Func. Req. 8: Update ontologies for description of services</i>	Required	Yes
<i>Func. Req. 9: Register services</i>	Required	Partial
<i>Func. Req. 10: Unregister services</i>	Required	Partial
<i>Func. Req. 11: Discover services</i>	Required	Partial
<i>Func. Req. 12: Ranking services</i>	Required	Partial
<i>Func. Req. 13: Select services</i>	Required	Partial
<i>Func. Req. 14: Invoke services</i>	Recommended	Partial
<i>Func. Req. 15: Monitor services</i>	Required	Partial
<i>Non-func. Req. 1: Configuration and management of SDP</i>	Recommended	Partial
<i>Non-func. Req. 2: Test functionalities</i>	Recommended	Yes
<i>Non-func. Req. 3: Available to others MSEE platforms</i>	Recommended	Yes
<i>Non-func. Req. 4: Secure access</i>	Optional	Yes
<i>Non-func. Req. 5: Extendable functionalities</i>	Required	Yes
<i>Non-func. Req. 6: Ease of use</i>	Recommended	Partial
<i>Non-func. Req. 7: Scalability</i>	Required	Yes
<i>Non-func. Req. 8: WSDL and REST services support</i>	Required	Partial

Analysing Table 1, we can conclude that the current design of the MSEE SDP mostly achieved all the original requirements. However, there is an ongoing effort on fulfilling the new requirement on the support of RESTful services. Consequently, functional requirements 9 to 15 are marked in Table 1 as *partially* fulfilled, as the SDP fully supports WSDL-based services but REST-based services support is being implemented. These requirements, as well as non-functional requirements 1, 6, and 8, will be completely fulfilled for the final prototype implementation to be described in deliverable D43.4. A detailed roadmap is discussed in Section 6.

4 Service Delivery Platform Architecture

The Service Delivery Platform (SDP) is one of the platforms designed in MSEE to provide the needed infrastructure for working with Web services. The SDP provides the technical foundations for the discovery, ranking, invocation and registration of Web Services, with both WSDL-based and REST-based underlying approaches.

We adopt a service view for both services and applications in MSEE. Applications and services are both seen as services and therefore modelled as such. In the following we present the updated component views of the different infrastructure services of the MSEE, with respect to the first version of the specification described in deliverable D43.1, including a more detailed overview of each module of the SDP, in terms of dependencies with other modules and the behavioural views of the main functionalities.

4.1 Global component view

The SDP is provided as a set of decoupled infrastructure services that can be accessed through standard APIs, both internally and from the other platforms conforming MSEE.

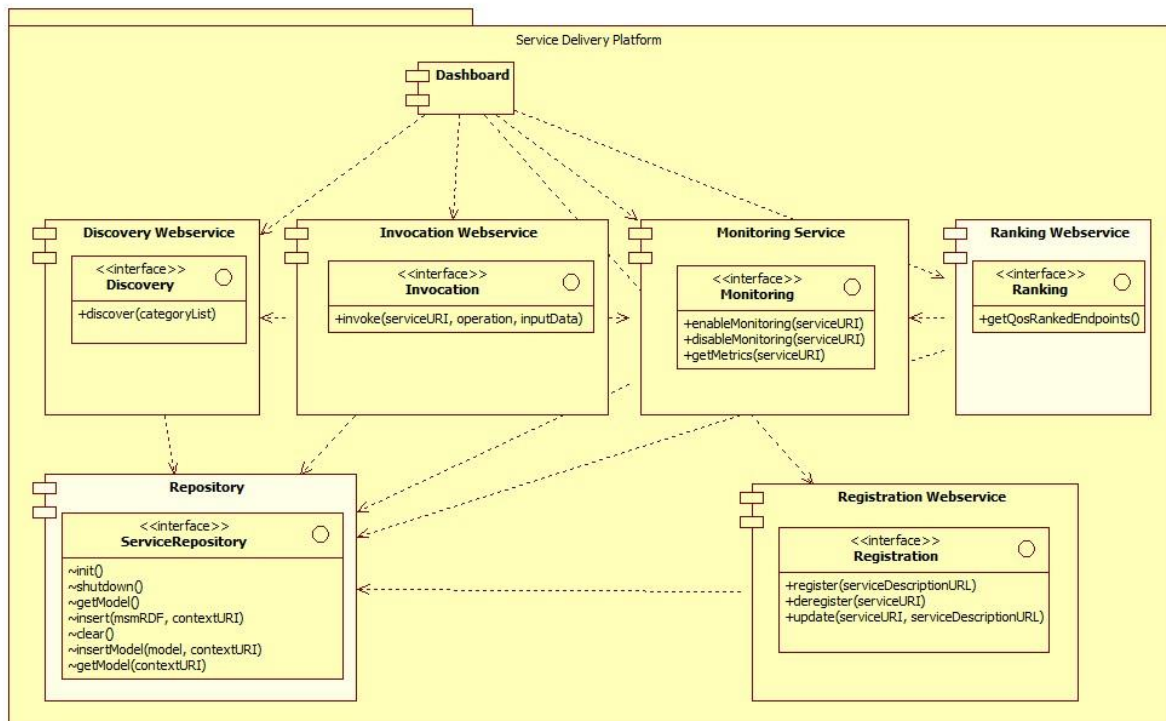


Figure 2.- Service Delivery Platform component view

The figure above illustrates the modules of the MSEE Service Delivery Platform as a component diagram with the provided interfaces. It also includes the dependencies between the components. The ServiceRepository component is not publicly usable, but its functionality gets used by all other components. More information about the specific interfaces and operations can be found at deliverable D43.1.

4.2 Component and behavioral views of each module

4.2.1 Repository

The *repository* module provides semantic storage capabilities inside the Service Delivery Platform. It acts as a generic storage module to be used by the rest of the components; therefore it has to provide a standard interface to store/retrieve semantic information.

Figure 3 presents the main components of the repository module. The core component, which provides the needed functionality as an interface to access the underlying triple store, relies on RDF2Go to abstract the access to the data layer, which consists in an OpenRDF / Sesame repository.

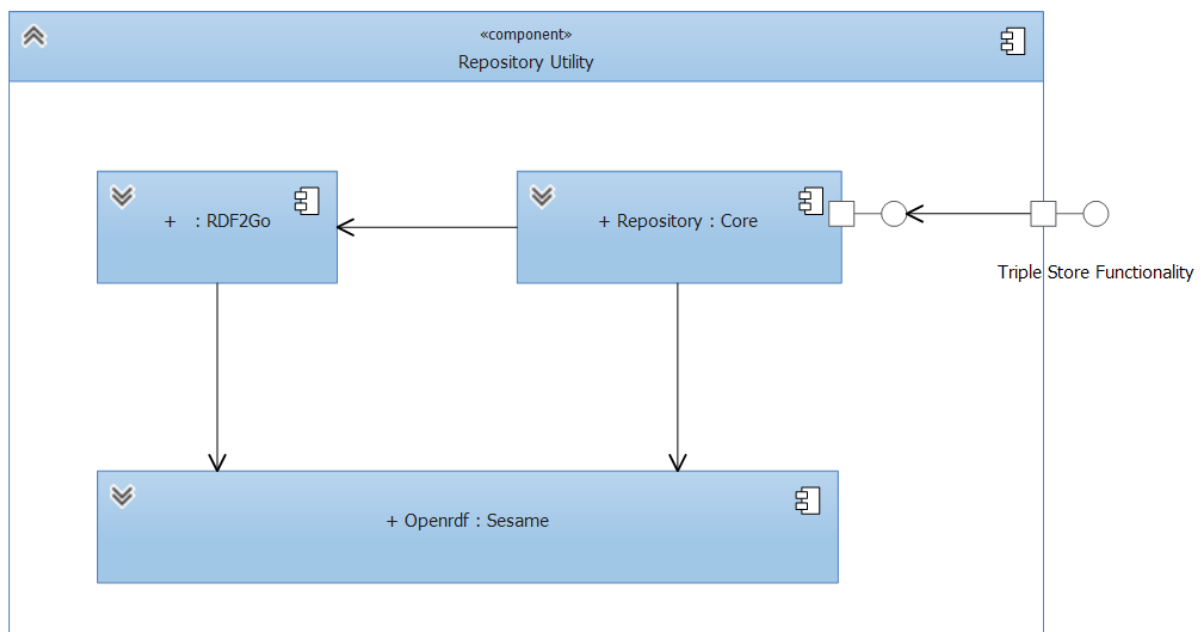


Figure 3.- Repository component diagram

4.2.2 Registry

The registry module provides the functionality required to manage service descriptions in the SDP. The SDP platform allows the registration of semantic service descriptions (as WSDL/SAWSDL, MSM/WSMO-Lite, or hRESTs) in order to make them available for discovery, ranking, invocation or monitoring.

4.2.2.1 Component diagram

The registration functionality provided by this module involves several components shown in Figure 4. The module basically offers its functionality as a web service that can be consumed by other MSEE platforms. It relies both on the repository module and also on iServe service repository functionality for importing WSDL- and MSM-based services.

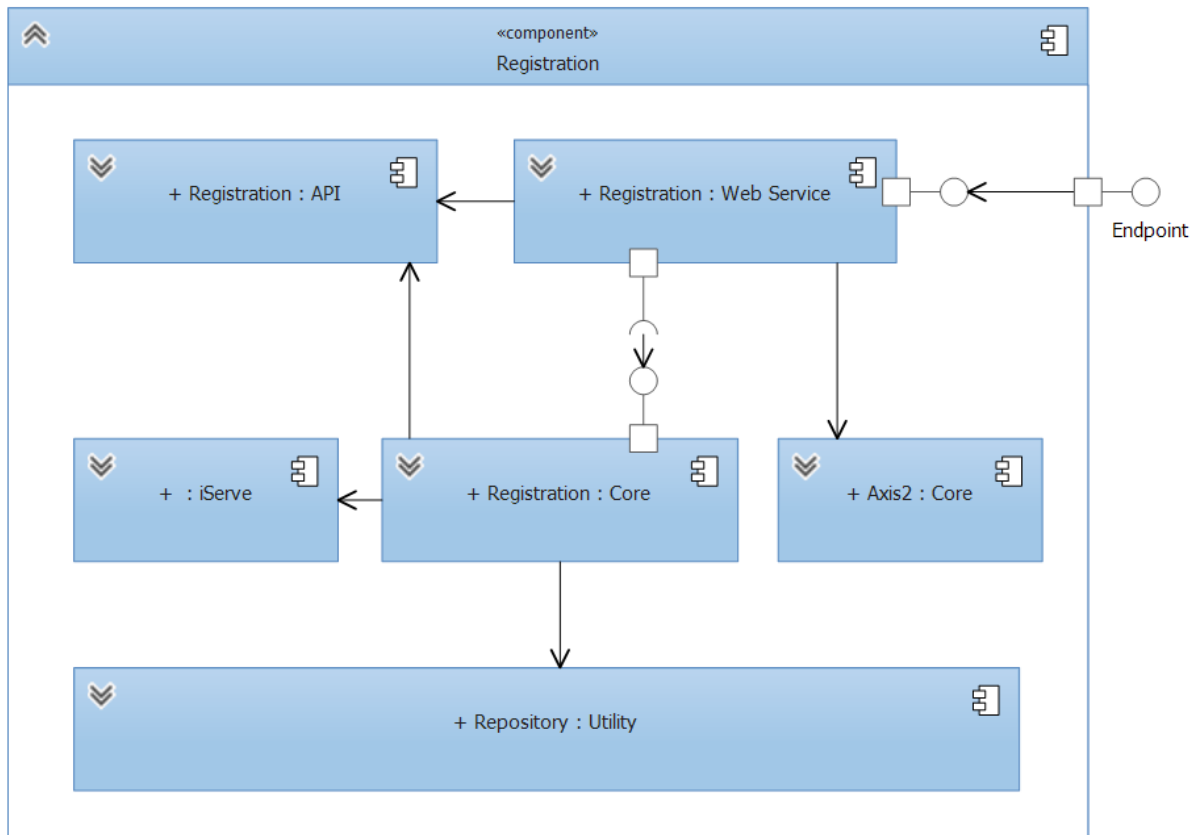


Figure 4.- Registry component diagram

4.2.2.2 Sequence diagrams

For the sequence diagrams, we split the main registration functionality in two diagrams, namely Figure 5 and Figure 6, depending on whether the service is WSDL- or REST-based, correspondingly.

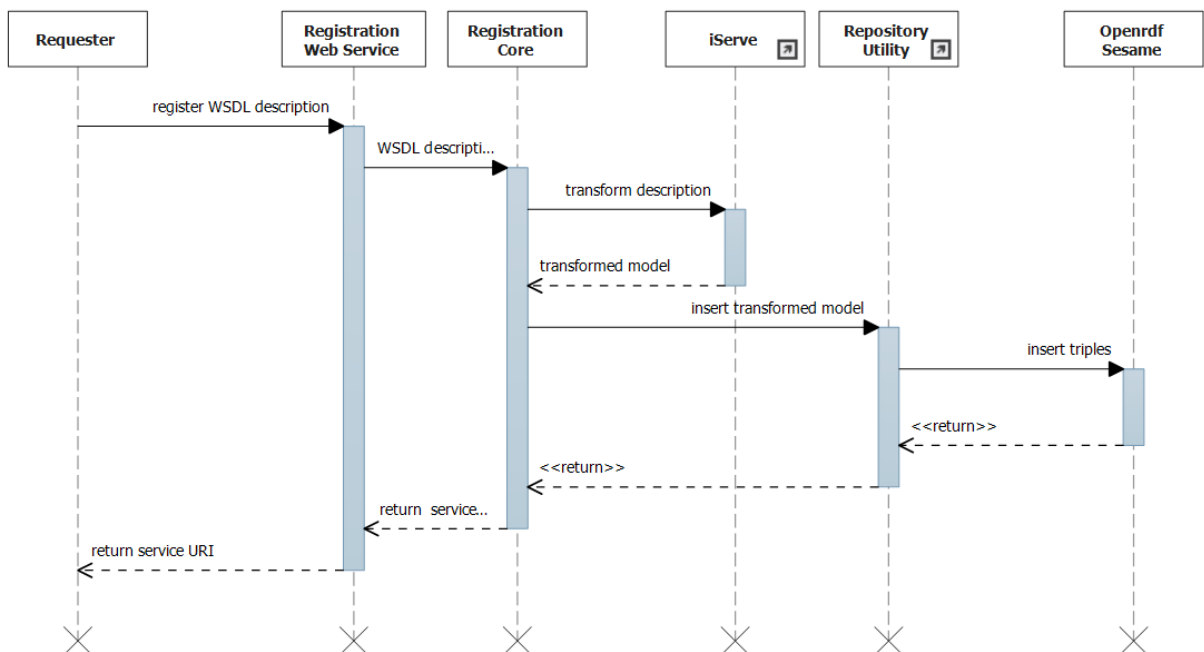


Figure 5.- Sequence diagram for a WSDL service registration

The main difference is on the use of iServe functionality to transform WSDL-based descriptions, which is not needed for REST-based services as this transformation is performed by the registration core component.

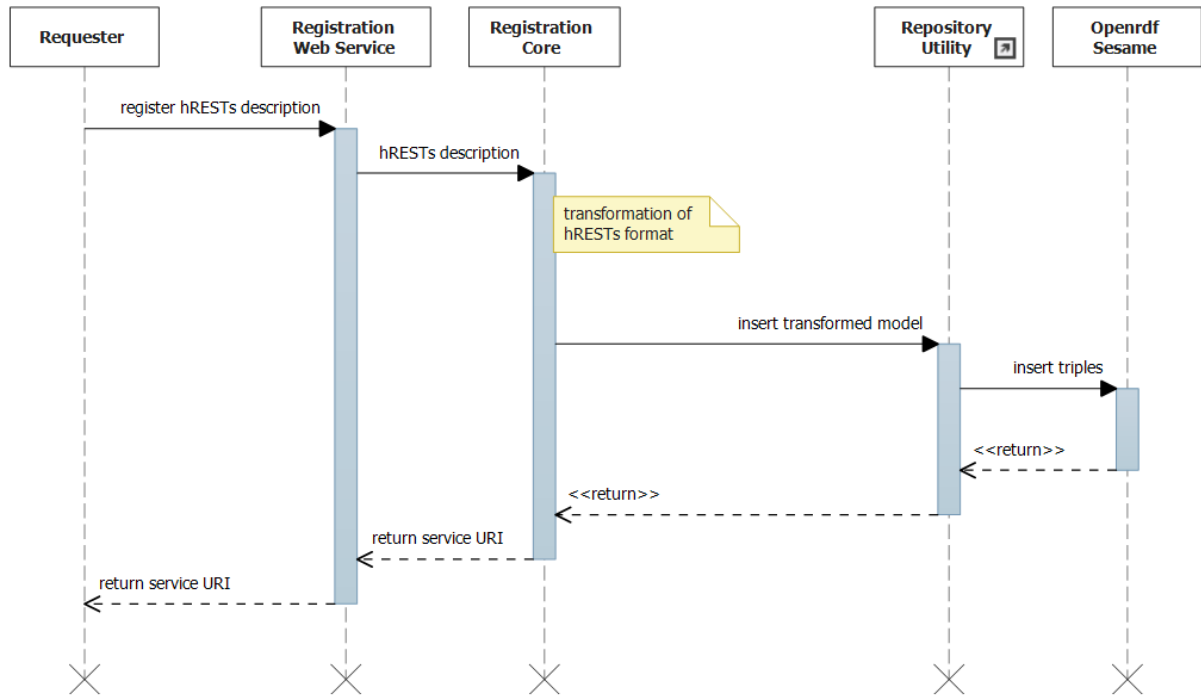


Figure 6.- Sequence diagram for a REST service registration

4.2.3 Discovery

The discovery module exploits the semantic service descriptions previously registered in the SDP platform to provide advanced search features based on semantics. Using the semantic descriptions many discovery strategies can be applied. The discovery module in the SDP will provide classification of categories and properties, leaving room for advanced queries using a standard semantic query language like SPARQL.

4.2.3.1 Component diagram

In the case of the discovery module, the involved components are only the repository and the web service interface infrastructure provided by the Axis2 library, as showcased in Figure 7.

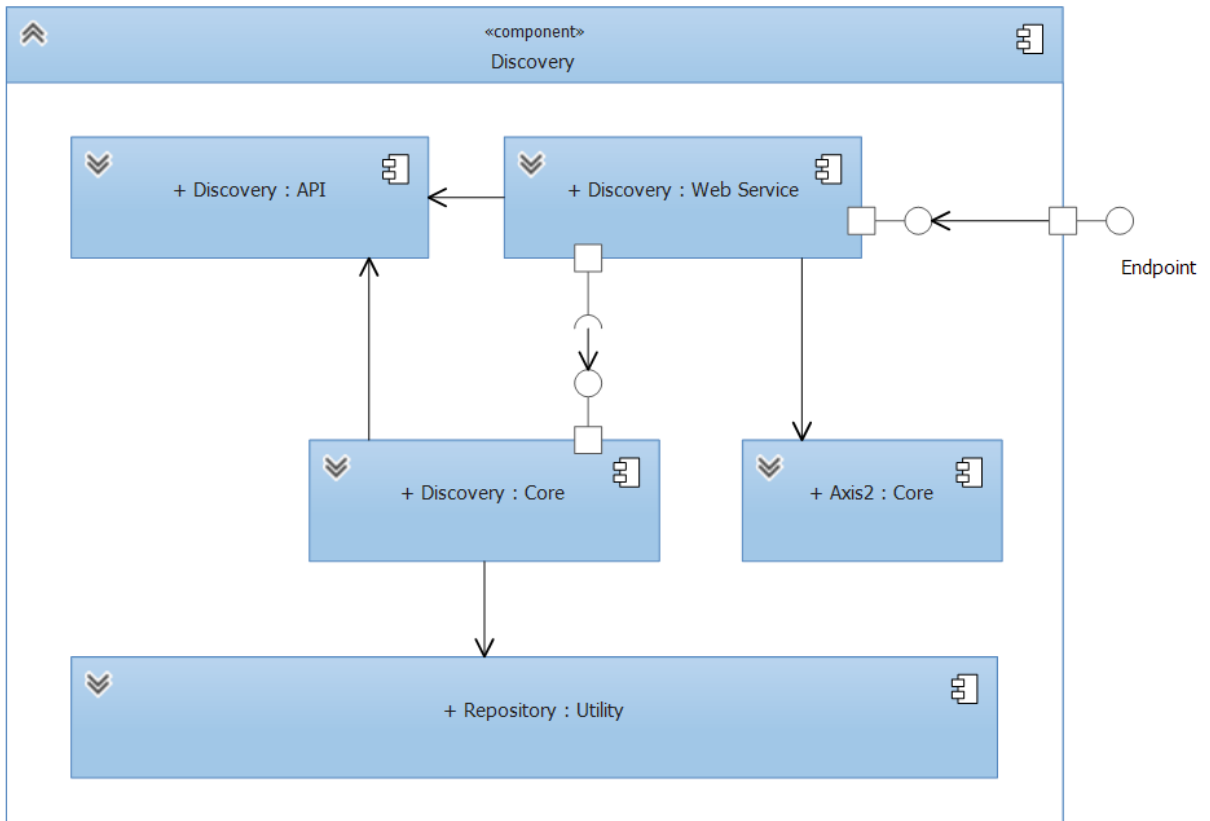


Figure 7.- Discovery component diagram

4.2.3.2 Sequence diagram

The discovery module makes no distinction between WSDL- and REST-based registered services when searching for appropriate services on the input category lists. Therefore, its functionality is simply shown in Figure 8.

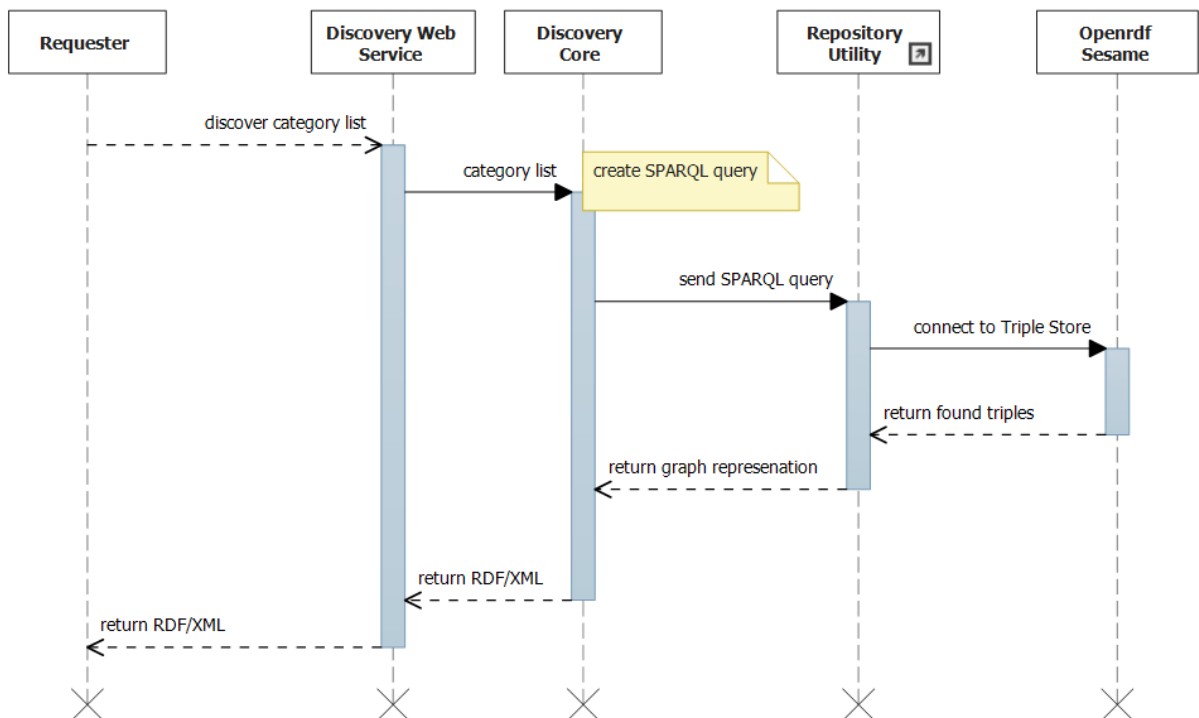


Figure 8.- Sequence diagram for service discovery

4.2.4 Ranking

The Ranking module is used to generate an ordered list of services out of the candidate services provided by the discovery module. As ranking criteria, various non-functional properties such as Service Level Agreements (SLA), Quality of Services (QoS), etc. can be included in the ranking algorithm. In the SDP, the ranking is carried out based on the properties specified in the discovery goal and the service descriptions by means of logical rules.

4.2.4.1 Component diagram

Although the list of services to be ranked are usually obtained from a previous call to discovery functionality, the ranking module is not coupled with the discovery components of the SDP, so they can be used in different scenarios. Figure 9 presents the component view of this module, where functionality from the SOA4All multi-criteria ranking component is being used to rank services.

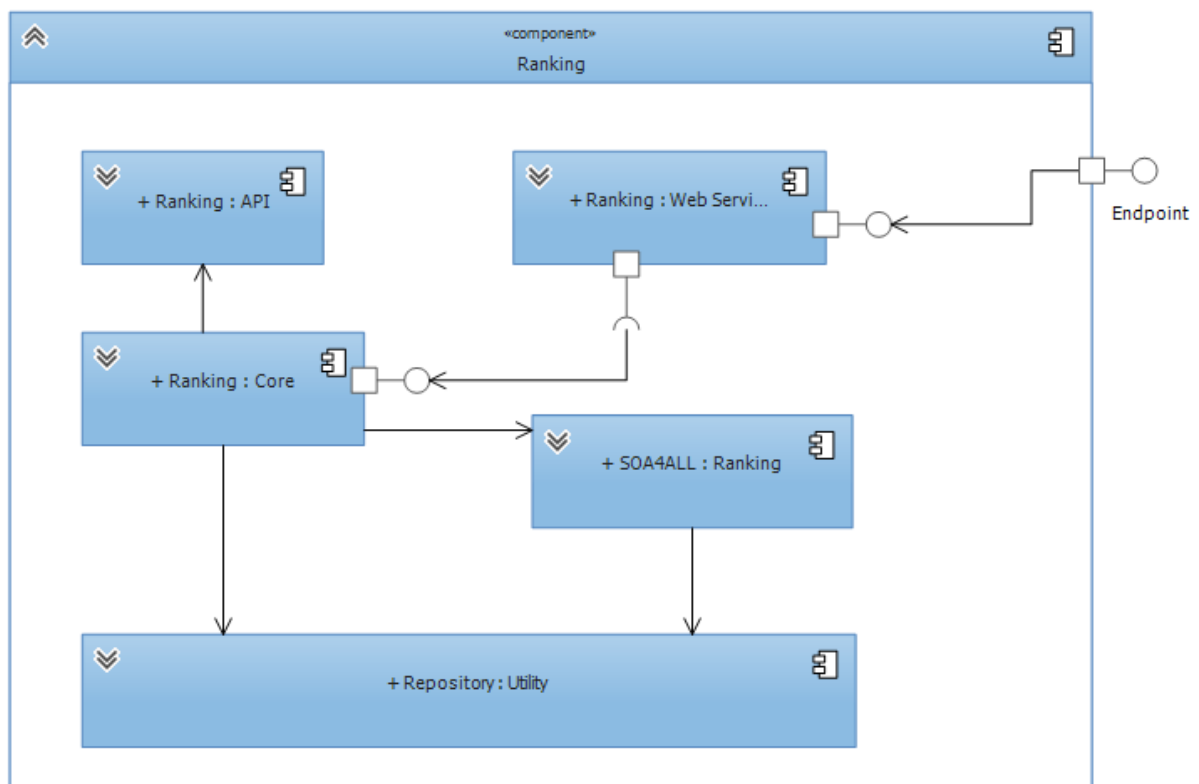


Figure 9.- Ranking component diagram

4.2.4.2 Sequence diagram

Again, the ranking functionality does not differentiate between the nature of service descriptions, so we show a single sequence diagram for the ranking module in Figure 10.

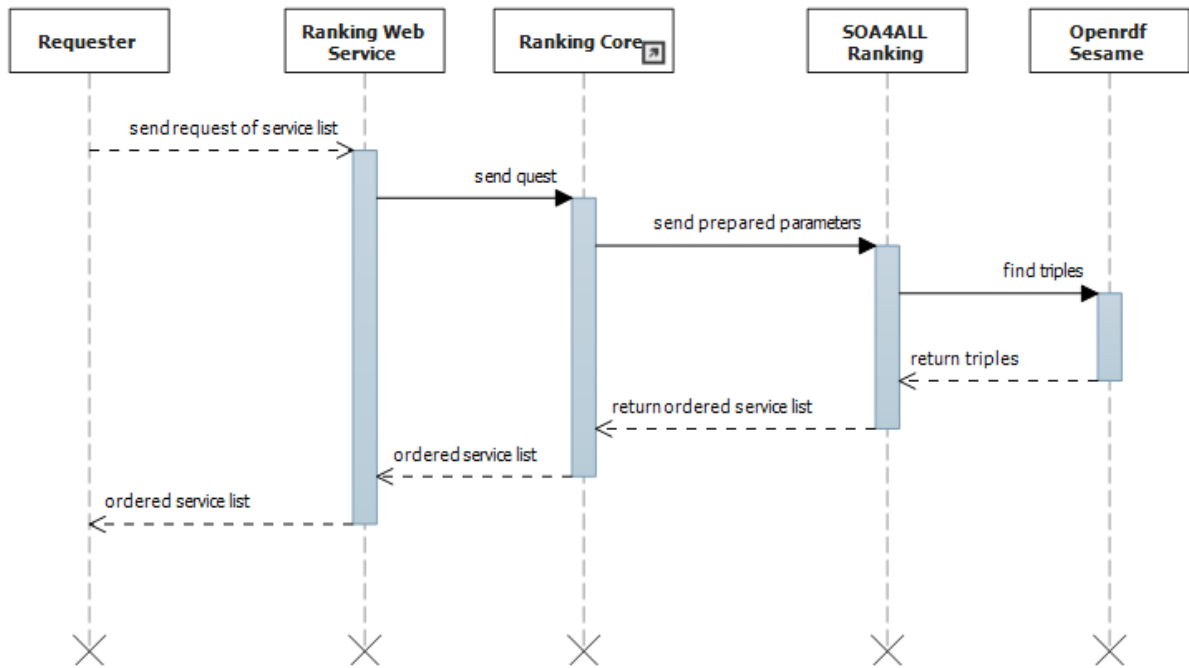


Figure 10.- Ranking sequence diagram

4.2.5 Invocation

The invocation module enables external components to invoke the registered services without having to deal with the specificities of the invocation protocol. This means, the invocation module acts as a transparent proxy for the invocation of services.

4.2.5.1 Component diagram

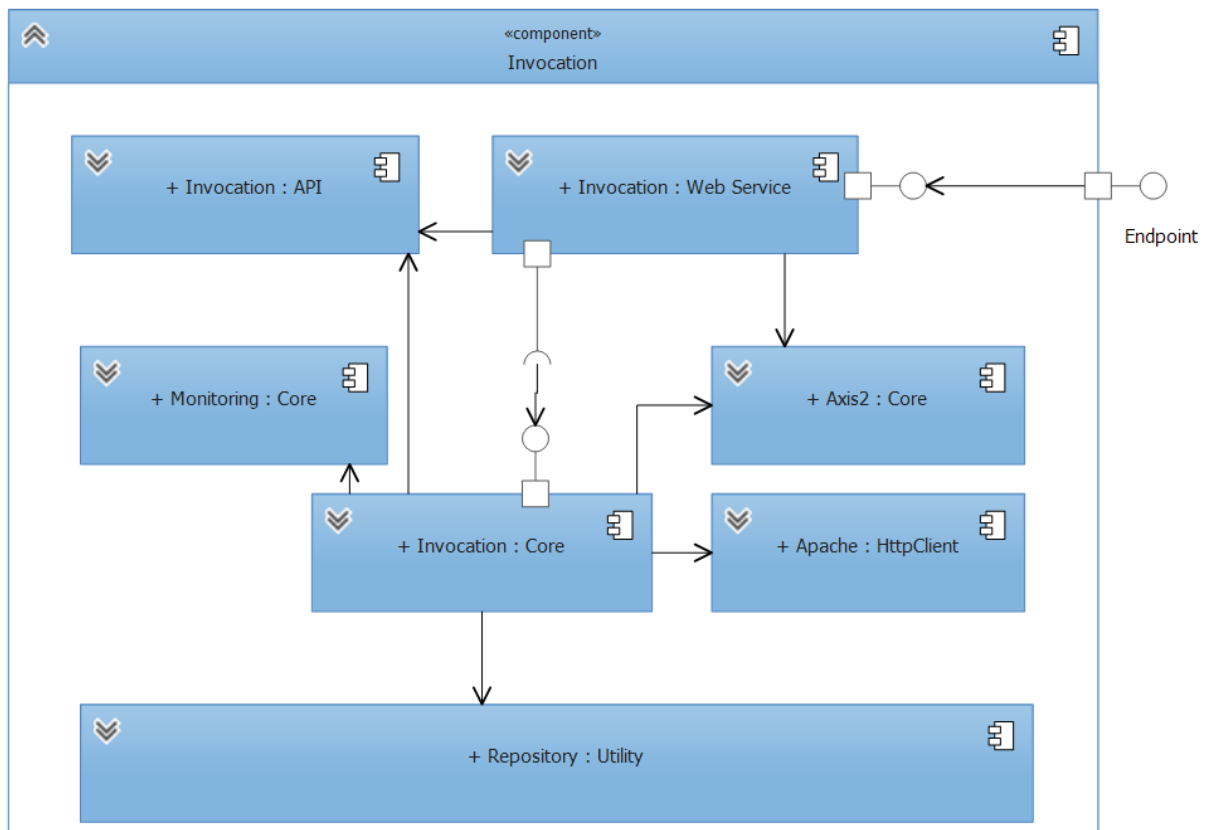


Figure 11.- Invocation component diagram

Figure 11 presents the corresponding component diagram. Depending on the nature of invoked services, the invocation module relies on the Axis2 library (WSDL-based services) or Apache HttpClient component (REST-based ones). The monitoring module is used by the invocation module. It retrieves all data for monitoring based on the invocation of services.

4.2.5.2 Sequence diagrams

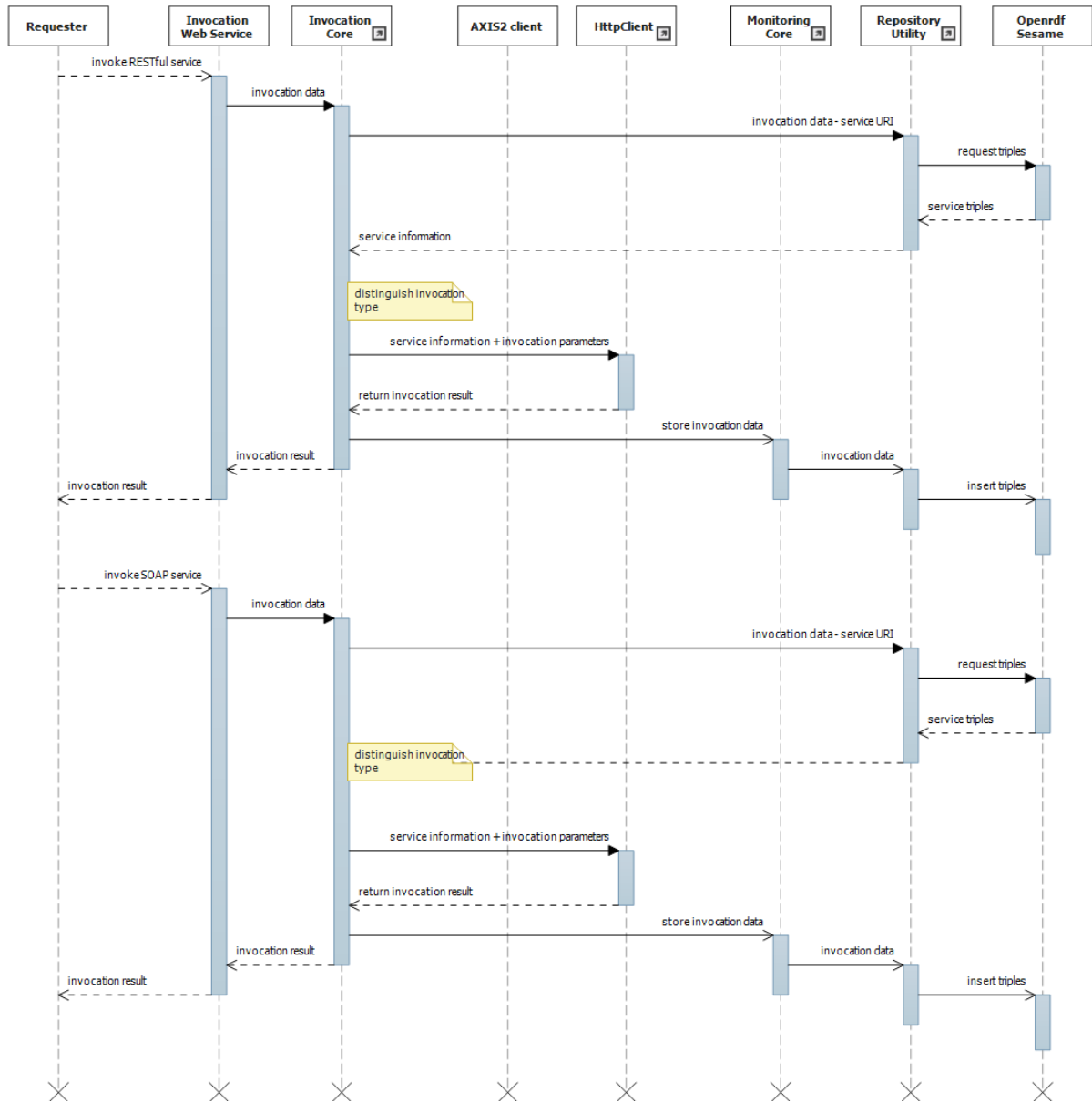


Figure 12.- Invocation sequence diagrams for WSDL and RESTful services

As we have already stated, Figure 12 presents two different sequences depending on the particular type of service that needs to be invoked.

4.2.6 Monitoring

Services registered in the SDP can be monitored in terms of QoS parameters based on the execution of the service. The Monitoring module is strongly connected to the Invocation module, because, in order to monitor the execution, the service has to be invoked using the Invocation module itself.

4.2.6.1 Component diagram

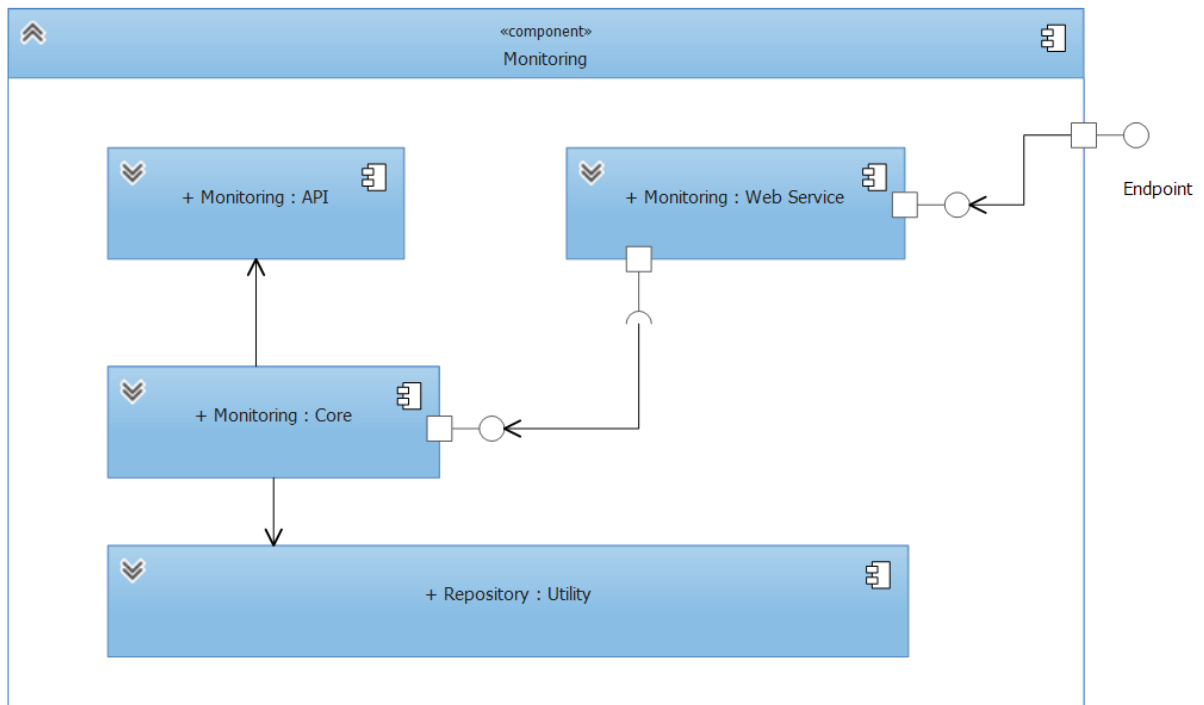


Figure 13.- Monitoring component diagram

The components involved in the monitoring module are shown in Figure 13. The core monitoring functionality accesses the repository module, where all the monitoring information is stored as shown in Figure 12.

4.2.6.2 Sequence diagram

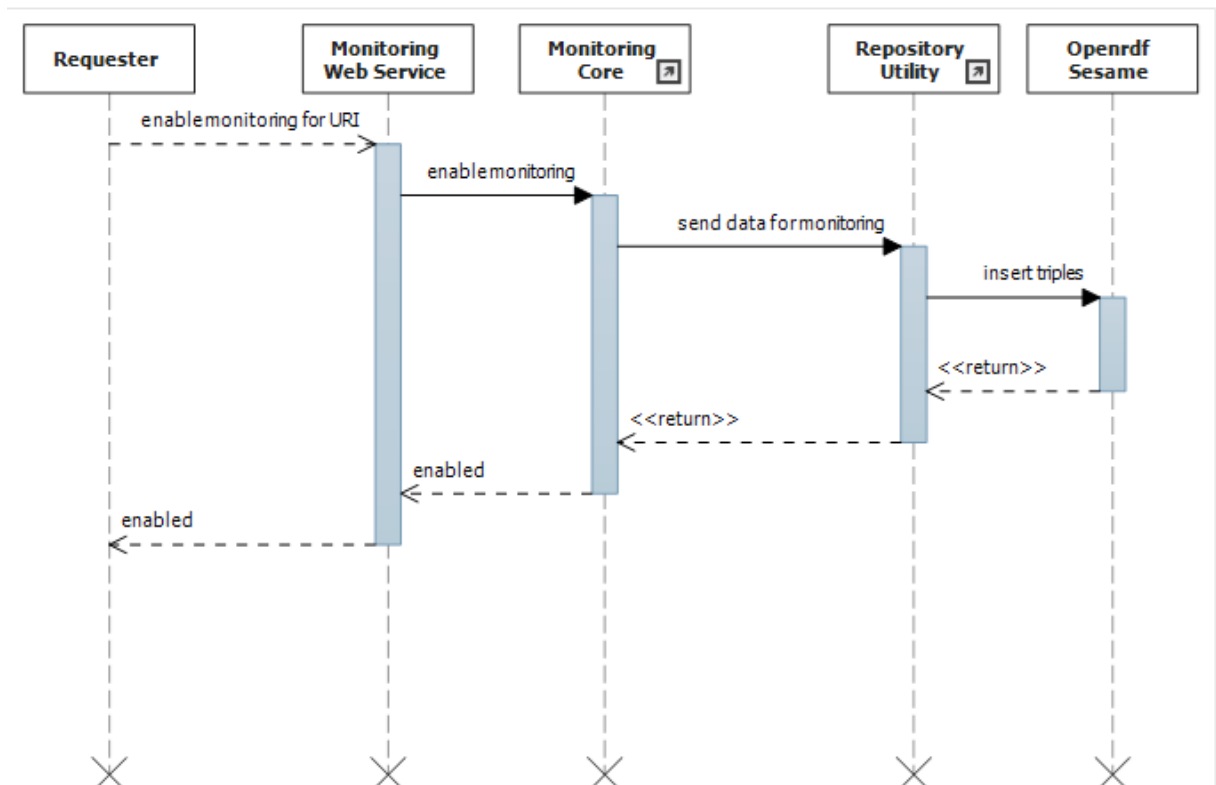


Figure 14.- Monitoring sequence diagram

The sequence diagram for the monitoring functionality presented in Figure 14 showcases how monitoring functionality is enabled for a particular service to be invoked.

4.2.7 Service Delivery Platform Dashboard

The SDP Dashboard acts as a management interface where the rest of the modules can be configured and executed. The SDP Platform encapsulates essential functionality for the SDP users and administrators.

Each module representing a functionality of the SDP (Registry, Discovery, Ranking, Invocation and Monitoring) will have a management UI in the Dashboard, where the technical aspects can be configured. They will provide also a unified interface where the functionality of the SDP can be showcased.

The SDP Dashboard can be implemented as an independent component, because it will also be built on the standard API provided by the different modules.

The SDP Dashboard will require a user profile schema for authentication and authorization which will be covered by standard mechanisms of the underlying technical platform.

4.2.7.1 Component diagram

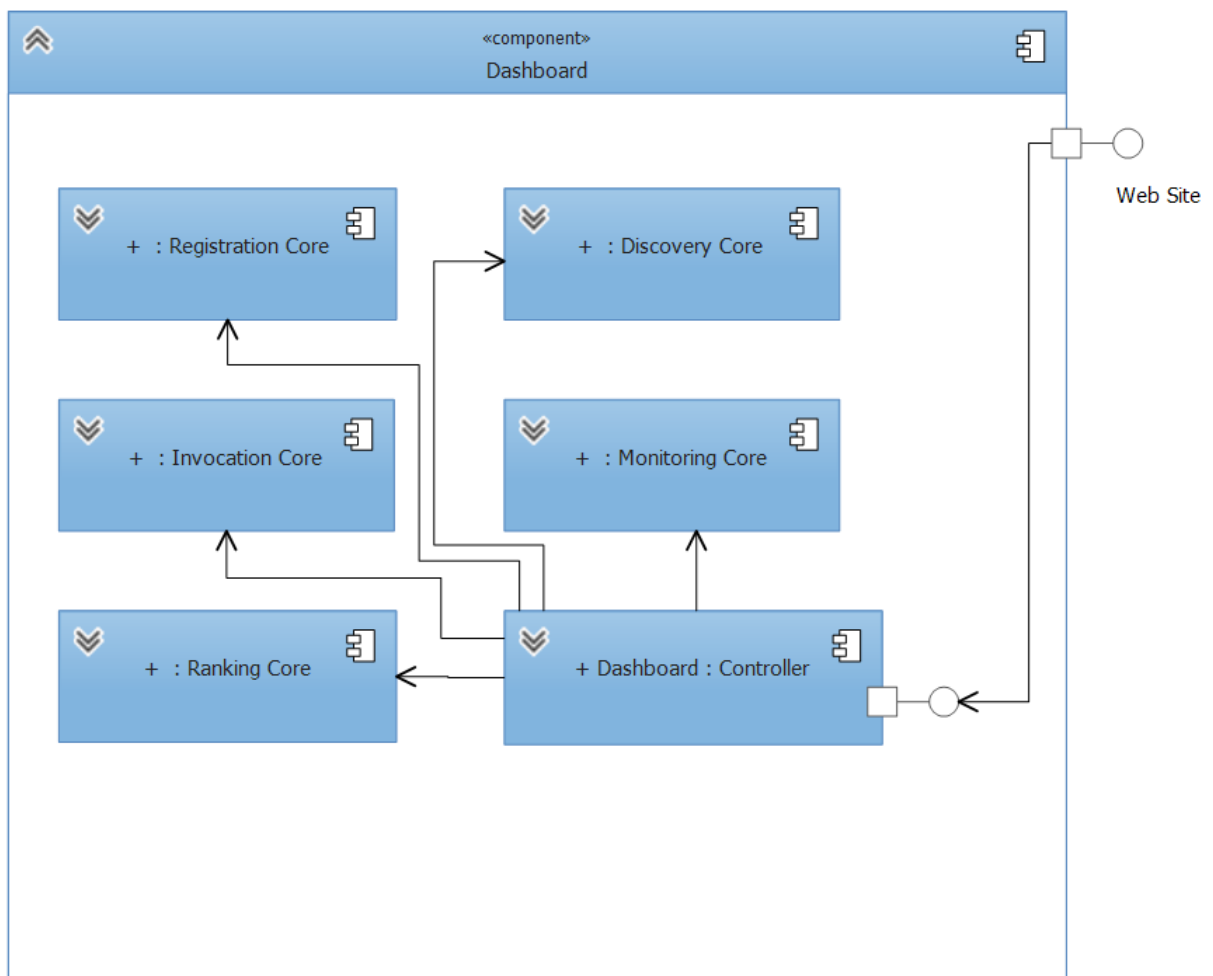


Figure 15.- Dashboard component diagram

Basically, the dashboard module, as presented in Figure 15, consists of a controller that redirects the functionality requested by the Web-based user interface to the corresponding module of the SDP.

4.2.7.2 Screenshots by functionality

The following figures shows some screenshots on the functionality currently available on the SDP dashboard (Figure 16), namely registration, discovery and invocation. Further details will be included in the description of the final prototype to be discussed in deliverable D43.4

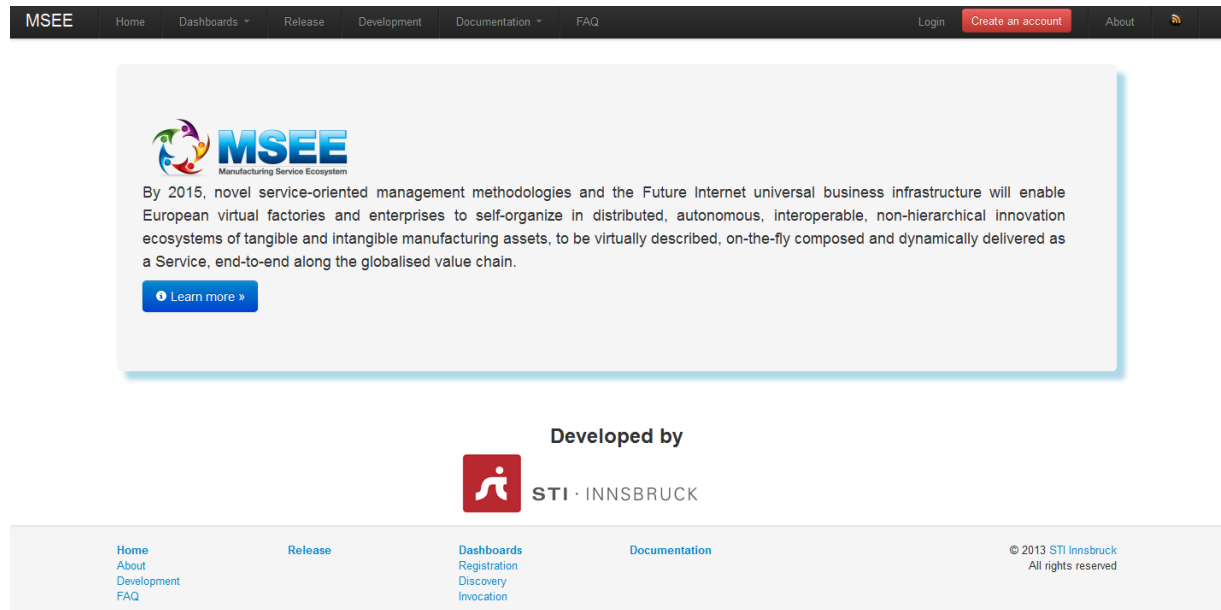


Figure 16.- SDP Dashboard

Thus, in a typical scenario, where a certain service has to be registered, as the “Carefree Washing Service” described in deliverable D61.1, the registration dashboard provides an easy to use interface to perform this task. For the registration functionality, an authentication step must be performed before allowing the user to register any service (Figure 17).

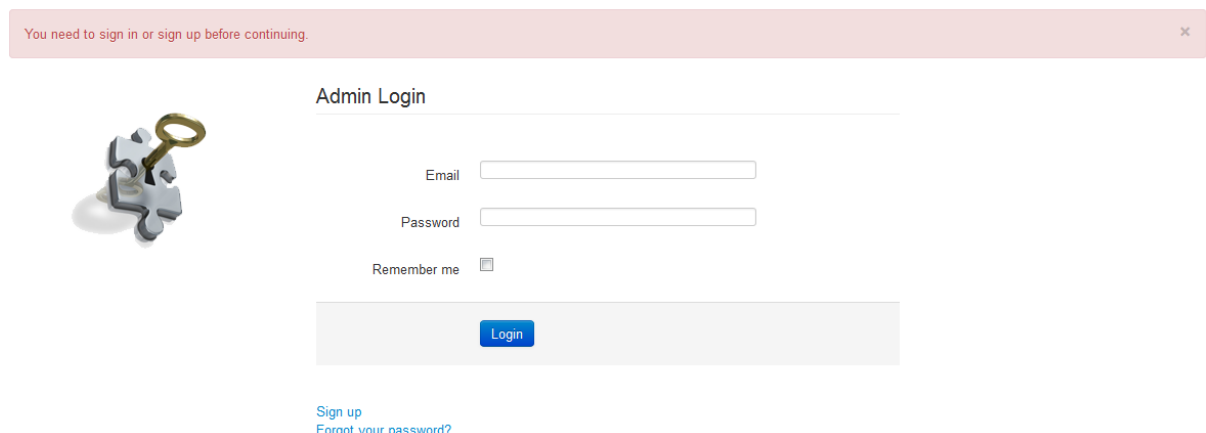


Figure 17.- Login to the dashboard

After a successful login, the user is able to register the particular service, which should be previously annotated, using the functionality shown in Figure 18, where the user provides the location of the annotated WSDL file that describes the service.

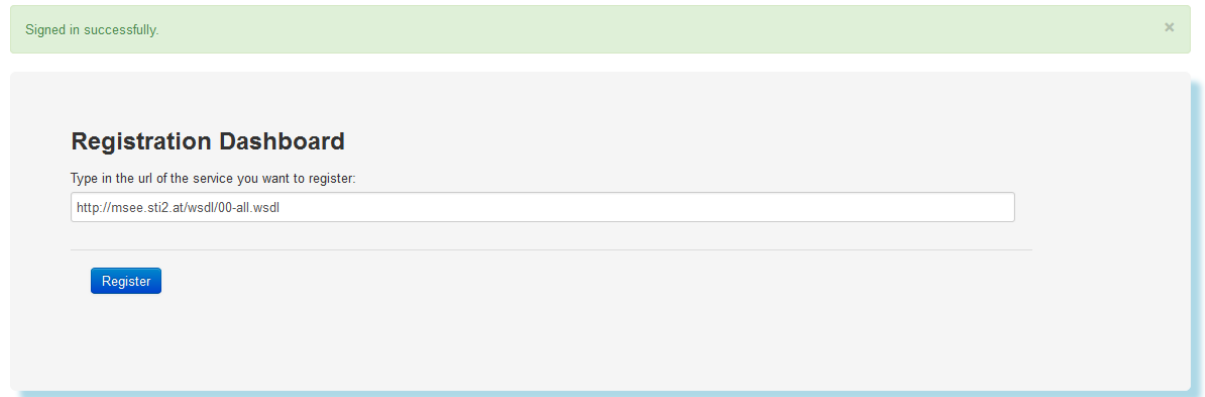


Figure 18.- Registration dashboard

For instance, in the case of the “Carefree Washing Service” mentioned before, after analysing the provided WSDL file, the registration dashboard communicates the user the successful registration of the service as in Figure 19.

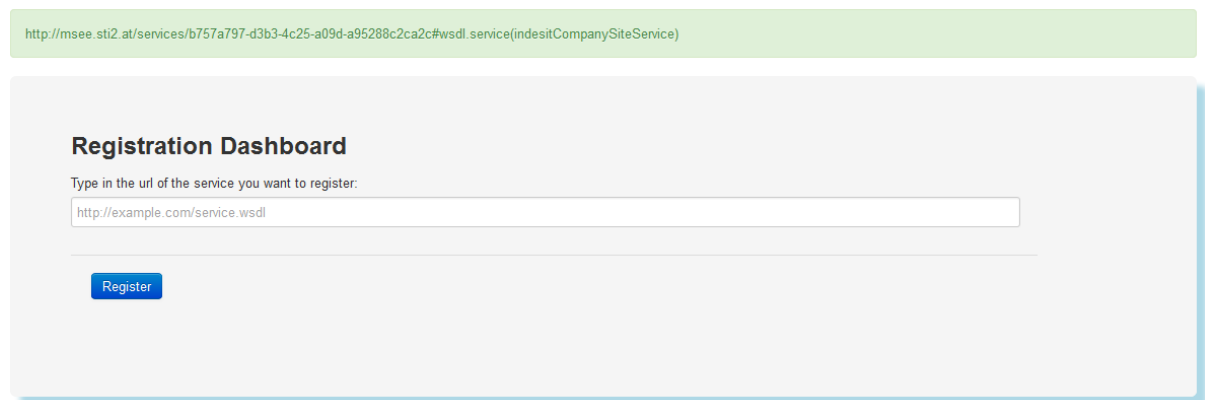


Figure 19.- Successful registration through the dashboard

After these services have been registered, one can access the discovery dashboard to look for certain services according to their categories. Figure 20 presents this part of the dashboard, where a form containing the desired category can be filled in.

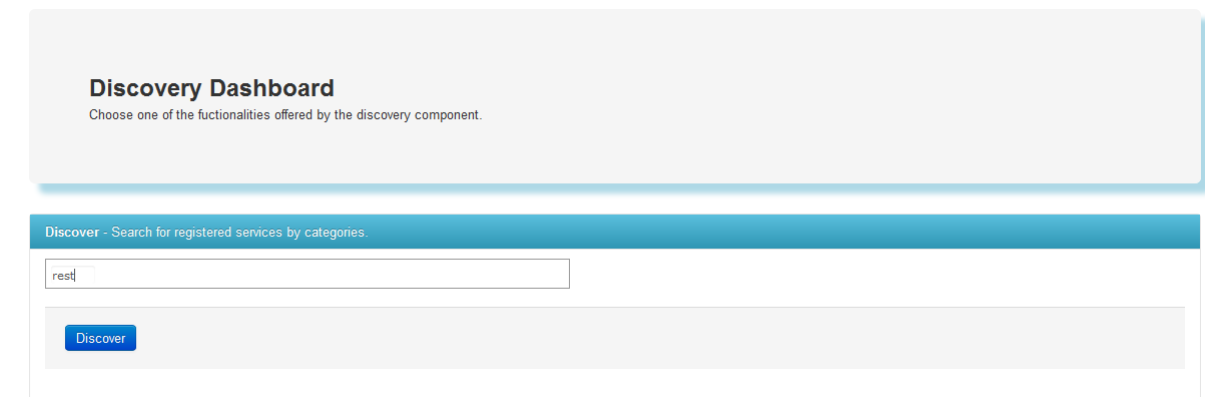


Figure 20.- Discovery dashboard

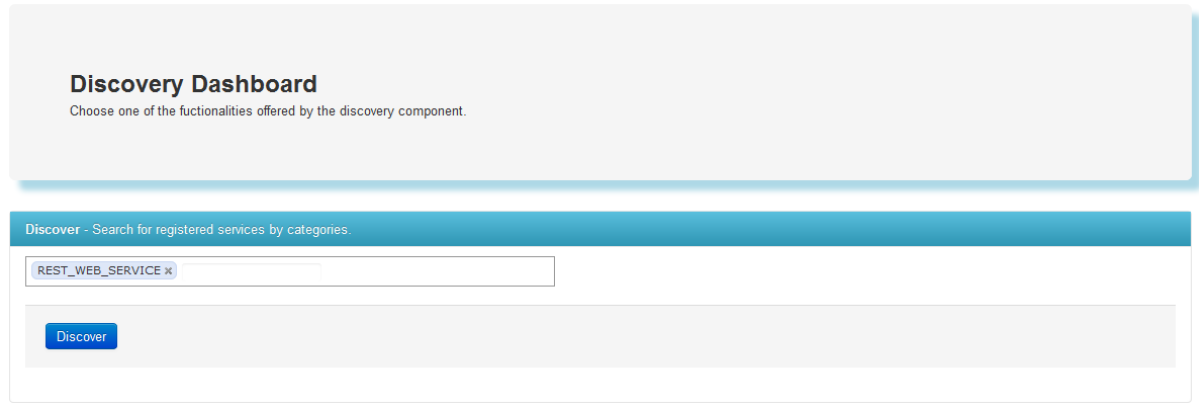


Figure 21.- Discovery category

This form supports auto-completion, so that the user only needs to type the first characters of the desired category in order to get suggestions about already registered categories, as the one shown in Figure 21. After choosing a category and clicking the button, the discovery results are shown as in Figure 22, where the RDF description stored in the repository for each result is presented.

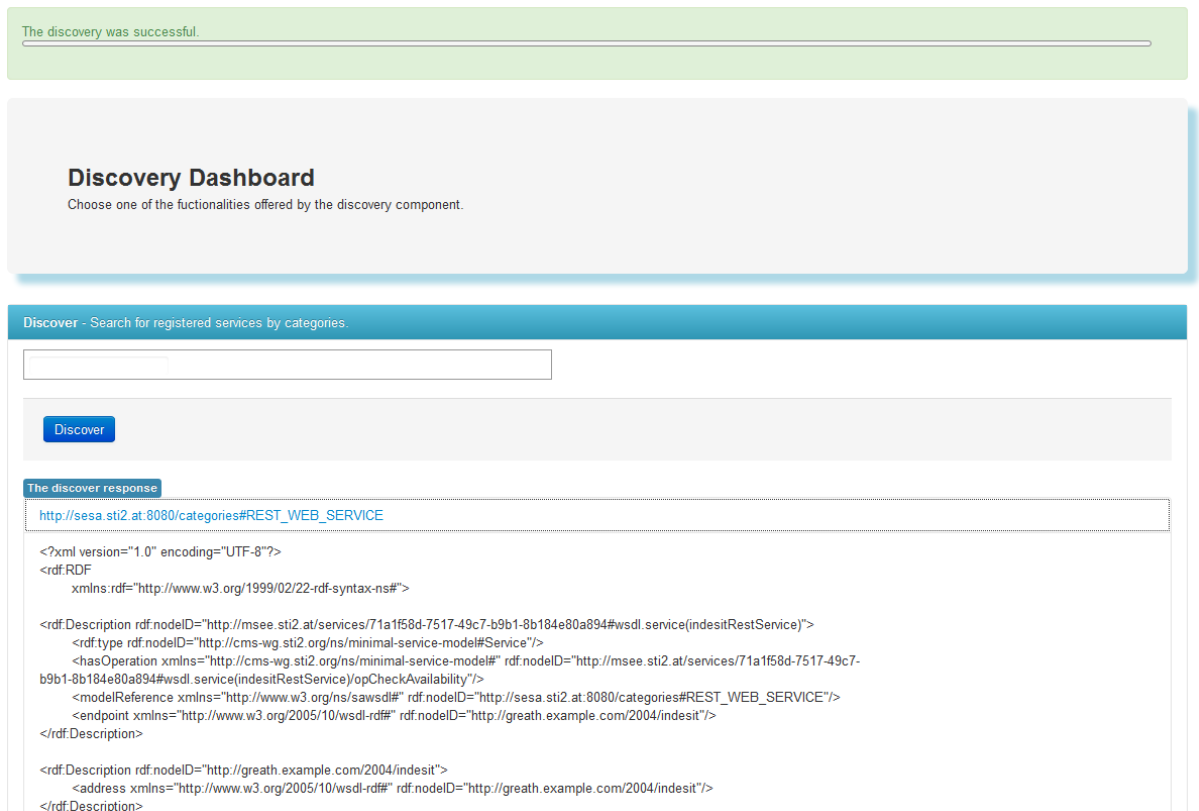
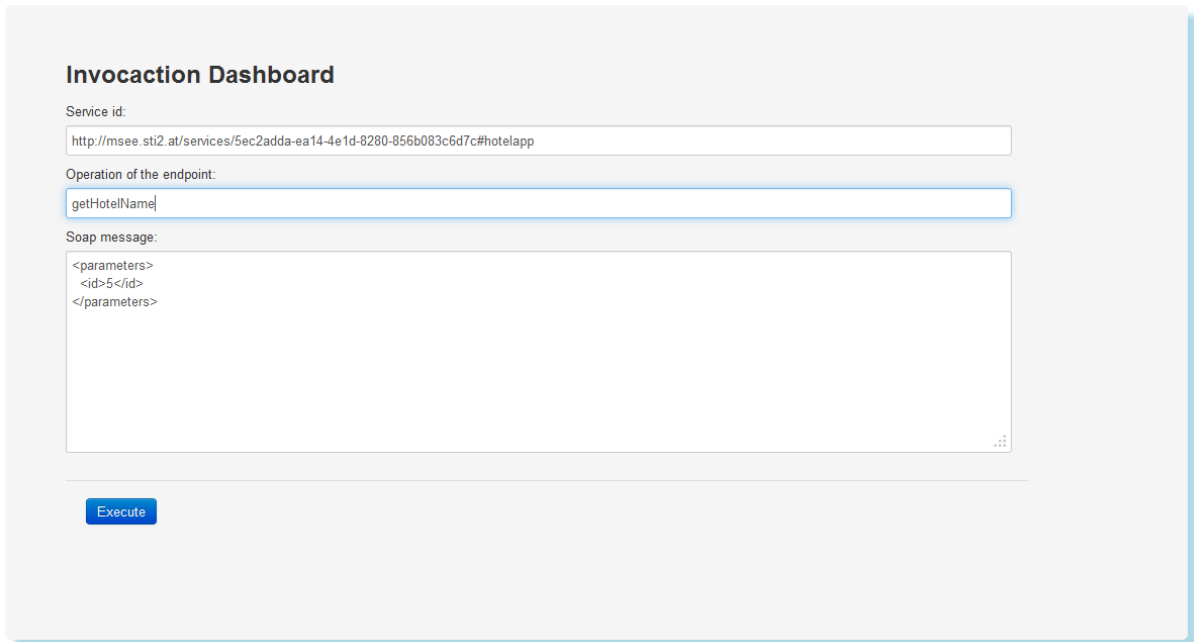


Figure 22.- Discovery category result



Invocation Dashboard

Service id:

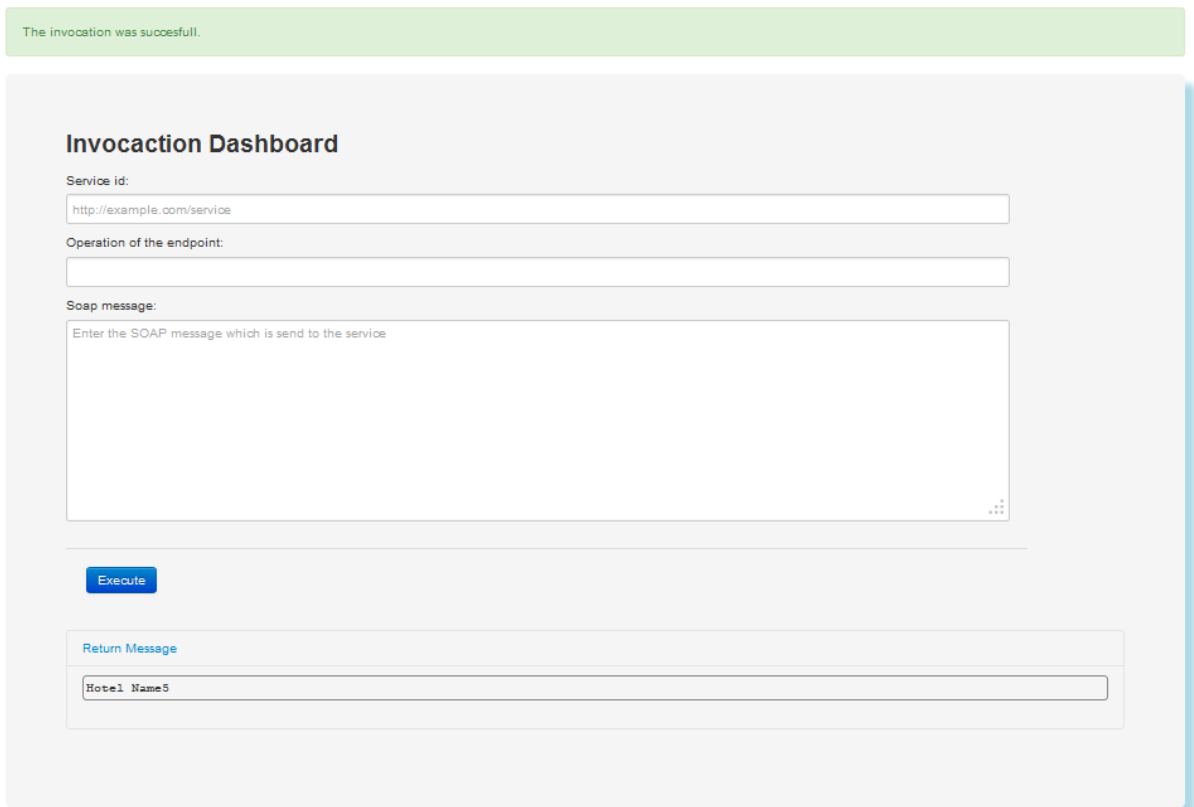
Operation of the endpoint:

Soap message:

```
<parameters>
<id>5</id>
</parameters>
```

Figure 23.- Invocation dashboard

Finally, the invocation dashboard allows to effectively invoke one of the registered services, providing the operation to be called and the needed parameters to be included in the SOAP message (Figure 23). If the invocation was successfully executed, then the dashboard shows the obtained return message as showcased in Figure 24.



The invocation was successful.

Invocation Dashboard

Service id:

Operation of the endpoint:

Soap message:
 Enter the SOAP message which is send to the service

Return Message

Figure 24.- Invocation result

5 Service Delivery Platform Integration

After the first prototype of the Service Delivery Platform released in M18, integration of the platform's first prototype has been tested in parallel to the final design to be provided in the current deliverable. In comparison with the previous design, the interfaces used by the rest of the platforms in MSEE remain the same, with minor modifications at the implementation level that will be reported in the final release of the prototype.

In regards to the Service Delivery Platform, it provides its functionality as a set of APIs, therefore the integration with the rest of the components is made through it. During the current period we have added new integration points with two of the utility services defined in MSEE, the Federated Single Sign On and the Process Indicator Registry. In the following sections we clarify the final integration of each component, as well as the new additions of the Utility Services (more information about the utility services can be found at D33.2 FI Utility Services specifications and architecture).

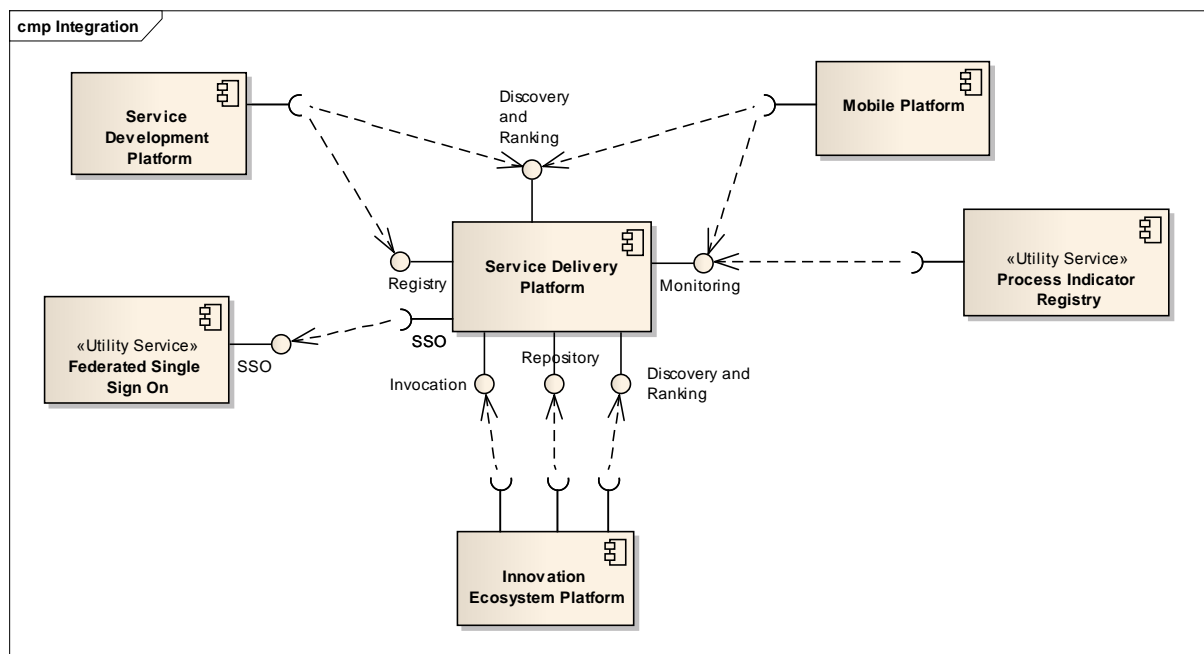


Figure 25.- Integration of the Service Delivery Platform

5.1 Integration with the Service Development Platform

During the servitization process, technical services are developed using the Service Development Platform. These services need to be registered in the Service Delivery Platform to be accessible inside the innovation ecosystems. The Service Development Platform access to the Service Delivery Platform is twofold. On the one side, it accesses the Registration API to register the technical services that have been created. On the other side, services already registered can be used as part of the creation process; therefore the Discovery API enables the search of services to be composed. This access is integrated in the Service Development Platform IDE as part of Export (registration)/Import (discovery) functionality.

5.2 Integration with the Mobile Platform

The integration with the Mobile Platform was already defined in previous deliverable and remains the same. As one of its characteristics is to provide access in the Mobile Platform to the Discovery and Monitoring functionality, it will access the correspondent APIs from the Service Delivery Platform. As part of the integration work carried out during this final revision of the design, we explored the addition of external properties to the service

descriptions, to enrich the mobile experience (see section **Error! Reference source not found.**)

5.3 Integration with the Innovation Ecosystem Platform

In the first version of the Service Delivery Platform we stated that the Innovation Ecosystem Platform accesses the services as defined in the Development Platform and registered in the Service Delivery Platform through the Discovery, Ranking and Invocation interfaces. This remains up to date, as the interfaces have not changed.

5.4 Integration with the Federated Single Sign On Utility Service

In the previous release the integration with MSEE Utility Services was not carried out. As the Service Delivery Platform provides different functionality susceptible to have different levels of access rights, for the final release we will integrate the Federated Single Sign On service to provide access control to the functionality of the Service Delivery Platform, where specially registration of services, invocation and posterior access to the monitoring information should be control by different access mechanisms. For the integration the Service Delivery Platform requires a client to the SSO protocol and connection to the MSEE provided servers. The Federated SSO Utility Service is further explained in the correspondent deliverable D33.2 FI Utility Services specifications and architecture.

5.5 Integration with the PIR - Process Indicator Registry Utility Service

Another one of MSEE Utility Services is the Process Indicator Registry (PIR), which provides the base capabilities for creating and sharing KPI models, KPI definitions and KPI historical values, in well-defined scopes like that of a single Enterprise, of a specific Virtual Enterprise (VME) or of an entire Ecosystem (MSE). In regards to the Service Delivery Platform the execution and performance of the technical services are part of the indicators that enable the measurement and analysis in this case of the technical services that are being used in the ecosystem. This is why the PIR needs to have access to the monitored services and the historical values in the different variables that are being registered. The Monitoring API provides the real time access to the data of the Service Delivery Platform that can be integrated in the PIR registry. More information about the operation of the PIR Utility Service can be found in the correspondent deliverable D33.2 FI Utility Services specifications and architecture.

6 Roadmap

For the final prototype implementation of the Service Delivery Platform, to be delivered in M30, we envision the following roadmap, based on the fulfilment status of the requirements discussed in Section 3. Figure showcases the timeline of the remaining tasks.

	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30
REST support	█	█	█	█						
Service registration	█	█	█	█						
Service discovery	█	█	█	█						
Service ranking	█	█	█	█	█	█	█			
Service invocation	█	█	█	█	█	█	█			
Service monitoring	█	█	█	█	█	█	█			
SDP dashboard improvements					█	█	█	█	█	
Testing and integration					█	█	█	█	█	█

Figure 26 Roadmap for SDP final prototype

The new requirement that was introduced concerning the support of REST-based services within the SDP modules has been implemented and will be finalized by M24. Concerning service registration and discovery, the implementation of the functionality of these modules will be finished by M24, including the fixes for the problems found during the tests performed in M21. Ranking, invocation, and monitoring modules functionality is still not fully implemented, so we plan to finish their main functionality by M24, and their complete integration with the rest of the components and the dashboard by M27.

The SDP dashboard usability improvements and integration with the rest of the components will be taken into account during the last part of the roadmap, once all the required functionalities are fully implemented. We will focus on the ease of use and management part of the dashboard, enabling a better coverage of the test cases.

Finally, we will begin the final testing and integration of the platform in M25, where all the developed modules will be thoroughly tested, especially focusing on the integration between the different components. This last task will be performed until M30, which is the expected delivery date of the final prototype of the SDP.

7 Conclusions

This deliverable covers the second and last design and specification of the MSEE Service Delivery Platform (SDP). The initial version of the design and specification was refined and updated based on the feedback received from using the platform as part of the various use cases, new user requirements and further interactions among the partners.

To represent services, we use an underlying semantic service model based on MSM. The service model has not been changed since the previous version of the SDP but it has been extended to include additional information covering aspects needed for the use cases and also for the monitoring information.

The requirements list, covering both functional and non-functional requirements, was updated and extended. For each requirement we described the importance level, as well as the degree of fulfilment in the current implementation of the SDP. While for the first version of the SDP the focus was mostly on supporting service related tasks for WSDL based services, with the second and last version of the SDP the focus has shifted to full support for both WSDL and RESTful services as well as on further development of the SDP components to enrich their functionality.

The SDP covers five different aspects of the lifecycle of service operations: registration, discovery, ranking, invocation and monitoring, which utilize a common semantic repository infrastructure. The SDP platform also includes a Dashboard where the functionalities can be tested and configured. For each of these components we provide updated component views, with respect to the first version of the specification described in deliverable D43.1, including a more detailed overview of each module of the SDP, in terms of dependencies with other modules and the behavioural views of the main functionalities.

In this deliverable we have also revisited the integration with other relevant platforms defined in MSEE, the Service Development Platform, the Mobile Platform and the Innovation Ecosystem Platform. This integration will be carried out at public API level of each module of the MSEE Service Delivery Platform, as well as defining shared resources, such as the semantic repository.

Finally, we are in the process of implementing the specification provided in this deliverable and finalizing the development of the final version of the Service Delivery Platform to give full support for the use cases of MSEE.