D4.2

Service Platform Design and Specification - Final

Instrument	Collaborative Project
Call / Topic	H2020-SEC-2016-2017/H2020-SEC-2016-2017-1
Project Title	Multi-Hazard Cooperative Management Tool for Data Exchange, Response Planning and Scenario Building
Project Number	740689
Project Acronym	HEIMDALL
Project Start Date	01/05/2017
Project Duration	42 months
Contributing WP	WP 4
Dissemination Level	PU
Contractual Delivery Date	M38
Actual Delivery Date	09/11/2020
Editor	Spyros Pantazis (SPH)
Contributors	Georgios Gardikis, Socrates Costicoglou, Nikos Papadakis, George Kolonias, George Vamvakas, Alexandros Bartzas (SPH)

Document History			
Version	Date	Modifications	Source
0.1	10/9/2020	First draft	SPH
0.2	15/9/2020	Update of requirements and system tests	SPH
0.3	20/9/2020	Update of technical specifications	SPH
0.4	9/10/2020	Technical description update	SPH
0.5	13/10/2020	API methods update, QA version	SPH
0.6	15/10/2020	QA review	CTTC
1.0.D	19/10/2020	Final version	SPH
1.0.F	09/11/2020	Approved version	DLR

Table of Contents

List of Figures

List of Tables

List of Acronyms

AB	Advisory Board
AOI	Area of Interest
API	Application Programming Interface
AVA	Avanti Communication Ltd.
C&C	Command & Control Centre
CAP	Common Alerting Protocol
CIMA	Centro Internazionale in Monitoraggio Ambientale – Fondazione CIMA
CPU	Central Processing Unit
DB	Database
DES	Decision Support Service
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
DLR-DFD	Deutsches Zentrum für Luft- und Raumfahrt e.V.; German Remote Sensing Data Center
DLR-KN	Deutsches Zentrum für Luft- und Raumfahrt e.V.; Institute of Communications and Navigation
EDXL	Emergency Data Exchange Language
EKUT	Eberhard Karls Universität Tübingen
EO	Earth Observation
EUW	End User Workshop
FBBR	Frederiksborg Brand & Redning
FCP	Forward Command Post
FFS	Forest Fire Simulator
FLI	Fireline Intensity
FR	First Responder
FRS	Fire and Rescue Service
FTP	File Transfer Protocol
GB-SAR	Ground Based Synthetic Aperture Radar
GIS	Geographic Information System
GML	Geography Markup Language
GUI	Graphical User Interface
HDD	Hard Disk Drive

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IC	Incident Commander
IG	Information Gateway
ISA	Impact Summary
ISAS	Impact Summary Service
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LU	Local Unit
MODIS	Moderate Resolution Imaging Spectroradiometer
OGC	Open Geospatial Consortium
OS	Operating System
PCF	Fundació d'Ecologia del Foc i Gestió d'Incendis Pau Costa Alcubierre
PE	Plan Execution
PF	Plan Formation
RAM	Random Access Memory
REST	Representational State Transfer
ROS	Rate of Spread
RVA	Risk and Vulnerability Assessment
SA	Situation Assessment
SITREP	Situation Reporting Service
SM	Scenario Management
SMAC	Scenario Matching Service
SMES	Scenario Management Service
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
SP	Service Platform
SPH	SPACE Hellas S.A.
TOC	Table of Contents
UeRM	Users and Roles Management Module
UI	User Interface
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
VPN	Virtual Private Network
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WP	Work Package

Intentionally blank

Executive Summary

This document presents the final version of the technical requirements, architecture and functionality of the HEIMDALL Service Platform (SP) elaborated in close collaboration with the technical partners in the HEIMDALL project. The main objective of this document is to provide a technical specification enabling technical contributors and partners to understand how to communicate and share information with the SP. Therefore, topics include the external and internal architecture design, interfaces, formats, functionality, methods, configuration and software issues.

The main task contributing to this deliverable is T4.1 – Service Platform and Interfaces. However, contributions regarding the interfaces were made by the other technical tasks of WP4, WP5 and WP6 where the other technical components of HEIMDALL have been developed. Furthermore, T2.4 – Service Concept Specification and System Architecture defined the scope of the SP in the overall HEIMDALL system.

All the identified submodules of the SP were developed, integrated and tested. The SP was deployed as Virtual Machines (VM) and Docker images, with adequate resources, within a host server dedicated to HEIMDALL within the private data centre of SPACE Hellas (SPH). A test campaign, focused on the features needed, was executed with specifically defined test cases. The defined test cases were successfully executed and all SP functionalities were validated, including access control, monitoring, information routing, data handling, incident management and proper interfacing and information exchange to other subsystems.

The HEIMDALL SP has been successfully developed, integrated into the system and tested, thus bringing the platform at a stable status, ready to support the final demo.

1 Introduction

The discussions among technical partners within the context of WP2, as well as the other technical WPs led to the design of the HEIMDALL architecture and the placement of the SP as the component that will facilitate the communication among its different modules. This document describes WP4/T4.1 activities of the HEIMDALL project in finding and designing technical solutions facilitating the creation of a distributed planning and emergency response platform. The document focuses on the different requirements and functionalities that the SP satisfies and provides.

This document is an updated version of Deliverable D4.1[2]. It presents the final design and specifications of the SP and its interfaces, as well as release the software prototype.

In particular, this document is organised as follows:

- Section 2 specifies the technical requirements for the SP modules.
- Section 3 describes the SP in the context of the overall HEIMDALL system, inputs and outputs and interfaces with other HEIMDALL components.
- Section 4 focusses on the SP building blocks and their functionalities.
- Section 5 presents the technical specifications.
- Section 5 presents the internal technical testing scenarios and their results.
- Finally, Section 7 summarizes the work carried out so far.

2 Technical Requirements

This section includes the list of technical requirements for the module being addressed.

2.1 Interface Requirements

The SP interfaces to all services of the HEIMDALL system:

- User Role Management (more information is available in Deliverable D4.5 [3]).
- User interface (more information can be found in Deliverables D4.7 [4] and D4.9 [5]).
- Communication and information sharing services (Catalogue and interface to other instances).
    - Information gateway (more information can be found in Deliverable D4.14 [6]).
    - Satellite communications (more information is available in D4.17 [7]).
- HEIMDALL Data Sources
    - Earth Observation (more information can be found in Deliverable D5.2 [8]).
    - In-situ Sensors (aerial- and ground-based) (more information can be found in Deliverable D5.5 [9]).
- Mobile application for the first responders (more information will be available in Deliverables D5.7 [10] and D5.8 [11]).
- External data sources and services (e.g., Copernicus services, resource management, weather forecasting, etc.) (more information is available in Deliverable D5.10 [12]).
- Simulators (more information is available in D5.13 [13]).
- Risk assessment (more information can be found in Deliverable D6.3 [14]).
- Impact summary generation (more information can be found in Deliverable D6.5 [14] and D6.8 [16]).
- Decision support (more information can be found in Deliverable D6.11 [17]).
- Scenario management (more information is available in Deliverable D6.15 [18]).

2.1.1 **Hardware Interfaces**

The HEIMDALL Service Platform is deployed within the secure private data centre of SPH, which is certified as per ISO 27001:2013 with regard to information security. It connects to the internet via redundant leased lines. The physical server that hosts the SP software is a Dell PowerEdge R630 model (Figure 2-1) with the following characteristics:

- CPU: Intel Xeon E5-2620 16 Core@2.10 GHz
- Memory: 128 GB
- Storage: 3TB

Figure 2-1: Dell PowerEdge R630 server.

2.1.2 **Software Interfaces**

The HEIDMALL services are deployed as containers and/or virtual machines (VMs). In detail:

- SP and its components, namely the Geoserver the relational/GIS database, the FTP service and the main SP application are deployed in a VM with 4 Cores, 8 GB RAM and 256 GB HDD. OS is Windows 2012 Server. In addition, the SP includes bespoke

components in order to provide services described in Section 5 (like authentication, OGC services, integration of external services). It also utilises a WebSocket server to push notifications to the Graphical User Interface (GUI).
- GUI and WebSocket servers are deployed as containers in a Rancher farm [24] with four hosts, each one of them with 4 cores, 8 GB RAM and 500 GB shared storage. Hosts OS is Ubuntu 16.04 LTS. GUI depends on the SP to fetch data, store content and/or send commands to back-end systems like Information Gateway (IG) or Forest Fire Simulator (FFS), and depends on the WebSocket server to fetch asynchronous notifications.
- Load balancing software is also a container deployed at the same farm. Load balancing sits on top of the various SP services and makes them available to all VPN endpoints.
- IG is deployed as a VM with 2 Core, 4 GB RAM and 16GB HDD. OS is Ubuntu 18.04. IG is the Information Gateway of HEIMDALL.
- VPN Access Gateway is also a VM with 2 Cores, 512 MB RAM and 10 GB HDD. OS is FreeBSD.

The list above contains two external modules to the SP, namely the GUI and the IG. To speed up the deployment of the HEIMDALL system and its demonstration, in collaboration with AVA and DLR-KN, SPH decided to host the two modules in its own infrastructure dedicated to HEIMDALL.

The above mentioned configuration is related to system requirements Sys_Gen_2, Sys_Gen_4, Sys_Gen_7, Sys_Gen_8, Sys_Gen_9, Sys_Gen_17, Sys_Gen_18, Sys_Int_1, Sys_Int_2, Sys_Int_3, Sys_Int_4, Sys_Int_5, Sys_IntData_1, Sys_IntData_2, Sys_IntData_3, Sys_IntData_4, Sys_IntUeMan_1, Sys_IntUeMan_5, Sys_IntUeMan_6 and Sys_IntUeMan_7.

2.1.3 Communication Interfaces

The SP uses either HTTP or for secured connection HTTPS to connect to the HEIMDALL network and the internet.

In order to enable the HEIMDALL distributed architecture and ensure secure connectivity among the remote subsystems, a VPN, managed by SPH, has been established over the public network (Internet). Traffic among VPN peers is transported via “tunnels”, so that two remote hosts, even if they belong to different physical networks, can communicate as if they were co-located in the same subnet.

Traffic in the HEIMDALL VPN is encrypted on end-to-end basis using state-of-the-art encryption techniques, so that potential eavesdroppers cannot intercept and decode the exchanged information. Apart from encryption, VPN security mechanisms assure authentication of both peers (to prevent man-in-the-middle attacks) as well as data integrity (to prevent alteration of exchanged data by a malicious entity). In HEIMDALL, we are using the most popular security standards for VPN: SSL (Secure Socket Layer) and IPSec (IP Security).

This configuration is related to system requirements Sys_Int_3, Sys_Int_4

2.2 Functional Technical Requirements

The listed requirements have also been included in D4.1[1]. The new requirements defined in this document are TR_SP_16 and TR_SP_17.

The categorisation of the requirements as short-, med- or long-term follows the labelling of the respective system requirements from which they were inherited.

2.2.1 Short Term Requirements

Table 2-1: Technical Requirement TR_SP_01

Requirement ID:	TR_SP_01
Related SR(s):	• Sys_IntData_1 • Sys_IntData_3
Description: The SP shall provide a database to store GIS data.	
Rational: Storage and retrieval of georeferenced information is an integral part of the activities the users will request from the SP to perform during operations (either planning or response).	
Stimulus: Request for GIS data to be written or read.	
Response: The SP takes the GIS data and stores them in its database. Upon read request retrieves the data and serves them to the requesting module.	
Verification Criterion: Perform multiple read and write operations in the GIS database, and check that the data is correctly read/written.	
Notes: none	

Table 2-2: Technical Requirement TR_SP_02

Requirement ID:	TR_SP_02
Related SR(s):	• Sys_IntData_1 • Sys_IntData_2
Description: The SP shall provide a database to store EO data.	
Rational: The usage of EO data form the phases of planning and response to hazards are essential for successful operations.	
Stimulus: EO data in processed form (i.e., including metadata) is provided to the SP.	
Response: The SP takes this data and stores it in the corresponding database.	
Verification Criterion: The user is able to read EO data and metadata through the corresponding user applications.	
Notes: none	

Table 2-3: Technical Requirement TR_SP_03

Requirement ID:	TR_SP_03
Related SR(s):	• Sys_IntData_1 • Sys_IntData_2
Description: The SP shall provide a database to store other sensor data. More specifically, the SP shall store data coming from the following HEIMDALL sensors:	

• UAVs • GB-SAR • In-situ sensors
Rational: Within HEIMDALL UAV swarms and GB-SAR are going to be used to provide additional sources of information. The gathered sensor data, either in raw or in processed form, shall be stored in a database and made available for usage from the HEIMDALL services.
Stimulus: UAV data (e.g., hotspot detection) and GB-SAR data is fed to the SP.
Response: The SP receives the data and stores it in an appropriate database.
Verification Criterion: The user is able to retrieve the data from the corresponding HEIMDALL sensors.
Notes: none

Table 2-4: Technical Requirement TR_SP_04

Requirement ID:	TR_SP_04
Related SR(s):	• Sys_IntData_3
Description: The SP shall receive and store georeferenced information from first responders through the mobile applications.	
Rational: Storage and retrieval of georeferenced information received from operators in the field is an integral part of the activities the users will request from the SP to perform during operations (either planning or response).	
Stimulus: The corresponding app sends such information to the SP.	
Response: The SP stores the received information to the corresponding databases and notifies the affected services.	
Verification Criterion: The user is able to retrieve the data send from the users deployed in the field.	
Notes: none	

2.2.2 **Mid-Term Requirements**

Table 2-5: Technical Requirement TR_SP_05

Requirement ID:	TR_SP_05
Related SR(s):	• Sys_Int_1 • Sys_Gen_2
Description: The SP shall provide means to configure its operational parameters based on the disaster management and/or decision-making phases selected (activated). The operational parameters offered for reconfiguration are: • Thresholds and any parameters affecting data visualisation • Service orchestration parameters	

Rational: The SP is going to be used in all phases of disaster management; hence, it should be able to change its configuration to fit the needs.
Stimulus: Update of the SP configuration and its activation by the user.
Response: The SP passes the updated configuration parameters to the affected services.
Verification Criterion: The operation of the SP is adapted to match the configuration activated.
Notes: none

Table 2-6: Technical Requirement TR_SP_06

Requirement ID:	TR_SP_06
Related SR(s):	• Sys_Int_4
Description: The SP shall integrate multiple heterogeneous data sources using standard interfaces. More specifically, the SP shall connect to the following sensors and sources: 1 UAVs 2 GB-SAR 3 Geotechnical and hydrological landslide sensors Using the following standards: • REST APIs	
Rational: The reception of timely sensor information is essential for the planning and response phases.	
Stimulus: A sensor sends its data to the SP through standardised interfaces.	
Response: N/A	
Verification Criterion: Sensor data is entered in the platform and stored in the appropriate databases.	
Notes: none	

Table 2-7: Technical Requirement TR_SP_07

Requirement ID:	TR_SP_07
Related SR(s):	• Sys_Int_5
Description: The SP shall integrate multiple heterogeneous data sources using proprietary interfaces. More specifically, the SP shall be able to connect to sensors that do not provide a standardised interface. Note: Until M18, the need for interfacing with sensors providing proprietary interfaces has not been materialized.	

Rational: The reception of timely sensor information is essential for the planning and response phases.
Stimulus: A sensor sends its data to the SP through custom/proprietary interfaces.
Response: N/A
Verification Criterion: Sensor data is entered in the platform and stored in the appropriate databases.
Notes: none

2.2.3 **Long-Term Requirements**

Table 2-8: Technical Requirement TR_SP_08

Requirement ID:	TR_SP_08
Related SR(s):	• Sys_Int_2
Description: The SP shall offer a service orchestrator able to start services remotely, upon demand from authorised users belonging to other authorities.	
Rational: End users that belong to different authorities shall be able to invoke HEIMDALL services remotely through the SP service orchestrator.	
Stimulus: A service request received by an authorised user.	
Response: Upon completion of the service, the generated product is available to the user and/or other services (in the workflow).	
Verification Criterion: The SP administrator is able to see the service starting (e.g., through log inspection).	
Notes: none	

Table 2-9: Technical Requirement TR_SP_16

Requirement ID:	TR_SP_16
Related SR(s):	• Sys_Gen_19
Description: The SP shall offer the capability to create, store and share actions for teams.	
Rational: User requirement related to extending the C2 capabilities of HEIMDALL	
Stimulus: A request to create an action for teams and then to retrieve it.	
Response: Each team member should receive the action related to them.	
Verification Criterion: The created action is visible in the team member’s terminal	
Notes: none	

Table 2-10: Technical Requirement TR_SP_17

Requirement ID:	TR_SP_17
Related SR(s):	• Sys_Gen_20
Description: The SP shall offer the capability to create, store and manage country-specific information about legal/regulatory or organisational frameworks.	
Rational: Support to the user by providing the legal/regulatory context.	
Stimulus: Legal/regulatory information (free text) is stored and associated with a specific country. The SP is queried for country-specific information.	
Response: The SP should respond with the information stored.	
Verification Criterion: The SP should respond with the information stored.	
Notes: none	

2.3 Non-Functional Requirements

2.3.1 Short Term Requirements

Table 2-11: Non-Functional Technical Requirement TR_SP_09

Requirement ID:	TR_SP_09
Related SR(s):	• Sys_Gen_17
Description: The SP shall run on virtualised IT infrastructures.	
Rational: Large-scale SP deployments involving considerable resources and/or with high availability constraints will require deployment on data centre and/or cloud infrastructures.	
Verification Criterion: The SP operates as expected, with all core functionalities available, in a virtualised infrastructure.	
Notes: none	

2.3.2 Mid-Term Requirements

Table 2-12: Non-Functional Technical Requirement TR_SP_10

Requirement ID:	TR_SP_10
Related SR(s):	• Sys_Gen_4 • Sys_Int_2 • Sys_Int_4
Description: The SP shall be easily extended with new sensors, modules, etc.	
Rational: The addition of a new sensor platform or a new processing module can be achieved without modifications in the SP code.	
Verification Criterion: The sensors/modules to be added need to expose interfaces conforming to a pre-defined set of protocols.	

Notes: Utilisation of open standards heavily contributes towards this goal.

Table 2-13: Non-Functional Technical Requirement TR_SP_11

Requirement ID:	TR_SP_11
Related SR(s):	• Sys_Gen_8 • Sys_Gen_9 • Sys_Gen_17 • Sys_Gen_18
Description: The SP shall be stable and resilient to faults (either software or wrong usage from the users).	
Rational: Since the SP is handling critical information, the system should illustrate high availability under such circumstances.	
Verification Criterion: Verification that the SP achieves high availability for a period of one month. Verification that the operation of the SP is fully restored and critical data is maintained after a service restart.	
Notes: none	

Table 2-14: Non-Functional Technical Requirement TR_SP_12

Requirement ID:	TR_SP_12
Related SR(s):	• Sys_IntData_4 • Sys_IntUeMan_1 • Sys_IntUeMan_5 • Sys_IntUeMan_6 • Sys_IntUeMan_7
Description: Support multiple users operating simultaneously.	
Rational: HEIMDALL is supposed to be accessed by several users, including actors on the field.	
Verification Criterion: Simultaneous access of 20 users	
Notes: The target number of simultaneous users will be revised during the project execution. The final target will be presented in D4.2, delivered in M38.	

Table 2-15: Non-Functional Technical Requirement TR_SP_13

Requirement ID:	TR_SP_13
Related SR(s):	• Sys_Int_3
Description: The SP shall facilitate the exchange of information with existing operational tools. More specifically these tools shall be: • The location of units in the field (part of the dispatcher)	

Rational: The user wants to use HEIMDALL together with their existing tools, hence data exchange between them would facilitate the cooperation.
Verification Criterion: Data from existing tools can be read by the SP and be used by the rest of the HEIMDALL services.
Notes: none

Table 2-16: Non-Functional Technical Requirement TR_SP_14

Requirement ID:	TR_SP_14
Related SR(s):	• Sys_Int_3
Description: The SP shall facilitate the reception of data from external sources and other systems/services. More specifically, these external sources shall be: • Weather and hydrological services • Copernicus services	
Rational: HEIMDALL shall be able to handle input coming from external services, regarding weather and hydrological information as well as images and products from satellite services.	
Verification Criterion: The SP is able to interface with such external services in order to get information, as data and map layers, in order to make them available to the other modules of the HEIMDALL platform.	
Notes: none	

2.3.3 **Long-Term Requirements**

Table 2-17: Non-Functional Technical Requirement TR_SP_15

Requirement ID:	TR_SP_15
Related SR(s):	• Sys_Gen_7 • Sys_IntData_4 • Sys_IntUeMan_1 • Sys_IntUeMan_5 • Sys_IntUeMan_6 • Sys_IntUeMan_7
Description: The SP shall accommodate multiple tenants.	
Rational: During operations, either planning or response, the SP will be used by multiple tenants at the same time. A tenant is a group of users who share a common access with specific privileges to the software instance of the SP.	
Verification Criterion: The SP services are provided to the practitioners through a Software-as-a-Service scheme.	
Notes: none	

3 Reference Architecture

3.1 HEIMDALL overall architecture

The architecture of HEIMDALL's local unit is shown in Figure 3-1, whereas details about it are provided in deliverable report D2.12.

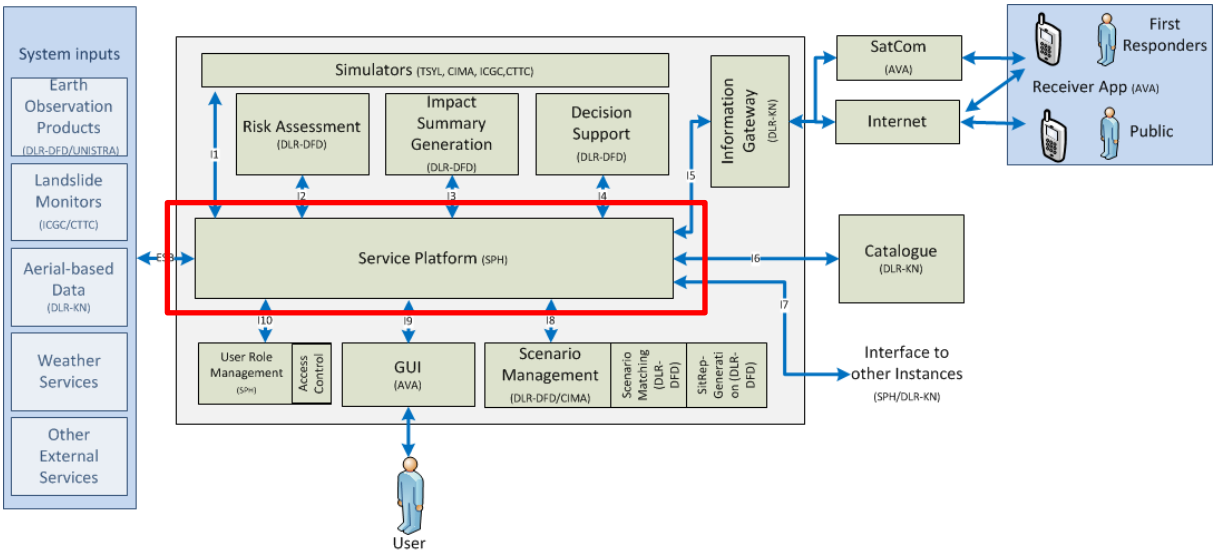

Figure 3-1: Local unit architecture.

The core element of the HEIMDALL architecture is the Service Platform (SP) offered to each individual authority for response planning and scenario building. The SP architecture is shown in Figure 3-2. Multiple instances of the platform can be interconnected in a federated scenario to facilitate cooperation. The SP offers services to be used either in a) the preparedness and mitigation phases; or in b) the response and recovery phases. The SP accommodates and interconnects various internal services and components to facilitate multi-hazard management realised by HEIMDALL. The SP also implements a repository for geospatial and plain data as well as a GIS engine for data representation and transformation. The SP offers interfaces for internal and external data sources as well as interfaces for the horizontal, peer-to-peer communication with other Local Units (see Section 3.2.7). A graphical user interface (see deliverable report D4.7 [4]) facilitates interaction with the end users in an intuitive and user-friendly manner. Overall, the SP and its user interface offer to end users a complete integrated environment for response planning and scenario building, also facilitating the exchange of data with other authorities.

As shown in Figure 3-2, the SP mainly consists of the following subsystems:

- The Data Repository, for storing all data (geo-referenced and generic data)
- The Enterprise Service Bus, for mediating messages and data between HEIMDALL services

Section 4 describes the technologies on which these components were developed.

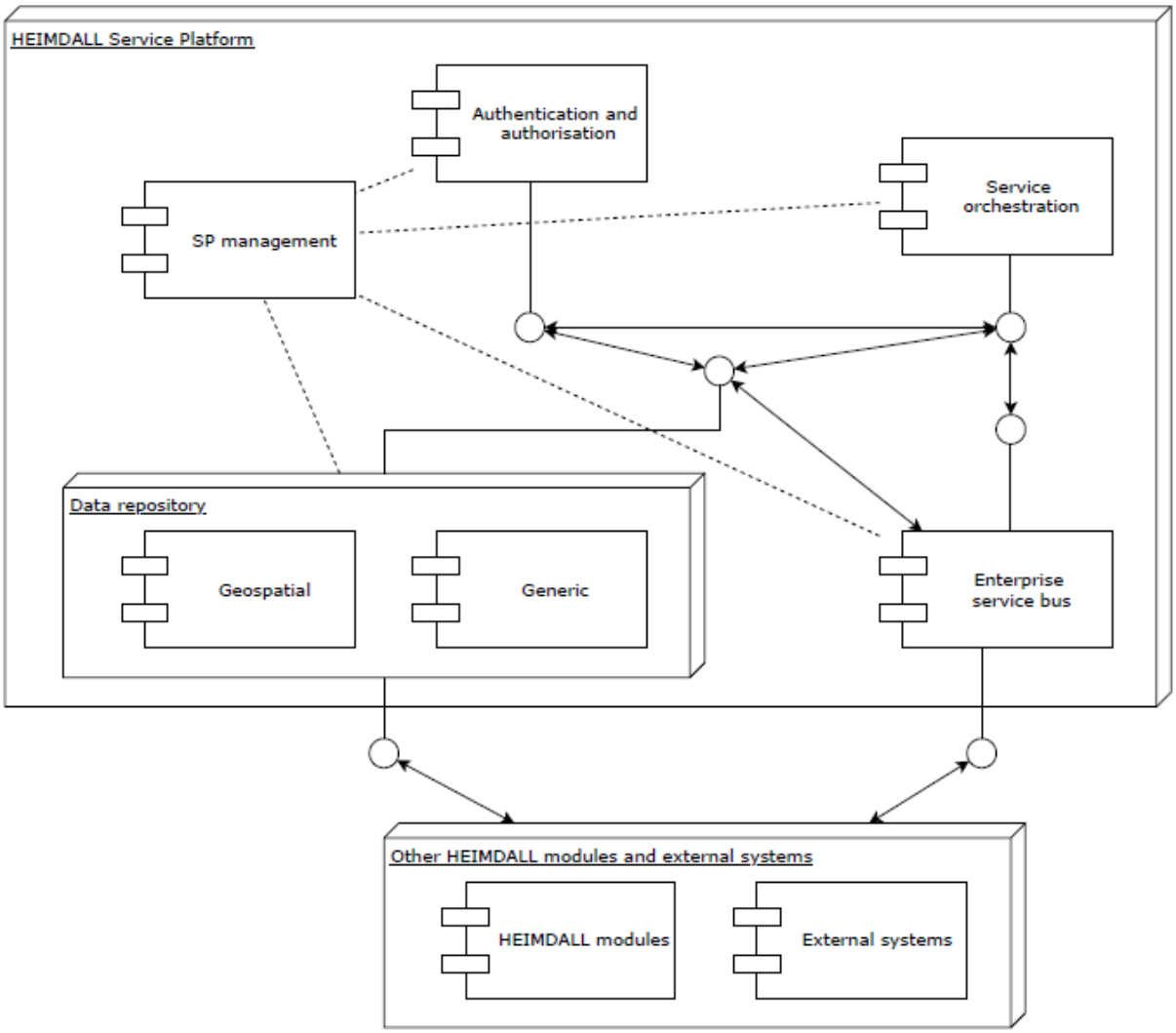

Figure 3-2: Service platform architecture.

The SP provides services to the users in order to:

1) Acquire data (either raw or processed) not provided by the currently installed data sources or tools.
2) Communicate sensor data, events and simulation results and retrieve decisions/ recommendations.

Table 3-1: SP products and services.

Products and/or Services	**Inputs needed** *inputs to generate each output*	**Provided by** *module or external system providing the input*	**Used by** *module consuming the product/service*
Workflow invocation service	Workflow description (sequence of services invoked and products manipulated)	GUI (such action is triggered by the GUI)	All other HEIMDALL services
SP monitoring/management	Basic monitoring metrics regarding	SP	• SP • UI

	SP operation, through log files, etc.		
Interfaces with various services	• Valid user credentials • Access rights	• UI • UeRM	All other HEIMDALL services

In Section 5, the status of the API providing these services is presented, accompanied with examples.

3.2 Interfaces with other HEIMDALL components

The HEIMDALL components interact with the Service Platform for two purposes – a) for sending and retrieving data and b) for triggering workflows.

Data exchange is mainly performed over HTTP directly to the SP data repository. Geospatial data can be published/retrieved via the OGC-compliant services (WFS, WCS) and can be retrieved fully rasterised via the WMS service. In addition, a REST-based interface is available. Sensor data is exchanged via the OGC SOS service, while generic data can be published and retrieved via a proprietary HTTP REST interface, whose API will be defined within the HEIMDALL project.

Workflow triggering can be conducted via the interfaces exposed by the workflow engine, commonly HTTP REST and SOAP, depending on the binding chosen.

The SP provides access to data resources and functionality by use of different RESTful web services. Table 3-2 shows Enterprise Service Bus (ESB) and I1 – I10 as the interfaces connecting the SP with the rest of the HEIMDALL modules.

Table 3-2: Interfaces with other components.

Interface	Short description	Methods	Protocol
I1- I10	RESTful web service interface	GET, POST, PUT, DELETE	HTTP(S)
ESB	Message brokering service	Multi-instance communication REST, SOAP, Sockets, FTP, e-mail, etc.	HTTP(S), FTP(S)

The SP provides a REST API to the HEIMDALL modules for accessing, creating, updating and deleting of their data resources. The client requesting must attach any input needed by the HEIMDALL modules as a data resource.

3.2.1 Interface with the scenario management module

The interface with scenario management is the I8 as indicated in Table 3-3. More details about this interface can be found in D6.15 [18].

Table 3-3: Interface with scenario management module.

Interface	Short description	Methods	Protocol
I8	RESTful web service interface	GET, POST, PUT, DELETE	HTTP(S)

From within the HEIMDALL VPN, the technical documentation of the API is available in http://10.0.8.123/heimdall/heimdall_sm_api/doc/, whereas the GET/PUT/DELETE endpoints

can be found in http://esb.heimdall.sp/services/sm/[scenario|risk|measure....]/{id}. The documentation as well as the endpoints are accessible only within the HEIMDALL VPN.

3.2.2 Interface with the modelling and simulation module

The interface with the modelling and simulation module is the I1 indicated in Table 3-4. More details about this interface can be found in D5.13 [13].

Table 3-4: Interface with the modelling and simulation module.

Interface	Short description	Methods	Protocol
I1	RESTful web service interface	GET, POST, PUT, DELETE	HTTP(S)

3.2.3 Interface with external data and services

Apart from the data provided by HEIMDALL subsystems, the HEIMDALL workflows also involve externally available information by third party providers, such as e.g. weather data. For this purpose, the HEIMDALL SP implements service-specific interfaces as plug-ins which retrieve the information from the external service provider using the service provider's API, adapt it and feed it to the SP via the already provided open interfaces. The external data and services are accessible through the ESB, whereas more details are reported in D5.10 [12].

3.2.4 Interface with information gateway

The interface with the information gateway module is the I5 indicated in Table 3-5. More details about this interface can be found in D5.13 [13].

Table 3-5: Interface with the information gateway module.

Interface	Short description	Methods	Protocol
I5	RESTful web service interface	GET, POST, PUT, DELETE	HTTP(S)

3.2.5 Interface with the graphical user interface module

The interface with the HEIMDALL graphical user interface module is the I9, as indicated in Table 3-6. More details about this interface can be found in D4.7 [4].

Table 3-6: Interface with the GUI module.

Interface	Short description	Methods	Protocol
I9	RESTful web service interface	GET, POST, PUT, DELETE	HTTP(S)
I9	Websockets, RFC 6455	HTTP requests	HTTP

3.2.5.1 WebSockets Notification Service

SP pushes notifications to the GUI about events occurring at the back-end system by using Websockets. Thus, the GUI is being notified asynchronously about:

1. The completion of long running processes (e.g. simulations).
2. The occurrence of new hazards (e.g. new fire observation or incidents creation).
3. The availability of new data from various sources (e.g. new rapid mapping layers added into Geoserver).

The notifications have the structure below:

```
{
  "serviceid" : "SERVICEID",
  "event" : "EVENT",
  "resourceurl" : "RESOURCEURL",
   "userid" : []
}
```

`serviceid`: The service/component that the notification refers to. Values are:

- `alert`: a new alert received from an alerting source
- `simulator`: a simulation has been completed
- `position`: the position of an asset was changed/updated
- `content`: new content is available
- `export`: an export operation has been completed

`event`: The status of the event to which the notification refers to. Values are:

- `new`: new event
- `update`: update of an entity
- `delete`: delete of an entity
- `assign`: linking of two entities (e.g. observation to incident)
- `failed`: failure of a long running process (i.e. simulation)

`resourceurl`: The actual entity in the system that this notification refers to (can be a simulation/observation/incident/content/position or other).

`userid`: The IDs of users (list) that should be notified about this event according to their preferences/settings (area of interest, group etc.).

The websockets server listens at: `ws://192.168.127.3:9999/echo`

3.2.6 **Interface with the user and role management module.**

The interface with the user and role management module in the I10 indicated in Table 3-7. More details about this interface can be found in D4.5 [3].

Table 3-7: Interface with the UeRM module.

Interface	Short description	Methods	Protocol
I10	RESTful web service interface	GET, POST, PUT, DELETE	HTTP(S)

3.2.7 **Interfaces to other Local Units**

The SP is able to communicate with other Local Units by making full usage of the HEIMDALL Communication Package.

Through the HEIMDALL Data and Service Catalogue, part or whole of the SP services and products are available to other Local Units within the same region, country as well as in cross-border events.

4 Module Functionality

4.1 Data repository / GIS service and plain data service

The data repository of the SP essentially comprises of two main components: a repository for geospatial data (GIS service) and a repository for generic platform information.

The geospatial data repository is an implementation of a GIS service. It allows the publication and retrieval of both raster and vector data via open standardised interfaces, mostly WMS and WFS (also supporting REST communication), enabling various heterogeneous services and users to share, process and edit geodata. Data conversion is also possible among main popular formats, such as shapefiles, GML/KML, GeoTIFF and GeoJSON. Stored vector data can also be internally digitised and retrieved as raster, allowing simpler client and user GUI implementations, since the need for rendering at the GUI is minimised.

Within the repository, data is hierarchically organised, so that they can be easily categorised according to both the source and the nature of information. Examples of HEIMDALL data to be published to and retrieved from the geospatial repository are EO images (raw or processed), incident (fire, flood etc.) fronts and perimeters, base map layers, vegetation maps and simulation results (risk/hazard assessment maps, incident evolution/propagation curves).

The generic platform repository is used for information that does not fit in the geospatial database. It is implemented via a common relational database with a data access layer and a REST interface front-end. Examples of HEIMDALL data to be published to and retrieved from the generic platform repository are incident information, user data and miscellaneous platform and service configuration parameters.

The Database Management System (DBMS) used for hosting the databases is PostgreSQL [19]. PostgreSQL is an open-source relational database, available for most operating systems. It is being developed by a strong community for more than two decades and is based on a proven architecture. The schema has been designed to accommodate all the needs of HEIMDALL and store both geo-referenced and plain data.

For the georeferenced data, PostGIS [20] is used additionally. PostGIS is a spatial database extender for PostgreSQL object-relational database. It adds support for geographic objects allowing location queries to be run in SQL.

The GIS engine is built on GeoServer [21]. GeoServer is a popular open source server used to share geospatial data. It is designed for interoperability, handling data from any major spatial data source using open standards. Among others, it is the reference implementation for several OGC standards, including WFS, WCS and WMS, which are heavily used in HEIMDALL.

With the help of the following services, the user shall be able to:

1) Retrieve cartographic data from a web map service (WMS)
2) Retrieve map features from a web feature service (WFS)
3) Retrieve coverage information from a web coverage service (WCS)
4) Submit coverages and features to the geodata repository
5) Submit events and observations to the geodata repository
6) Retrieve/submit data to a general purpose (non-georeferenced) data repository

Table 4-1: SP data management services.

Products and/or Services	**Inputs needed** *inputs to generate each output*	**Provided by** *module or external system providing the input*	**Used by** *module consuming the product/service*

Georeferenced data storage service	An OGC compliant server allowing users to view and edit geospatial data.	• Simulators • External systems • Scenario management	• UI • Scenario management • Information gateway • Decision support • Impact summary generation
GIS database	A database that allows the storage of spatial and geographic information and execution of location-based information retrieval.	• Simulators • External systems Scenario management	• UI • Scenario management • Information gateway • Decision support Impact summary generation
“Plain” data storage service	A web service allowing the storage and retrieval of generic data and/or documents to the corresponding database(s).	• External systems • Scenario management	• UI • Scenario management • Information gateway • Decision support Impact summary generation
“Plain” database	A database that allows the storage and retrieval of plain data information.	• External systems Scenario management	• UI • Scenario management • Information gateway • Decision support Impact summary generation
Historic data service	Provide access to historic data (past incidents)	• Simulators • External systems Scenario management	• UI • Scenario management • Information gateway • Decision support Impact summary generation

4.2 Enterprise service bus (ESB)

The role of the Enterprise Service Bus (ESB) within the HEIMDALL SP is to promote agility and flexibility regarding the communications among the different HEIMDALL subsystems. Especially, when it comes to control/invocation messages, rather than storage and retrieval of data, which is already handled by the data repository, as aforementioned. Thanks to the ESB, each HEIMDALL subsystem does not have to directly interface with each of the other subsystems (e.g. the Decision Support System - DSS with the simulator) for the communication of control and data messages. Instead, it interfaces solely with the ESB component of the SP that adapts and forwards the message appropriately to its destination. Furthermore, the operation of each subsystem is not blocked due to reduced availability of the peer subsystem, as all messages are buffered/queued until the recipient becomes available.

In order to serve this role, the ESB makes use of an integration broker (middleware), which provides an abstraction layer on top of a messaging system. The ESB provides the following core services:

- multi-interface communication (REST, SOAP, Sockets, FTP, e-mail, etc.)
- routing of messages among different subsystems/components
- data transformation, protocol conversion
- message queuing
- message sequencing
- support of service registration / subscription / discovery

In HEIMDALL, the Enterprise Service Bus is implemented around a common message bus/ message queue that interfaces with several protocol adapters whose role is to translate external messages and push/pull them to/from the queue. The role of the ESB is to act as the central hub enabling the communication among all HEIMDALL subsystems and components. Within the SP, the ESB communicates with the data repositories for storing and retrieving data and with the service orchestration for communicating workflow commands and results. It inherits most of its concepts from widely used ESBs, such as Apache Camel, WSO2 and Microsoft BizTalk. However, instead of employing an already available ESB platform, it has been decided within the consortium to build a tailored ESB module dedicated to HEIMDALL. The main reason was that it was anticipated that, during the system integration, there would be a need for significant customisation in terms of protocol plug-ins and therefore, it would be desirable to have better control over the ESB functionality and code. The HEIMDALL ESB was developed in C# using the Microsoft .NET framework. The following modules/libraries were used for the ESB development:

- ActiveMQ [22] as the main message broker
- OAuth [23] for access control (authentication and authorisation)

The ESB is an application running continuously in the background, offering a real-time log capturing the messages exchanged.

5 Technical Specification

The entire HEIMDALL platform as well as the SP functionality is only accessible from within the HEIMDALL VPN. Therefore, in order to test the functionality presented in the following subsections the users should have access to the HEIMDALL VPN.

The main extensions of the technical features compared to the ones presented in D4.1 are:

- Support for flood simulations
- Support for landslide simulations
- Support for impact management
- Support for information from the drones subsystem
- Support for INSPIRE metadata
- Addition of Registry service
- Addition of Chat server and messaging service
- Addition of Catalogue service
- Addition of map helper functions service

5.1 User login service API

In order for any user or application to be able to interact with the HEIMDALL SP a successful login has to be performed, as presented in Table 5-1.

Table 5-1: The SP login service

Service ID	**SP_login_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User name and password
Operations	N/A
Main parameters	User name and password
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	JWT token and expiration data (JSON format)
Notes	Without a successful login operation the SP will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

```
POST http://esb.heimdall.sp/services/rest/login
```

Where the user or application has to provide a JSON file with the following format:

```
{
"UserName" : "JohnDoe",
"Password" : "Password"
}
```

And receive the following response, which includes the token and its expiration date and time:

```
{
```

```
    "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6ImNyYyIsImh0dHA6L
y9zY2hlbWFzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvaWRlbnRpdHkvY2xhaW1zL3NpZCI6IjM2
MDQ4NjA0LTQzNzUtNDRjZC04M2E2LTVjZTIwMzE3NzViNiIsInJvbGUiOiJDb250cm9sIFJvb20
gQ2hpZWYiLCJwcmltYXJ5c2lkIjoiYmE2YmMxOTctMTZhZC00Yjg3LTlhMWYtOWRkM2FjNDdkM2
FkIiwibmJmIjoxNTM3Nzc3NjMwLCJleHAiOjE1Mzc4NjQwMzAsImlhdCI6MTUzNzc3NzYzMH0.X
bHOXvdjZ8ZimGN7pX1zPSyqXDLrS0-lGVS5IVIx6S0",
    "expires": "20180925T080910"
}
```


5.2 Map and layer management API

This section describes the different data exchange services available, which allow interaction with the data repositories hosted by the service platform.

5.2.1 Web Map Service (WMS)

The OpenGIS® Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined if needed. For full specification, you can visit the following URL http://www.opengeospatial.org/standards/wms. The Web Map Service offers a base map of cartographic data as a common reference layer for superimposing all HEIMDALL-generated information.

5.2.1.1 Retrieving map layers

Map Layers are being served through WMS (Web Map Service (http://docs.geoserver.org/latest/en/user/services/wms/reference.html) ready to be consumed by a map client. You have to use an appropriate client library in order to request and render map tiles from services like the above (i.e. openlayers or leaflet.js). The following tables and examples illustrate this functionality.

Table 5-2: Retrieving map layers.

Service ID	**SP_layers_01**
Assumed consumers (via reference point)	I9, I8, I3, I4, I1, I6, ESB
Data exchanged	Raster map data for a specific region
Operations	GetCapabilities, DescribeLayer, GetMap
Main parameters	Bounding box coordinates, spatial reference system, resolution, output format
Data representation protocol	GeoTIFF/JPG/PNG
Communication protocol	HTTP (GET)
Response	Map data (Image)
Notes	-

The following example shows a GET request and provides details about its structure.

```
GET
http://esb.heimdall.sp/services/ogc/<namespace>/wms?service=WMS&version=1.1.0&request=GetMap&layers=<layer>&styles=&bbox=<coordinates>&width=512&height=433&srs=EPSG:4326&format=<imageformat>


<namespace>: The namespace that the requested layer belongs to

<layer>: The name of the map layer

<coordinates>: WGS84 coordinates of the cropped map requested in terms of left-bottom corner, top-right corner

<imageformat>: The file type of the result. Valid values are: image/gif, image/png, image/jpg, image/svg


To run an example, from within the HEIMDALL VPN, use the following

http://esb.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.0&request=GetMap&layers=gois&styles=&bbox=1.5308649890000652,42.008436569000025,1.6366447990000665,42.09794787200003&width=512&height=433&srs=EPSG:4326&format=image/gif
```

Some of the most important static (and dynamic but with `const` name) layers (available on M18) are:

```
heimdall:gois

heimdall:airports

heimdall:municipis

heimdall:firegrowth

heimdall:firerisk

heimdall:hazard

heimdall:impact

heimdall:lakes

heimdall:highways

heimdall:railways

heimdall:rivers

heimdall:virtuals
```

A list of available layers can be requested through a `GET` operation (Table 5-3 and following examples):

Table 5-3: Fetching all layers

Service ID	**SP_layers_02**
Assumed consumers (via reference point)	I9, I8, I3, I4, I1, I6, ESB
Data exchanged	Layer details

Operations	N/A
Main parameters	layertype
Data representation protocol	JSON
Communication protocol	HTTP(GET)
Response	JSON
Notes	-

```
GET http://esb.heimdall.sp/services/rest/layers
```

The above call returns all layers registered in SP. With `all=false` it will return only the layers that are available (configured as accessible by GUI users). The main parameters are as follows:

`layertype`: Type of the layer. Can be `MapLayer`, `VectorLayer`, `RasterLayer` depending on the nature of the data in Geoserver.

- `MapLayer` is generic and it means that you can consume that layer through WMS.
- `VectorLayer` is for features that can be consumed either by WMS or by WFS and
- `RasterLayer` is for raster images that can be consumed through WMS or WFS

The response following a valid GET request to fetch all the map layers is the following. Note the results of this request depicts the status of the platform during the Release A period and will differ based on the number of layers stored in the platform and coupled to the running scenario. The following response shows that the HEIMDALL platform has already integrated the layers developed within the previous PHAROS platform.


```
{
    "Layers": [
        {
            "wms": "http://esb.heimdall.sp/services/ogc/pharos/wms",
            "name": "pharos:firerisks",
            "description": "Overall Firerisk",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/pharos/wms",
            "name": "pharos:modis",
            "description": "EO Image",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
```

```
            "layertype": "RasterLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/pharos/wms",
            "name": "pharos:highways",
            "description": "Highways",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": null,
            "name": "pharos:METRO",
            "description": "pharos:METRO",
            "isbase": false,
            "isexternal": true,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/pharos/wms",
            "name": "pharos:municipis",
            "description": "Municipalities",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/pharos/wms",
            "name": "pharos:aeroports",
            "description": "Airports",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
```

```
        {
            "wms": "http://esb.heimdall.sp/services/ogc/pharos/wms",
            "name": "pharos:railways",
            "description": "Railways",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name": "heimdall:gois",
            "description": "Heimdall First Review Virtual Objects",
            "isbase": false,
            "isexternal": false,
            "metadata": [
                {
                    "name": "CODI",
                    "type": "string",
                    "value": null
                },
                {
                    "name": "NAME",
                    "type": "string",
                    "value": null
                },
                {
                    "name": "ADDRESS",
                    "type": "string",
                    "value": null
                },
                {
                    "name": "USE",
                    "type": "string",
                    "value": null
                },
                {
                    "name": "SECTOR",
```

```
                    "type": "string",
                    "value": null
                }
            ],
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:sertit.lajonquera.heimdall.fireseverity.20120724",
            "description": "SERTIT Fire Severity - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:icubesertit.lajonquera.heimdall.impact_building.20120724",
            "description": "SERTIT Building Impact - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:sertit.lajonquera.heimdall.fireextend.20120724",
            "description": "SERTIT Fire Extend - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
```

```
            "name":
"heimdall:icubesertit.lajonquera.heimdall.impact_LULC.20120724",
            "description": "SERTIT LULC Impact - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:icubesertit.lajonquera.heimdall.impact_road.20120724",
            "description": "SERTIT La Jonquera Road Impact",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:sentintel1.dlr.heimdall.standing_water.20160904T000338",
            "description": "SENTINEL 1 Standing Water - Bangladesh",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:sentintel1.dlr.heimdall.flood.20160904T000338",
            "description": "SENTINEL 1 Flood - Bangladesh",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
```

```
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:S2A_MSIL2A_20170704T112111_N0205_R037_T29TNE.dlr.heimdall.class_b
urnscar.20170704T112431",
            "description": "SENTINEL 2 Burnscar - Portugal",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:pleiades.burnscar.heimdall.firedelineation.20120724",
            "description": "Pleiades Burnscar Fire Delineation - La
Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:modis.dlr.heimdall.firehotspot.20180617T210100",
            "description": "MODIS Fire Hotpot - Marseille",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name": "heimdall:terrasar-
x.dlr.heimdall.flood_compressed.20130612T052528",
            "description": "TERRASAR-X Flood - Berlin",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
```

```
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":                    "heimdall:jonquera.dlr.heimdall.3d-
building_model_example.20180913",
            "description": "3D Building Model - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name": "heimdall:jonquera.dlr.heimdall.admin_bounds.20180913",
            "description": "Admin Bounds - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":
"heimdall:jonquera.dlr.heimdall.transportation_example.20180913",
            "description": "Transportation Example - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name": "heimdall:jonquera.dlr.heimdall.lulc_example.20180913",
            "description": "LULC Example - La Jonquera",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
```

```
        },
        {
            "wms": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
            "name":                                    "heimdall:terrasar-
x.dlr.heimdall.standing_water_compressed.20130612T052528",
            "description": "TERRASAR-X Standing Water - Berlin",
            "isbase": false,
            "isexternal": false,
            "metadata": null,
            "layertype": "VectorLayer"
        }
    ]
}
```

5.2.1.2 Adding a map layer

Table 5-4: Adding a layer.

Service ID	**SP_layers_03**
Assumed consumers (via reference point)	I9, I8, I3, I4, I1, I6, ESB
Data exchanged	Layer details
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP(POST)
Response	HTTP STATUS CODE
Notes	-

The following REST call adds a new layer by reference (does not add the actual layer's data to the map server). If the layer has the flag `isexternal` set to false, it is assumed that the layer is already registered to the map server with a manual process by the SP administrator.

```
POST http://esb.heimdall.sp/services/rest/layers
{
  "name": "heimdall:traffic2",
  "description": "External Traffic 2",
  "isbase": false,
  "isexternal" : true,
  "metadata": [
                {
                    "name": "number",
```

```
                    "type": "int",
                    "value": 2
                },
                {
                    "name": "changedat",
                    "type": "dateTime",
                    "value": "2018-09-05T13:05:00"
                }
  ]
}
```

Metadata for layer are key/value pairs with one of the predefined types: `int, double, string,` and `dateTime`. If someone wishes to add a layer by value (both data and information about the layer), a multipart/form-data http message should be sent to the SP, including a JSON part (with name .json) and a value as the one described above, and a file part containing the actual data of the layer. The file may be a .zip file containing a shapefile layer or a geotiff image.

5.2.1.3 Updating a map layer

Updating an existing layer can be performed through a PUT operation.

Table 5-5: Updating a layer.

Service ID	**SP_layers_04**
Assumed consumers (via reference point)	I9, I8, I3, I4, I1, I6, ESB
Data exchanged	Layer details
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP(PUT)
Response	HTTP STATUS CODE
Notes	-

The following is an example of a map layer update request:

```
PUT http://esb.heimdall.sp/services/rest/layers
{
  "name" : "heimdall:traffic2",
  "description": "External Traffic Second",
  "isbase": true,
  "isexternal" : true,
  "metadata": [
```

```
                {
                    "name": "addtional label",
                    "type": "string",
                    "value": "This is an additional label"
                },
                {
                    "name": "changedat",
                    "type": "dateTime",
                    "value": "2018-09-05T13:05:00"
                }
  ]
}
```

5.2.2 **Web Coverage Service (WCS)**

A Web Coverage Service (WCS) offers multi-dimensional coverage data for access over the Internet. WCS Core specifies a core set of requirements that a WCS implementation must fulfil. For the full specification of WCS you can visit the following URL http://www.opengeospatial.org/standards/wcs. The Web Coverage Service offers raster data to HEIMDALL components. The WCS could be seen as an enhancement compared to WMS described in the previous section, in the sense that WCS can also provide multi-dimensional raster data (not only optical images) and data which evolve in time (i.e. adding the time dimension). In this context, the WCS can be used e.g. for retrieving EO images as well as rasterised simulation results. The following Table presents a summary of the web coverage service.

Table 5-6: SP Workflow triggering service specification.

Service ID	**SP_layer_05**
Assumed consumers (via reference point)	I9, I8, I3, I4, I1, I6, ESB
Data exchanged	Raster map data for a specific region, incl. MODIS Data
Operations	GetCapabilities, DescribeCoverage, GetCoverage
Main parameters	Bounding box coordinates, spatial reference system, resolution, output format, time (optional)
Data representation protocol	GeoTIFF / JPG / PNG
Communication protocol	HTTP (GET)
Response	Map data (Image)
Notes	-

Following are some sample URLs of the service:

```
GET
http://esb.heimdall.sp/services/ogc/wcs?REQUEST=GetCoverage&SERVICE=WCS&VERSION=1.0.0&COVERAGE=<namespace>:<layer>&BBOX=<coordinates>&CRS=EPSG:4326&WIDTH=575&HEIGHT=650&FORMAT=GEOTIFF
```

<namespace>: The namespace that the requested layer belongs to
<layer>: The name of the map layer
<coordinates>: WGS84 coordinates of the cropped map requested in terms of left-bottom corner, top-right corner

i.e.
http://esb.heimdall.sp/services/ogc/wcs?REQUEST=GetCoverage&SERVICE=WCS&VERSION=1.0.0&COVERAGE=heimdall:firegrowth&BBOX=1.3265870958235,41.870257575996,1.81440267180455,42.2325486012425&CRS=EPSG:4326&WIDTH=575&HEIGHT=650&FORMAT=GEOTIFF

Following the above request, the SP retrieves and returns the requested image (in GeoTIFF format) for the area specified.

5.2.3 **Web Feature Service (WFS)**

The Web Feature Service offers feature maps to HEIMDALL components. Feature maps normally include vector data (e.g. points, lines, polygons etc.) as opposed to raster data provided by the WCS and WMS services. The full specification of WFS is accessible in http://www.opengeospatial.org/standards/wfs. The following Table presents a summary of the web feature service.

Table 5-7: SP Workflow triggering service specification

Service ID	**SP_layer_06**
Assumed consumers (via reference point)	I9, I8, I3, I4, I1, I6, ESB
Data exchanged	Feature map data for a specific region
Operations	GetCapabilities, DescribeFeature, GetFeature
Main parameters	Bounding box coordinates, spatial reference system, resolution, time (optional)
Data representation protocol	GeoJSON / GML
Communication protocol	HTTP (GET)
Response	GeoJSON / GML
Notes	-

Following are some example URLs:

http://esb.heimdall.sp/services/ogc/<namespace>/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=<namespace>:<layer>&outputFormat=application/json

<namespace>: The namespace that the requested layer belongs to
<layer>: The name of the map layer

i.e.
http://esb.heimdall.sp/services/ogc/space/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=space:simulation&outputFormat=application/json&cql_filter=simulationid=%277e23ecd1-4606-4b3b-ac93-07c32fa48f37%27

RESPONSE

{

```
    "type": "FeatureCollection",
    "totalFeatures": 10,
    "features": [
        {
            "type": "Feature",
            "id": "simulation.40557",
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ]
                    ]
                ]
            },
            "geometry_name": "geom",
            "properties": {
                "fid": 40557,
                "hour": 0,
                "simulationid": "7e23ecd1-4606-4b3b-ac93-07c32fa48f37"
            }
        },......
```

5.3 Simulation APIs

This section describes the Simulation REST APIs as implemented within HEIMDALL's Service Platform.

5.3.1 Triggering a fire simulation

The user through the GUI or a terminal window requires starting a new simulation. The user will utilise the REST API for this. The following examples work with the La Jonquera scenario, for an area of 40x40Km around the La Jonquera fire affected area. More details about this scenario are provided in [13].

The minimum and maximum values for the simulation coordinates range are the following:

- MIN LATITUDE: 42,22837
- MAX LATITUDE: 42,48328
- MIN LONGITUDE: 2,71006
- MAX LONGITUDE: 3,05501

The following example presents the POST request to initiate a simulation and the response message that includes the simulation ID.

```
POST http://esb.heimdall.sp/services/rest/simulations
{
  "name" : "test",
  "description" : "lasttestnewer",
  "points" : [{ "longitude" : 2.852357, "latitude" : 42.415679 , "hour": 0}],
  "lines" : [ "vertices" : [{ "longitude" : 2.884769, "latitude" : 42.438694 }, { "longitude" : 2.884819, "latitude" : 42.438102 }, { "longitude" : 2.885446, "latitude" : 42.437908 }, { "longitude" : 2.884769, "latitude" : 42.438694 } ], "hour" : 0  ]
  "hours" : 10,
  "starttime" : "2018-08-27T13:15:40.000Z",
  "extentrows" : 500,
  "extentcolumns" : 500,
  "weather" :
         {
           "type" : "CUSTOM",
           "wind_speed" : 15,
           "wind_direction" : 30,
           "temperature" : 27,
           "humidity" : 25,
           "shadow" : 0,
           "moisturetype" : "CTE"
         }
}
```

- `name`: The name of the simulation
- `description`: A small description of what the simulation is about
- `points`: a list of ignition points. Currently only the first point is taken into account. Coordinates should be expressed in WGS84 system
- `lines`: a list of fire lines. Last point of the line must coincide with the first point.
- `hours`: Number of hours to be simulated.
- `extentcolumns`:. Generally, it must be within 250 and 1000. Default value is zero.
- `extentrows`:. Generally, it must be within 250 and 1000. Default value is zero.
- `type`: Weather service type. The options are: `CUSTOM` or the name of the weather service provider. In case of choosing a service, the weather values in this input will not be used in the simulation. When the users provide `CUSTOM` as the option, they would have to specify the weather parameters shown below.
    - `wind_speed`: Wind speed in meter seconds (m/s).
    - `wind_direction`: Wind direction in angle (0 to 360º).
    - `temperature`: Air temperature in centigrade degrees (ºC).
    - `humidity`: Air moisture in percentage (%).
    - `shadow`: Cloudiness (0 to 100).
    - `moisturetype`: Moisture type. If omitted default will be used. Valid values are CTE (constant) and ROTHERMEL.

```
RESPONSE:
{
  "SimulationId" : "ca03abae-6fca-40fa-94d9-9ec1e5780ada"
}
```

5.3.2 Triggering a Flood Simulation

Flood simulations can be triggered by a corresponding REST API. The REST API takes the input required by the flood simulator, commands the simulator to start computations with that input and finally fetches the results of the simulations, registers the results to its internal geodatabase and provides a handful of APIs to access those results.

Following is an example of starting a flood simulations:

```
POST http://esb.heimdall.sp/services/rest/floodsimulations
{
   "name":"testmulti",
   "params":{
      "resolution":20,
      "bounding_box":[
         9.327735900878906,
         44.33821471875343,
```

```
            9.379749298095703,
            44.36294902658837
         ],
         "w_speed":6,
         "duration":4,
         "discharge_peaks":[
            {
               "coordinates":[
                  9.35620451,
                  44.36288070
               ],
               "peak_value":840
            },
            {
               "coordinates":[
                  9.33425085,
                  44.36293044
               ],
               "peak_value":1460
            }
         ],
         "t_init":"2019-10-07T11:22",
         "saveInt":1800
      }
}
```


The required input parameters are:

- **name**: label for the simulation
- **resolution**: simulation resolution in meters
- **bounding_box**: limit of the simulation domain in geographic coordinates
- **w_speed**: speed of the river flow (m/s)
  duration: duration time of the simulation (hours)
- **discharge_peaks**: peak discharge value for the simulation (m^3/s) for each river stream
- **t_init**: initial time as timestamp
- **saveInt**: save interval for the simulation (seconds)

5.3.3 **Triggering a Landslide Simulation**

The last type of supported simulations are landslide simulations. Landslide simulations can be of four types: Rockfall, ,Debris, Landslide and Rainfall.

An example of the REST API call to initiate a landslide simulations follows.

```
POST http://esb.heimdall.sp/services/rest/landslidesimulations/requests
{
     "name" : "THE NAME OF THE SIMULATION",
     "simulationtype" : "SIMULATION TYPE",
     "rockfalldebris" :
     {
        "study_area_rect" : [COORDINATES OF THE STUDY AREA RECTANGULAR],
        "sources_geometry" :
        [[
            [POLYGON A POINT A], [POLYGON A POINT B], .... [POLYGON A POINT
A],
            [POLYGON B POINT A], [POLYGON B POINT B], .... [POLYGON B POINT
A],
            ....
        ]],
        "operation": "TYPE OF OPERATION",
        "size": 1,
        "precision": 1,
        "materialtype" : 1
    },
    "landslide" :
    {
        "study_area_rect" : [COORDINATES OF THE STUDY AREA RECTANGULAR],
        "soilhumidity" : "1",
        "soiltype" : "1",
        "landslide_size" : "1"
    }
    "rainfall" :
    {
        "lon" : 2.453431115,
        "lat" : 40.2311123,
        "threshold_climatic_area" : 1
```

```
        "simulationdate" : "2019-03-08"
    }
}
```

The input parameter of a landslide simulation are:

- **simulationtype** : SimulateRockfall/SimulateDebris/SimulateLandslide/SimulateRainfall
- **operation** : rockfall/debris_flow
- **size** : 1/2/3 (small/medium/large) for Rockfall / 1/2 (small-medium/large) for Debris flow
- **precision** : 1/2 (high/medium)
- **materialtype** : Should be set for debris_flow operation, 1/2/3 (coarse,coarse_and_fine,fine)
- **soilhumidity** : 1/2/3/4 (dry/low/medium/wet)
- **soiltype** : 1/2/3/4 (gravel/sand/silt/clay)
- **landslide_size** : 1/2/3/4 (very big/big/medium/small)--> Size for Rotational Landslide
- **threshold_climatic_area** : Depending on the location of the event the user will select between: 1: Warm Mediterranean climate/2: Cool Mediterranean climate/3: Mountain climate

5.3.4 **Fetching simulation results**

The results of a simulation can be fetched with a `GET` like in the example below. Keep in mind that the simulation is a long running process, so the results will not be available immediately.

```
GET http://esb.heimdall.sp/services/rest/simulations?simulationId=ca03abae-6fca-40fa-94d9-9ec1e5780ada
```

Returns:


```
{
    "Title": "test - lasttestnewer",
    "RequestJson": null,
    "InnerSimulationId": "ca03abae-6fca-40fa-94d9-9ec1e5780ada",
    "SimulationId": "7e23ecd1-4606-4b3b-ac93-07c32fa48f37",
    "AdjustmentId": "00000000-0000-0000-0000-000000000000",
    "Mode": 1,
    "HasError": false,
    "FirePerimeterUrl":
"http://esb.heimdall.sp/services/ogc/space/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=space:simulation&outputFormat=application/json&cql_filter=simulationid=%277e23ecd1-4606-4b3b-ac93-07c32fa48f37%27",
    "ArrivalTimeUrl": null,
    "WeatherJson": null,
    "Incident": null,
    "Incident_Id": 0,
    "DssSimulationId": 0,
```

```
    "StartTime": "0001-01-01T00:00:00",
    "RequestTime": "2018-07-17T11:23:21",
    "FinishTime": "2018-07-17T11:24:38",
    "FirePerimeterWmsUrl":
"http://esb.heimdall.sp/services/ogc/space/wms?service=WMS&version=1.1.0&request=GetMap&cql_filter=simulationid%3d'7e23ecd1-4606-4b3b-ac93-07c32fa48f37'&layers=space:simulation&styles=perimeter_style&srs=EPSG:4326&format=image%2fgeotiff&width=512&height=512&bbox=-2.0%2c38.0%2c4.0%2c43.0",
    "ControlPoints": null,
    "LinkedSimulationId": 0,
    "UserId": null,
    "Id": 14277
}
```


In case the results are not yet available `FirePerimeterUrl` will be null. `FirePerimeterUrl` is the most important property. It points to the actual results of the simulation.

```
GET http://esb.heimdall.sp/services/ogc/space/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=space:simulation&outputFormat=application/json&cql_filter=simulationid=%277e23ecd1-4606-4b3b-ac93-07c32fa48f37%27
```

The response message is as the following example. This is a GeoJson including polygons for each hour of propagation (hour0 ... hourN where N is the requested `hours` simulation parameter). For full simulation result sample check: https://redmine.space.gr/attachments/797/simulation_output.json.


```
{
    "type": "FeatureCollection",
    "totalFeatures": 10,
    "features": [
        {
            "type": "Feature",
            "id": "simulation.40557",
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
```

```
                        ],
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ],
                        [
                            -116.6927,
                            32.89237
                        ]
                    ]
                ]
            },
            "geometry_name": "geom",
            "properties": {
                "fid": 40557,
                "hour": 0,
                "simulationid": "7e23ecd1-4606-4b3b-ac93-07c32fa48f37"
            }
        },.....
```

In order to fetch the results of a landslide simulation the following REST call can be used.

GET http://esb.heimdall.sp/services/rest/landslidesimulations?id=ID

This call returns in example:

```
{
     "name" : "THE NAME OF THE SIMULATION",
     "simulationtype" : "SIMULATION TYPE",
         "request" : <THE INITIAL REQUEST JSON OF THIS SIMULATION>,
     "rockfalldebris" :
     {
        "study_area_rect" : [COORDINATES OF THE STUDY AREA RECTANGULAR],
        "sources_geometry" :
        {
            "type": "Polygon",
            "coordinates": [[
```

```
            [POLYGON A POINT A], [POLYGON A POINT B], .... [POLYGON A POINT
A],
            [POLYGON B POINT A], [POLYGON B POINT B], .... [POLYGON B POINT
A],
            ....
            ]]
        },
        "operation": "TYPE OF OPERATION",
        "size": 1,
        "precision": 1,
        "materialtype" : 1
    },
    "landslide" :
    {
        "study_area_rect" : [COORDINATES OF THE STUDY AREA RECTANGULAR],
        "soilhumidity" : 1,
        "soiltype" : 1,
        "landslide_size" : 1
    }
    "rainfall" :
    {
        "lon" : 2.453431115,
        "lat" : 40.2311123,
        "threshold_climatic_area" : 1,
        "simulationdate" : "2019-03-08"
    },
        "products" :
        [
            {
               "type" : "vector",
               "url" : "http://esb.heimdall.sp/services/ogc/...."
            }
        ]
}
```

The response of the API contains the parameters of the simulation request and an additional section named “products”, which contains a list of the layers produced by the simulator. **Type** can be either vector or raster and **url** points to the location of the layer’s data.

Similarly in order to fetch the results of a flood simulation the API below can be used:

GET http://esb.heimdall.sp/services/rest/floodsimulations?id=ID

```
{
    "id": 35,
    "name": "flood in italy 3587",
    "status": 5,
    "type": "flood",
    "request" : <THE INITIAL REQUEST JSON OF THIS SIMULATION>,
    "products": [
        {
            "name": "Flood (h)",
            "hours": [
                {
                    "hour": 0,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.0&request=GetMap&layers=heimdall:flood_YmRhYzg1MzktNzAyOC00ZGJjLWE4ZGItOWFhMWMxYWQyNDJh&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.347455097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transparent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3322294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
                },
                {
                    "hour": 1,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.0&request=GetMap&layers=heimdall:flood_Nzc5OThkYzAtNmFhZi00ZTQxLTg4YjUtNmRmNjdhMzJjNjNk&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.347455097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transparent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3322294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
```

```
                },
                {
                    "hour": 2,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.
0&request=GetMap&layers=heimdall:flood_MTBhMWY0ZWItOTEwYi00NDNiLWFhZDEtM2Zk
M2FmMjdiYTVi&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.34745
5097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transp
arent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3322294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
                },
                {
                    "hour": 3,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.
0&request=GetMap&layers=heimdall:flood_YmM5ZDE5ZDktMzcyYS00ODA5LTk4MDItOTBj
M2Q2YmY1ZjU3&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.34745
5097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transp
arent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3322294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
                },
                {
                    "hour": 4,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.
0&request=GetMap&layers=heimdall:flood_MGMyZWEyNDUtNzc2Ny00ZGM5LWFjYjktY2Fh
ODNjMDI3MTEy&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.34745
5097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transp
arent=true",
                    "layertype": "raster",
```

```
                    "bbox": [
                        9.3222294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
                }
            ]
        },
        {
            "name": "Flood (hmax)",
            "hours": [
                {
                    "hour": 0,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.0&request=GetMap&layers=heimdall:flood_MDIzNzU3MjctOTc4Mi00MDQ2LThhOGYtMzk2ODgzNjdiN2Mz&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.347455097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transparent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3222294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
                },
                {
                    "hour": 1,
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.0&request=GetMap&layers=heimdall:flood_OTAzYzYxYTItZDU3Yy00YjkzLTg1NzctZGU2Y2NhMGVhMGQ5&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.347455097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transparent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3222294199868168,
                        44.347455097958495,
                        9.3478726494573134,
```

```
                    44.355534568124575
                ]
            },
            {
                "hour": 2,
                "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.
0&request=GetMap&layers=heimdall:flood_NGRmN2VlOTctODJhZS00ZjFkLWFiNWYtNzQ3
YmVlNTYwYmQ5&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.34745
5097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transp
arent=true",
                "layertype": "raster",
                "bbox": [
                    9.3322294199868168,
                    44.347455097958495,
                    9.3478726494573134,
                    44.355534568124575
                ]
            },
            {
                "hour": 3,
                "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.
0&request=GetMap&layers=heimdall:flood_NmFiZTg2NGYtYTlmOC00ZjFhLThiNWMtOWJk
NTFmYjYxMGVj&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.34745
5097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transp
arent=true",
                "layertype": "raster",
                "bbox": [
                    9.3322294199868168,
                    44.347455097958495,
                    9.3478726494573134,
                    44.355534568124575
                ]
            },
            {
                "hour": 4,
                "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/wms?service=WMS&version=1.1.
0&request=GetMap&layers=heimdall:flood_NzlkZjM1M2ItZmU5ZC00MDcwLWI1YmQtZmU1
OWUwM2QyMmE5&srs=EPSG:4326&format=image/png&bbox=9.332229419986817,44.34745
```

```
5097958495,9.347872649457313,44.355534568124575&WIDTH=273&HEIGHT=141&transp
arent=true",
                    "layertype": "raster",
                    "bbox": [
                        9.3322294199868168,
                        44.347455097958495,
                        9.3478726494573134,
                        44.355534568124575
                    ]
                }
            ]
        },
        {
            "name": "Flood - Vector Layer",
            "hours": [
                {
                    "url":
"http://esb2.heimdall.sp/services/ogc/heimdall/ows?service=WFS&version=1.0.
0&request=GetFeature&typeName=heimdall:flood_956de7e2-2eb1-4fc7-9e05-
74ff286c4bf2&outputFormat=application/json",
                    "layertype": "vector"
                }
            ]
        }
    ]
}
```


The products list at the above consists of two major sections, one including the simulation results per simulated time interval showing the height of the water and another one including the maximum height of the water per simulation time interval.

5.4 Impact Assessment API

The Impact Assessment module computes the impact of a simulated hazard to the basic infrastructures of a region (buildings, roads) giving insight to the human and economic impact that this hazard could potentially cause.

Following is an example of an API call to initiate an impact assessment.

POST http://esb.heimdall.sp/services/rest/impactassessments

```
{
    "name" : "name of assessment",
```

```
    "simulationURI" :
"http://esb.heimdall.sp/services/landslidesimulations?id=2"
    "landslideid" : 2,
    "floodid" : null,
    "exposurelayername" : "heimdall:building_model_monesi",
    "exposurelayertype" : "buildings"
}
```


The input is self-explanatory. There are mainly two things required to start and impact analysis, and those are the output of a simulator (hazard layer) and the layer containing the assets that we want to calculate the impact upon (exposure layer).

The response of the impact analysis process can be fetched by a REST API as well.

GET http://esb.heimdall.sp/services/rest/impactassessments?id=ID


```
{
    "id" : 1,
    "name" : "name of assessment",
    "landslideid" : 2,
    "exposurelayername" : "heimdall:building_model_monesi",
    "status" : "Completed",
    "wfsurl" :
"http://esb.heimdall.sp/services/ogc/heimdall/ows?service=WFS&version=1.0.0
&request=GetFeature&typeName=heimdall:assessmentname&outputFormat=applicati
on/json",
    "prettywfsurl" :
"http://esb.heimdall.sp/services/rasorapi/impact_1231231231/"
}
```


That response contains, apart from the input parameters, two urls that point to the layer resulted from the impact analysis.

5.5 Asset management API

The HEIMDALL SP facilitates the management of assets being responders or equipment from the various modules of the system through the corresponding API. The user is able to retrieve the status of the assets from the SP, as well as add new ones, update existing ones and perform delete operations. The following subsections present the API.

5.5.1 Fetching assets

The user (being a human operator or another HEIMDALL module/service) can fetch all assets from the SP through the service summarised in Table 5-8.

Table 5-8: Fetching all assets from the SP.

Service ID	**SP_asset_01**
Assumed consumers (via reference point)	I9, I4, I8, I5, ESB
Operations	N/A
Main parameters	N/A

Message representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	The service returns all registered assets in the platform and their timestamped positions.

The following examples present the GET request and the response as provided by the HEIMDALL SP.

```
GET http://esb.heimdall.sp/services/rest/assets
```

```
RESPONSE


[
    {
        "Id": 9,
        "Username": "graf01",
        "Name": "Fire Analyst Coordinator",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "b0cdd89c-add4-4a51-a607-16258446477c",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 12,
        "Username": "crc",
        "Name": "Control Room Chief",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "36048604-4375-44cd-83a6-5ce2031775b6",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 13,
        "Username": "do",
```

```
        "Name": "Dispatcher Operator",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "79a9eb3b-cf3c-4240-81d8-72ecc90f5c54",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 14,
        "Username": "ic",
        "Name": "Incident Commander",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "cb6f3f83-030a-4bd4-a3b4-85471bc0f29e",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 15,
        "Username": "fr-mmee",
        "Name": "Police Department Mossos d’Esquadra. Autonomic Police Force
of Catalonia",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "7ee1372c-9c2d-47d4-aa7f-afd9cfca8880",
        "Type": "FirstResponder",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 17,
        "Username": "fr-lp",
        "Name": "Police Department Local Police",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "636f8315-5447-4c3f-91c4-1ca72430bb10",
        "Type": "FirstResponder",
```

```
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 18,
        "Username": "cra",
        "Name": "Alarm Reception Centre (Local)",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "bc79e252-d51a-4ccb-aacb-cca8a9b3f92d",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 7,
        "Username": "angel",
        "Name": "Angel Grablev",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "f0a19e04-301d-47af-a5bc-bed50d17d254",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 19,
        "Username": "fr-b",
        "Name": "Fire Service",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "a1478596-3a6e-478e-81ad-afcf93548d95",
        "Type": "FirstResponder",
        "Longitude": -0.1027215,
        "Latitude": 51.5131835,
        "Positions": null
    }
]
```

Through the service, summarised in Table 5-9, the user is able to retrieve the assets based on their TYPE.

Table 5-9: Fetching assets from the SP, based on their TYPE.

Service ID	**SP_asset_02**
Assumed consumers (via reference point)	I9, I4, I8, I5, ESB
Operations	N/A
Main parameters	TYPE
Message representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	The service returns all registered assets of the requested TYPE in the platform and their timestamped positions.

The following examples present the GET request and the response as provided by the HEIMDALL SP.

`GET http://esb.heimdall.sp/services/rest/assets?type=TYPE,`

Where `TYPE` in `(Drone, User, FirstResponder)`. For example, when the `TYPE` is `User`, we get the following response.

```
RESPONSE


[
    {
        "Id": 9,
        "Username": "graf01",
        "Name": "Fire Analyst Coordinator",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "b0cdd89c-add4-4a51-a607-16258446477c",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 12,
        "Username": "crc",
        "Name": "Control Room Chief",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
```

```
        "UserId": "36048604-4375-44cd-83a6-5ce2031775b6",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 13,
        "Username": "do",
        "Name": "Dispatcher Operator",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "79a9eb3b-cf3c-4240-81d8-72ecc90f5c54",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 14,
        "Username": "ic",
        "Name": "Incident Commander",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "cb6f3f83-030a-4bd4-a3b4-85471bc0f29e",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    },
    {
        "Id": 18,
        "Username": "cra",
        "Name": "Alarm Reception Centre (Local)",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "bc79e252-d51a-4ccb-aacb-cca8a9b3f92d",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
```

```
    },
    {
        "Id": 7,
        "Username": "angel",
        "Name": "Angel Grablev",
        "GroupId": "1f3fcdf9-b4ee-45df-8810-3456ca01a48a",
        "UserId": "f0a19e04-301d-47af-a5bc-bed50d17d254",
        "Type": "User",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "Positions": null
    }
]
```


The user, apart from the TYPE of the asset, is able to narrow down the response received by providing an area of interest, hence limit the scope of the search the SP will performed, as summarise in Table 5-10.

Table 5-10: Fetching assets from the SP, based on their TYPE and if they are within the area of interest.

Service ID	**SP_asset_03**
Assumed consumers (via reference point)	I9, I4, I8, I5, ESB
Operations	N/A
Main parameters	TYPE, AREAOFINTEREST
Message representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	The service returns all registered assets of the requested TYPE in the platform and their timestamped positions only if they are within the area of interest.

`GET http://esb.heimdall.sp/services/rest/assets?type=TYPE&bbox=AREAOFINTEREST`, where `AREAOFINTEREST` defines the extent (rectangle in terms of west, south, east, and north) expressed at WGS84 coordinate system.

5.5.2 Adding, modifying and deleting assets

Through the service presented in Table 5-11, the user is able to add a new asset in the SP database.

Table 5-11: Adding a new asset.

Service ID	**SP_asset_04**

Assumed consumers (via reference point)	I9, I4, I8, I5, ESB
Operations	N/A
Main parameters	Name, Longitude, Latitude, Type
Message representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP STATUS CODE
Notes	The service adds an asset to the list maintained in the SP.

The following example show the POST request and provide a sample JSON file.

```
POST http://esb.heimdall.sp/services/rest/assets
```

The module/user has to provide in a JSON file, the name of the asset, its location (longitude and latitude values in WGS84) and its TYPE. An example of the JSON file used is the following:

```
{
    "Name": "Test Drone",
    "Longitude": 2.555,
    "Latitude": 41.333,
    "Type": "Drone"
}
```

Through the service presented in Table 5-12, the user (operator or another HEIMDALL module/service) is able to update the asset location in the SP database.

Table 5-12: Updating an existing asset.

Service ID	**SP_asset_05**
Assumed consumers (via reference point)	I9, I4, I8, I5, ESB
Operations	N/A
Main parameters	Asset ID, Longitude, Latitude
Message representation protocol	JSON
Communication protocol	HTTP (PUT)
Response	HTTP STATUS CODE
Notes	The service adds an asset to the list maintained in the SP.

The following example show the PUT request.

```
PUT http://esb.heimdall.sp/services/rest/assets
```

An example of the JSON file used is

```
{
    "Id" : 6,
    "Longitude": 2.155,
    "Latitude": 40.333,
}
```

Through the service presented in Table 5-13, the user is able to delete asset from the SP database.

Table 5-13: Deleting an existing asset.

Service ID	**SP_asset_06**
Assumed consumers (via reference point)	I9, I4, I8, I5, ESB
Operations	N/A
Main parameters	Asset ID
Message representation protocol	JSON
Communication protocol	HTTP (DELETE)
Response	HTTP STATUS CODE
Notes	The service adds an asset to the list maintained in the SP.

```
DELETE http://esb.heimdall.sp/services/rest/assets/6
```

5.6 Drones API

The Drones API support the feeding of information generated by a drones' platform at the field. The information gathered by the drones are mainly geo-located images and hotspots in the case that the drones are equipped with the corresponding sensors.

SP provides 3 API endpoints to ingest that information.

```
POST http://esb.heimdall.sp/services/rest/drones/position
```


```
{"id": 6, "height": 322.2694510183659, "latitude": 1.8084789611000007,
"timestamp": "2019-03-13 13:09:36.723676", "longitude": 41.67026354255337}
```


The message above is being used to send the position of the drone.

```
POST http://esb.heimdall.sp/services/rest/drones/hotspot
```


```
{"temperature": 3000.0, "id": 6, "drone": 6, "height": 409.1601339285714,
"longitude": 41.672884273900124, "thermal_image": "serializedimagebytes",
"visual_image": "serializedimagebytes", "latitude": 1.8089331391774406,
"timestamp": "2019-03-13 12:54:56.107070"}
```


The above message contains information about a detected hotspot.

```
POST http://esb.heimdall.sp/services/rest/drones/image
```

```
{"temperature": 3000.0, "id": 6, "drone": 6, "height": 409.1601339285714,
"longitude": 41.672884273900124, "thermal_image": "serializedimagebytes",
"visual_image": "serializedimagebytes", "latitude": 1.8089331391774406,
"timestamp": "2019-03-13 12:54:56.107070"}
```


This message is being used to upload an photo taken by the drone to the SP.

5.7 Scenario management API

Through this API the functionality of the scenario management module is exposed to the other components of HEIMDALL. This is done through a REST API allowing for POST/GET/PUT/DELETE operations to the following endpoint http://esb.heimdall.sp/services/sm/[scenario|risk|measure....]/{id}

5.7.1 Create Scenario

Through the service presented in Table 5-14, the user is able to create a new scenario.

Table 5-14: Scenario creation.

Service ID	**SP_scenario_01**
Assumed consumers (via reference point)	I8
Operations	Creation of a new scenario
Main parameters	N/A
Message representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP STATUS CODE (201 upon successful execution)
Notes	

The following shows a valid POST request that has to be performed by the user in order to create a scenario.

```
POST http://esb.heimdall.sp/services/sm/scenario/
{
  "casualties": 0,
  "conditions": [
    {
      "humidity": 25,
      "id": 1,
      "resource_uri": http://esb.heimdall.sp/services/sm/condition/1/,
      "temperature": 27,
      "winddirection": 360,
      "windspeed": 15,
```

```
      "datetime": "2012-07-23T11:00:00"
    }
  ],
  "credibility": 10,
  "hazardlocation": {
    "coordinates": [
      2.87635353,
      42.635262626
    ],
    "type": "Point"
  },
  "hazardtime": "2012-07-22T10:20:00",
  "hazardtype": {
    "id": 1,
    "name": "forest fire",
    "resource_uri": http://esb.heimdall.sp/services/sm/hazardtype/1/
  },
  "impact": {
    "id": 1,
    "name": "1",
    "resource_uri": http://esb.heimdall.sp/services/sm/impact/1/
  },
  "injured": 0,
  "lessonslearnt": [],
  "name": "Dry Run 1 Test Scenario",
  "ongoing": true,
  "responseplans": [],
  "risklevel": {
    "id": 1,
    "name": "very low",
    "resource_uri": http://esb.heimdall.sp/services/sm/risklevel/1/
  },
  "scenarios": [],
  "type": {
    "id": 2,
    "name": "simulated",
    "resource_uri": http://esb.heimdall.sp/services/sm/scenariotype/2/
  }
```

```
}
```

The order of elements is not important. All enumerations and more complex sub-elements such as "type", "hazardtype", "impact" and "risklevel" have been defined as related data structures. As an alternative to using the full sub-element body, you can specify the sub-resource URI only. For example, "risklevel" could be also specified as

```
{
  "risklevel": http://esb.heimdall.sp/services/sm/risklevel/1/
}
```

You can see which enumeration values are available by accessing them over the corresponding references:

```
GET http://esb.heimdall.sp/services/sm/hazardtype/
GET http://esb.heimdall.sp/services/sm/impact/
GET http://esb.heimdall.sp/services/sm/risklevel/
GET http://esb.heimdall.sp/services/sm/scenariotype/
```

The POST response returns status code 201 and the URI of the new scenario in the "Location" header:

```
{
  "Location": "/services/sm/scenario/7/"
}
```

5.7.2 Addition of weather conditions

When creating a scenario, a user leave "conditions" sub-element empty and add it at a later point in time, through the following POST operation, as summarised in Table 5-15.

Table 5-15: Addition of weather information in a scenario.

Service ID	SP_scenario_02
Assumed consumers (via reference point)	I8
Operations	Addition of weather data
Main parameters	N/A
Message representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP STATUS CODE (201 upon successful execution)
Notes	

```
POST http://esb.heimdall.sp/services/sm/condition/
{
  "windspeed": 15,
```

```
  "temperature": 27,
  "scenario": http://esb.heimdall.sp/services/sm/scenario/7/ ,
  "winddirection": 360,
  "humidity": 25,
  "datetime": "2012-07-23T11:00:00"
}
```

The attribute "`datetime`" refers to the time of forecast. Consequently, current weather conditions should get a "`now`"-timestamp. In the following releases of the HEIMDALL system, the weather conditions will be automatically added to scenario.

5.7.3 Association of products (by reference)

The user is able to link an existing EO product to a specific scenario (by reference) and hence this information to be included in the scenario and made available to the user in a more intuitive manner. This can be performed through the following `POST` operation, as summarised in Table 5-16.

Table 5-16: Association of EO products in a scenario.

Service ID	**SP_scenario_03**
Assumed consumers (via reference point)	I8
Operations	Association of products to an existing scenario
Main parameters	N/A
Message representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP STATUS CODE (201 upon successful execution)
Notes	

To check which relationship types are allowed by the scenario the use can perform the following operation.

```
GET http://esb.heimdall.sp/services/sm/relationshiptype/
```

The following relationship types are returned:

```
{
  "meta": {
    "limit": 20,
    "next": null,
    "offset": 0,
    "previous": null,
    "total_count": 8
  },
  "objects": [
    {
```

```
      "id": 2,
      "name": "aerial-products",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/2/
    },
    {
      "id": 3,
      "name": "landslide-products",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/3/
    },
    {
      "id": 4,
      "name": "crowd-products",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/4/
    },
    {
      "id": 5,
      "name": "simulations",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/5/
    },
    {
      "id": 6,
      "name": "rva-products",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/6/
    },
    {
      "id": 7,
      "name": "isa",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/7/
    },
    {
      "id": 8,
      "name": "des",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/8/
```

```
    },
    {
      "id": 1,
      "name": "eo-products",
      "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/1/
    }
  ]
}
```

Then create relationship through the following `POST` operation.

```
POST http://esb.heimdall.sp/services/sm/relationship/
```

With request body:

```
{
  "description": "MODIS test data",
  "scenario": http://esb.heimdall.sp/services/sm/scenario/7/ ,
  "type": http://esb.heimdall.sp/services/sm/relationshiptype/1/ ,
  "uri": "<some-eo-product-uri>"
}
```

Upon successful execution, the relationship is added the scenario. In the following releases the corresponding products/relationships will be automatically added to the scenario, while still the user would have the possibility to perform manual association of products to scenarios.

5.7.4 Accessing scenario information

The user is able to access the information in a scenario by performing the following `GET` request (Table 5-17).

Table 5-17: Accessing scenario information.

Service ID	**SP_scenario_04**
Assumed consumers (via reference point)	I8
Operations	Accessing scenario information
Main parameters	N/A
Message representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	

The following provides an example of the GET request and the response provided by the scenario management module through the SP.

```
GET http://esb.heimdall.sp/services/sm/scenario/7/
RETURNS:
```

```
{
  "casualties": 0,
  "conditions": [
    {
      "humidity": 25,
      "id": 1,
      "resource_uri": http://esb.heimdall.sp/services/sm/condition/1/ ,
      "scenario": http://esb.heimdall.sp/services/sm/scenario/7/ ,
      "temperature": 27,
      "winddirection": 360,
      "windspeed": 15,
      "datetime": "2012-07-23T11:00:00"
    }
  ],
  "credibility": 10,
  "hazardlocation": {
    "coordinates": [
      2.87635353,
      42.635262626
    ],
    "type": "Point"
  },
  "hazardtime": "2012-07-22T10:20:00",
  "hazardtype": {
    "id": 1,
    "name": "forest fire",
    "resource_uri": http://esb.heimdall.sp/services/sm/hazardtype/1/
  },
  "id": 5,
  "impact": {
    "id": 1,
    "name": "1",
    "resource_uri": http://esb.heimdall.sp/services/sm/impact/1/
  },
  "injured": 0,
  "lessonslearnt": [],
  "name": "Dry Run 1 Test Scenario",
  "ongoing": true,
```

```
  "relationships": [
    {
      "description": "MODIS test data",
      "id": 1,
      "resource_uri": http://esb.heimdall.sp/services/sm/relationship/1/ ,
      "scenario": http://esb.heimdall.sp/services/sm/scenario/7/ ,
      "type": {
        "id": 1,
        "name": "eo-products",
        "resource_uri":
http://esb.heimdall.sp/services/sm/relationshiptype/1/
      },
      "uri": "<some-eo-product-uri>"
    }
  ],
  "resource_uri": http://esb.heimdall.sp/services/sm/scenario/7/ ,
  "responseplans": [],
  "risklevel": {
    "id": 1,
    "name": "very low",
    "resource_uri": http://esb.heimdall.sp/services/sm/risklevel/1/
  },
  "scenarios": [],
  "type": {
    "id": 2,
    "name": "simulated",
    "resource_uri": http://esb.heimdall.sp/services/sm/scenariotype/2/
  }
}
```


5.8 Information gateway API

The SP facilitates the exchange of information between the GUI and the IG of HEIMDALL acting as transparent proxy. The IG API is described in [7], whereas the SP provides the following API for fetching the alert areas. Table 5-18 provides a summary of the interface parameters to be used for retrieving the available areas.

Table 5-18: Retrieving the list of areas stored in the system.

Service ID	**SP_alert_01**
Assumed consumers (via reference point)	I5
Operations	Retrieving the list of areas
Main parameters	N/A
Message representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	

```
GET http://esb.heimdall.sp/services/rest/alertareas
RESPONSE:
[
    {
        "id": 250450,
        "comarca_id": 39,
        "provincia_": 25,
        "name": "Bausen",
        "precision": 3,
        "coordinates": [
            [
                0.667,
                42.826
            ],
            [
                0.736,
                42.849
            ],
            [
                0.728,
                42.829
            ],
            [
                0.667,
                42.826
            ]
        ]
```

```
    },
    {
        "id": 250637,
        "comarca_id": 39,
        "provincia_": 25,
        "name": "Canejan",
        "precision": 0,
        "coordinates": [
            [
                0.768,
                42.788
            ],
            [
                0.728,
                42.829
            ],
            [
                0.736,
                42.849
            ],
            [
                0.877,
                42.814
            ],
            [
                0.875,
                42.806
            ],
            [
                0.768,
                42.788
            ]
        ]
    }
]
```


Table 5-19 provides a summary of the service the user has to use in order to retrieve a list of the pre-defined areas stored in the system. Based on these areas the user will be able to specify the areas where his/her message will be delivered.

Table 5-19: Retrieving GeoJSON list of areas stored in the system.

Service ID	SP_alert_02
Assumed consumers (via reference point)	I5
Operations	Retrieving the list of areas
Main parameters	N/A
Message representation protocol	JSON
Communication protocol	HTTP (GET)
Response	GeoJSON
Notes	

If one wants to fetch GeoJSON through WFS the user should use:

```
GET
http://esb.heimdall.sp/services/ogc/heimdall/ows?service=WFS&version=1.1.0&request=GetFeature&typeName=heimdall:alertarea&maxFeatures=50&outputFormat=application/json

RESPONSE:

{
   "type":"FeatureCollection",
   "totalFeatures":707,
   "features":[
      {
         "type":"Feature",
         "id":"alertarea.1",
         "geometry":{
            "type":"Polygon",
            "coordinates":[
               [
                  [
                     0.667,
                     42.826
                  ],
                  [
                     0.736,
                     42.849
                  ],
                  [
                     0.728,
```

```
                  42.829
               ],
               [
                  0.667,
                  42.826
               ]
            ]
         ]
      },
      "geometry_name":"geom",
      "properties":{
         "fid":1,
         "id":250450,
         "comarca_id":39,
         "provincia_":25,
         "name":"Bausen",
         "precision":3
      }
   },
   {
      "type":"Feature",
      "id":"alertarea.2",
      "geometry":{
         "type":"Polygon",
         "coordinates":[
            [
               [
                  0.768,
                  42.788
               ],
               [
                  0.728,
                  42.829
               ],
               [
                  0.736,
                  42.849
               ],
```

```
                [
                   0.877,
                   42.814
                ],
                [
                   0.875,
                   42.806
                ],
                [
                   0.768,
                   42.788
                ]
             ]
          ]
       },
       "geometry_name":"geom",
       "properties":{
          "fid":2,
          "id":250637,
          "comarca_id":39,
          "provincia_":25,
          "name":"Canejan",
          "precision":-1
       }
    },
    {
       "type":"Feature",
       "id":"alertarea.3",
       "geometry":{
          "type":"Polygon",
          "coordinates":[
             [
                [
                   0.669,
                   42.802
                ],
                [
                   0.667,
```

```
                  42.826
               ],
               [
                  0.728,
                  42.829
               ],
               [
                  0.768,
                  42.788
               ],
               [
                  0.749,
                  42.785
               ],
               [
                  0.669,
                  42.802
               ]
            ]
         ]
      },
      "geometry_name":"geom",
      "properties":{
         "fid":3,
         "id":251214,
         "comarca_id":39,
         "provincia_":25,
         "name":"Les",
         "precision":-1
      }
   }
]
}
```

5.9 Secondary Services

5.9.1 Spatial resources Metadata Server (Geonetwork)

In order to comply with the INSPIRE, the SP hosts a geonetwork server. Geonetwork is a catalogue application to manage spatially referenced resources. It provides powerful **metadata editing** and **search** functions by providing various metadata templates. Among those templates there is also the INSPIRE template which can be used by content owners to fill in the metadata of their products.

The search functionality gives the user the ability to search with various keywords, or simply browse the products uploaded to the platform.

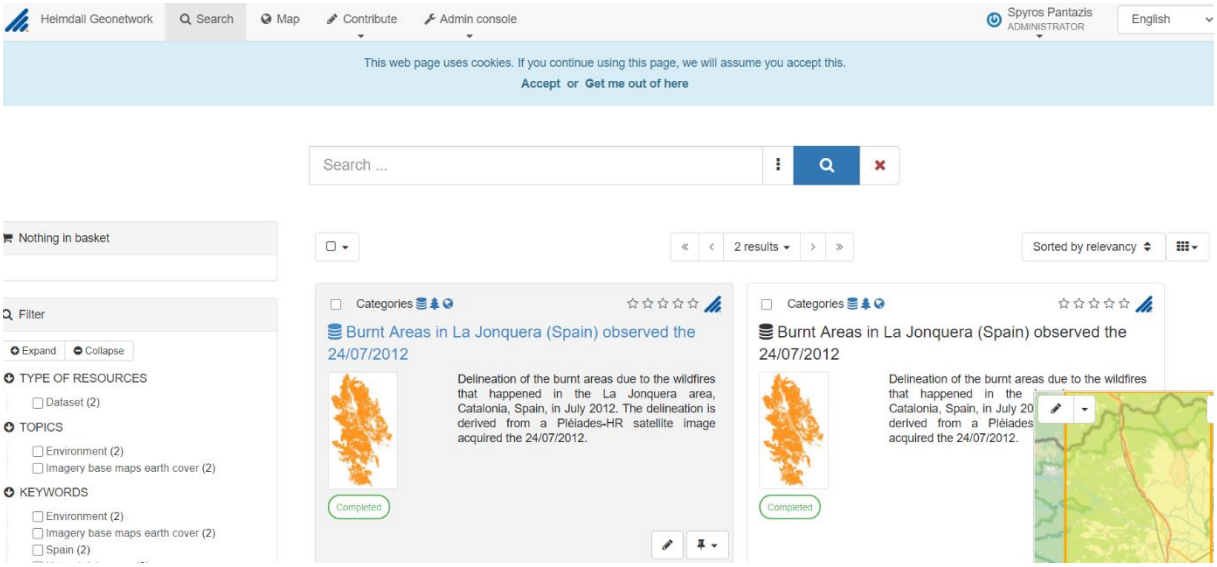

Figure 5-1: Geonetwork graphical interface

Each product presented contains a list of metadata, such information about the owner of the products and details of its contents.

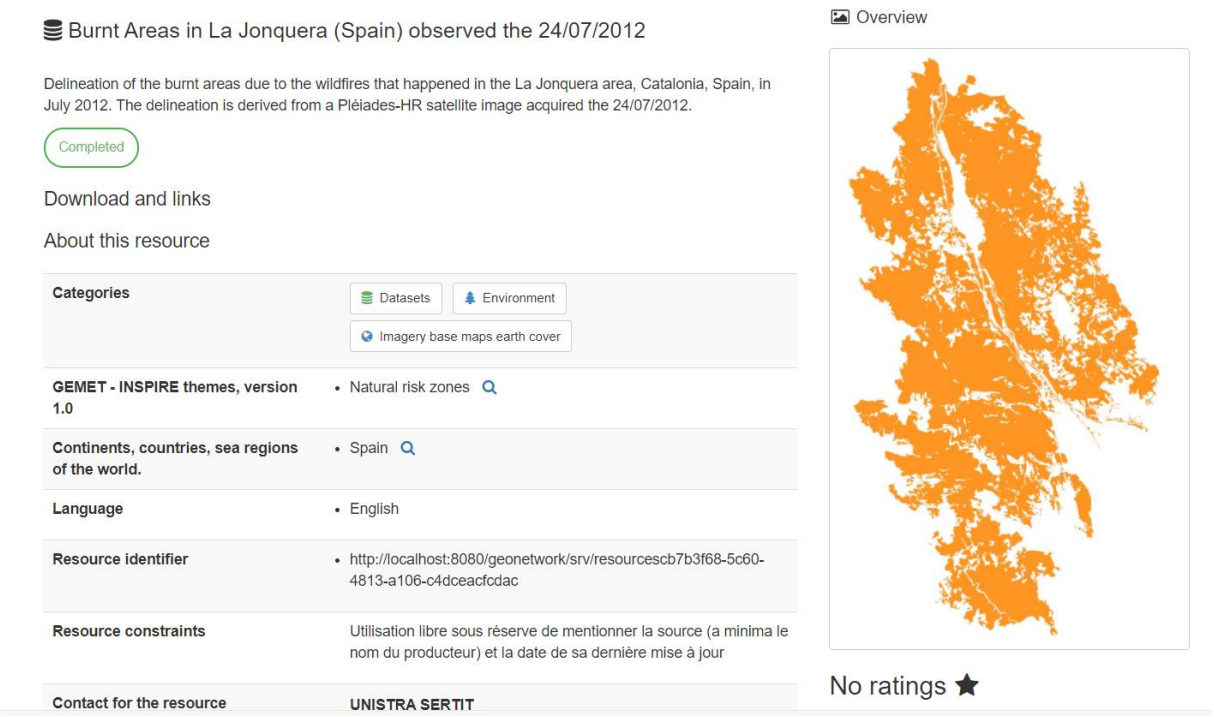

Figure 5-2: Metadata overview

The user can download the metadata or directly access and view the product if it is either public or the owner has provided corresponding access rights to that user.

Figure 5-3: Metadata download option and contacts

Apart from searching/viewing registered spatial products, a user can also manage the metadata templates, create its own templates and further use them in its own products.

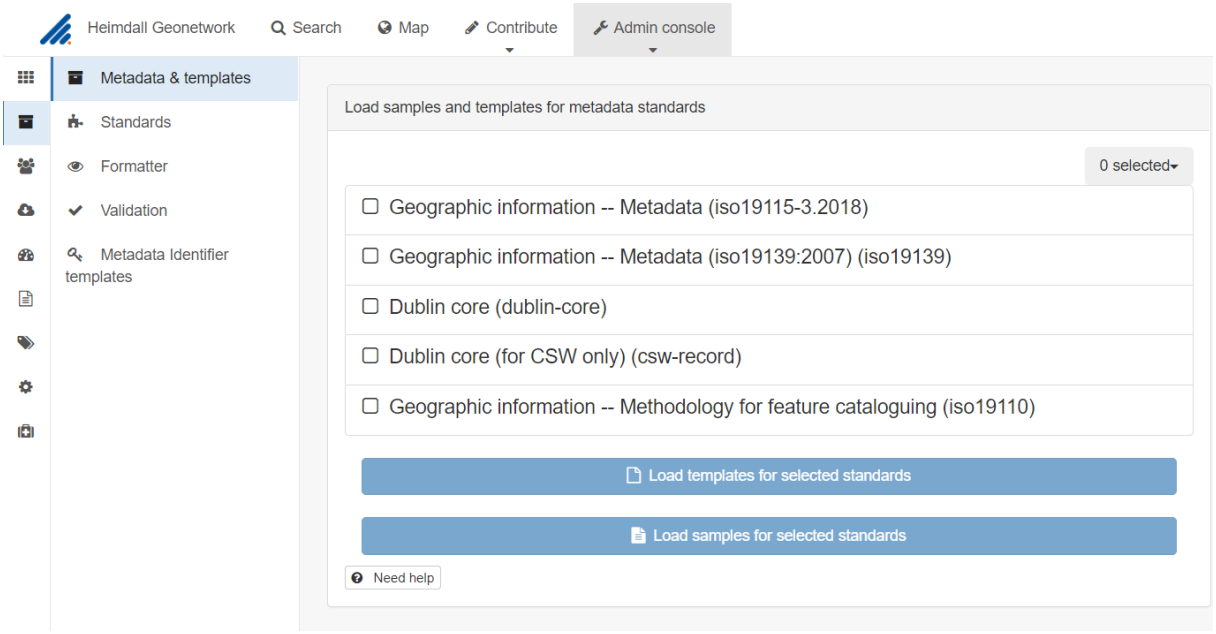

Figure 5-4: Metadata templates

Apart from the handy web UI, geonetwork server also supports a REST API for programmatically authoring metadata and spatial products.

5.9.2 **Chat Server (Openfire)**

The chat server provides the functionalities required for the users to be able to exchange chat messages. SP includes an Openfire XMPP server to cope with chat message exchange between the platform's users.
Openfire is a real time collaboration (RTC) server licensed under the Open Source Apache License. It uses the only widely adopted open protocol for instant messaging, namely XMPP.
Openfire comes with a web UI, which can be used to manage accounts, sessions and chatting functionalities in general.

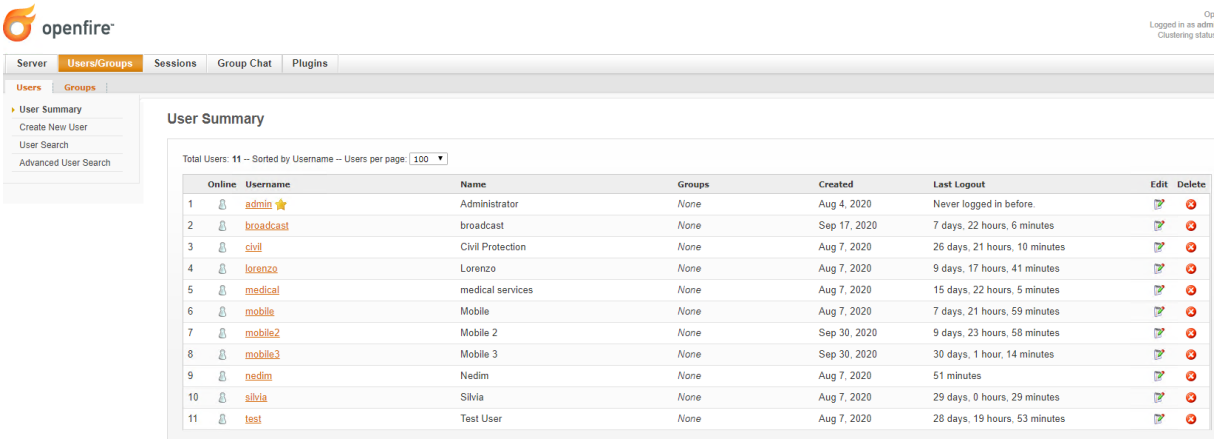

Figure 5-5: Chat server web UI

Apart from the web UI, Openfire server can be extended via 3rd party plugins. Such a plugin that the SP uses in order to communicate with Openfire server, register new user accounts and manage chat sessions, is the REST API plugin.

Heimdall's web GUI communicates with the chat server with the BOSH protocol. The mobile application on the other hand makes direct use of the XMPP protocol.

5.10 Other Services

5.10.1 Registry Service

The registry service is a directory service, which contains information about the various SP instances. That information is being used by:

i) The SP instances themselves to reach each other and exchange information
ii) The web GUI to resolve to which instance it should connect to, as also to fetch connection information related to the services that the instance provide, such as the chat server.
iii) The catalogue service to get information about the SP instances

The registry service exposes the REST API described at the example below.

```
GET http://registry.heimdall.sp/services/rest/registry
```


```
[
    {
        "LUID": "636a576f-dc8d-484b-8498-67a750047d00",
        "Fqdn": "esb2.heimdall.sp",
        "Description": "Heimdall Instance",
        "IPAddress": "192.168.127.4",
        "Country": "Italy",
        "State": "Liguria",
        "Region": "Monesi",
        "OrganisationName": "Italian Red Cross",
        "Discipline": null,
        "WSUrl": "ws://192.168.127.3:9999/",
        "XmppFqdn": "esb.heimdall.sp",
        "XmppDomain": "conference.esb.heimdall.sp",
        "XmppBoschUrl": "http://esb.heimdall.sp:7070/http-bind/"
    },
    {
        "LUID": "77200cf6-6e38-4905-9a44-229efb3e88b4",
        "Fqdn": "esb.heimdall.sp",
        "Description": "Heimdall Instance",
```

```
        "IPAddress": "192.168.127.2",
        "Country": "Spain",
        "State": "Catalonia",
        "Region": "La Jonquera",
        "OrganisationName": "Firefighters of the Generalitat of Catalonia",
        "Discipline": null,
        "WSUrl": "ws://192.168.127.3:9999/",
        "XmppFqdn": "esb.heimdall.sp",
        "XmppDomain": "conference.esb.heimdall.sp",
        "XmppBoschUrl": "http://esb.heimdall.sp:7070/http-bind/"
    }
]
```


Apart from the generic information of the LUs, registry can also provide information about the roles and disciplines of all registered LU instances.

```
GET http://registry.heimdall.sp/services/rest/lists/roles
```


```
[
    "Fire Analyst Coordinator",
    "Dispatcher Operator",
    "Control Room Chief",
    "Incident Commander",
    "Police Department Mossos d'Esquadra. Autonomic Police Force of Catalonia",
    "Police Department Local Police",
    "Alarm Reception Centre (Local)",
    "Fire Service",
    "Control room operator",
    "First responder",
    "Local Police"
]
```


```
GET http://registry.heimdall.sp/services/rest/lists/disciplines
```


```
[
```

```
    "Fire Fighters",
    "Red Cross",
    "Space Hellas"
]
```


There exists also an API endpoint for searching all users of the LUs with a specific role.

`GET http://registry.heimdall.sp/services/rest/roles?role=Incident Commander`


```
[
    {
        "LUID": "77200cf6-6e38-4905-9a44-229efb3e88b4",
        "name": "Incident Commander",
        "users": [
            {
                "username": "ic",
                "fullname": "Incident Commander"
            },
            {
                "username": "ic-a",
                "fullname": "Incident Commander"
            }
        ]
    },
    {
        "LUID": "636a576f-dc8d-484b-8498-67a750047d00",
        "name": "Incident Commander",
        "users": [
            {
                "username": "ic",
                "fullname": "Incident Commander"
            }
        ]
```

```
    },
    {
        "LUID": "77200cf6-6e38-4905-9a44-229efb3e88b4",
        "name": "Incident Commander",
        "users": [
            {
                "username": "ic",
                "fullname": "Incident Commander"
            },
            {
                "username": "ic-a",
                "fullname": "Incident Commander"
            }
        ]
    }
]
```

5.10.2 Messaging Service

The messaging service can be used by the GUI or any other components of the Heimdall platform to send a message to an LU or to a subset of its users.

An example of its API follows.

```
POST http://LU.Fqnd OR LU.IpAddress/services/rest/messagehub/send
```

```
{
        "SourceLUID": "636a576f-dc8d-484b-8498-67a750047d00",
        "DestinationLUID":          ["77200cf6-6e38-4905-9a44-229efb3e88b4",
"636a576f-dc8d-484b-8498-67a750047d00"],
        "Process" : "subscription",
        "Body":
        {
            "CUSTOM_JSON_HERE"
```

```
        }

}
```

SourceLUID : The LU that owns the information/content to be shared with destination LU
DestinationLUID: The LU to which information/content is published through the message hub
Process : (Optional) The process that initiated the message
Body : Custom Json describing the content/information to be shared. Depends on the implementation of external modules (i.e. Catalogue module)

The body of the message, which is a custom json object, can used to extend the messaging functionality of the SP. For example, in order to send an instant chat message to a user of an LU, one can use:

```
{

        "SourceLUID": "MYLUID",

        "DestinationLUID":          ["77200cf6-6e38-4905-9a44-229efb3e88b4",
"636a576f-dc8d-484b-8498-67a750047d00"],

        "Body":

        {

            "type" : "instant",

                "users" : [username1, username2]

                "message" : {"message here"}

        }

}
```

5.10.3 Map Helper Functions Service

This service provides supportive map functions, like reverse geocoding and countries/states/regions lists.
The following example shows how to get the address of one or more geo locations.
`POST http: //esb.heimdall.sp/services/rest/mapfunctions/reverse`

```
[[23.878, 38.9891], [1.788, 41.812312], [11.33989, 44.50155]]
```

Returns:

```
[

    {

        "Area": {

            "Country": "Greece",
```

```
            "State": "Unknown",
            "Region": "Unknown"
        }
    },
    {
        "Area": {
            "Country": "Spain",
            "State": "Catalonia",
            "Region": "Callús"
        }
    },
    {
        "Area": {
            "Country": "Italy",
            "State": "Emilia-Romagna",
            "Region": "Bologna"
        }
    }
]
```


The API to fetch the states of a country is as follows.

```
GET http: //esb.heimdall.sp/services/rest/mapfunctions/regions? Country = Italy
```


```
[
   "Agrigento",
   "Bari",
   "Bologna",
   "Bolzano",
. . .
]
```

5.10.4 Legends Service

The legends service facilitates the creation of custom legend information of a map layer. The service consists of a set of API endpoints.

Fetch a legend by the layer type

```
GET http://esb.heimdall.sp/services/rest/legends?type=firelineintensity
```

Result:


```
[
    {
        "label": "Low",
        "value": "#548235"
    },
    {
        "label": "Moderate",
        "value": "#92d050"
    },
    {
        "label": "High",
        "value": "#ed7d31"
    },
    {
        "label": "Very high",
        "value": "#FF0000"
    }
]
```


Type can be one of: minimumtraveltime / flamelength / firelineintensity / rateofspread / outofsuppressioncapacity

Fetch a legend by layer name

```
GET http://esb.heimdall.sp/services/rest/legends?layer=heimdall:Rius_Alt_Emporda
```

Result:


```
[
    {
```

```
        "label": "Low",
        "value": "#548235"
    },
    {
        "label": "Moderate",
        "value": "#92d050"
    }
]
```

Add a new legend to a type of layer

```
POST http://esb.heimdall.sp/services/rest/legends?type=firelineintensity
```

```
[
    {
        "label": "Low",
        "value": "#548235"
    },
    {
        "label": "Moderate",
        "value": "#92d050"
    },
    {
        "label": "High",
        "value": "#ed7d31"
    },
    {
        "label": "Very high",
        "value": "#FF0000"
    }
]
```

Add a legend to a specific layer

```
POST http://esb.heimdall.sp/services/rest/legends?layer=heimdall:Rius_Alt_Emporda
```


```
[
    {
        "label": "Low",
        "value": "#548235"
    }
]
```


5.11 Waypoints API

The waypoints service is being used by the GUI (web and mobile) for creating and sharing lists with points of interest between the users of the platform. The service comes with the following set of API endpoints.

Fetch all waypoints shared with the current user

```
GET http://esb.heimdall.sp/services/rest/waypoints
```

Returns:


```
[
    {
        "id": 1,
        "scenarioId": 137,
        "geoJson": {
            "type": "FeatureCollection",
            "totalFeatures": 1,
            "features": [
                {
                    "type": "Feature",
                    "id": "wpts_kvX-QTBjL0CFbnK0pv3yOQ.1",
                    "geometry": {
                        "type": "Point",
                        "coordinates": [
                            -0.0909003,
                            51.5282222
```

```
                        ]
                    },
                    "geometry_name": "geom",
                    "properties": {
                        "fid": 1,
                        "waypointid": 1,
                        "responses": []
                    }
                }
            ],
            "crs": {
                "type": "name",
                "properties": {
                    "name": "urn:ogc:def:crs:EPSG::4326"
                }
            },
            "bbox" : []
        },
        "users": [
            "span",
            "angel"
        ],
        "owner": "span"
    },
    {
        "id": 6,
        "geoJson": {
            "type": "FeatureCollection",
            "totalFeatures": 2,
            "features": [
```

```
                {
                    "type": "Feature",
                    "id": "wpts_Mg97iGqvmESsIlJfwUC6dw.4",
                    "geometry": {
                        "type": "Point",
                        "coordinates": [
                            -0.0909003,
                            51.5282222
                        ]
                    },
                    "geometry_name": "geom",
                    "properties": {
                        "fid": 4,
                        "waypointid": 6,
                        "responses": [
                            {
                                "span": "ok"
                            },
                            {
                                "angel": "notok"
                            }
                        ]
                    }
                },
                {
                    "type": "Feature",
                    "id": "wpts_Mg97iGqvmESsIlJfwUC6dw.5",
                    "geometry": {
                        "type": "Point",
                        "coordinates": [
```

```
                        9.35321693717188,
                        44.3483848925781
                    ]
                },
                "geometry_name": "geom",
                "properties": {
                    "fid": 5,
                    "waypointid": 0,
                    "responses": []
                }
            }
        ],
        "crs": {
            "type": "name",
            "properties": {
                "name": "urn:ogc:def:crs:EPSG::4326"
            }
        }
    },
    "users": [
        "angel",
        "crc"
    ],
    "owner": "span"
}
```


Fetch a specific waypoints list

GET http://esb.heimdall.sp/services/rest/waypoints?id=32

Returns:


```
{
```

```
    "id": 32,
    "scenarioId": null,
    "geoJson": {
        "type": "FeatureCollection",
        "features": [
            {
                "type": "Feature",
                "id": "wpts_xFutNWszOEm1v34oaim2ow.3",
                "geometry": {
                    "type": "Point",
                    "coordinates": [
                        8.90056699186212,
                        44.4601075426495
                    ]
                },
                "geometry_name": "geom",
                "properties": {
                    "fid": 3,
                    "waypointid": 0,
                    "description": "",
                    "responses": [],
                    "created": "2019-10-17T15:55",
                    "lastupdated": "2019-10-17T15:55"
                }
            },
            {
                "type": "Feature",
                "id": "wpts_xFutNWszOEm1v34oaim2ow.4",
                "geometry": {
                    "type": "Point",
```

```
                    "coordinates": [
                        -0.0908003,
                        51.5272222
                    ]
                },
                "geometry_name": "geom",
                "properties": {
                    "fid": 4,
                    "waypointid": 0,
                    "description": "test2",
                    "responses": [],
                    "created": "2019-10-18T07:07",
                    "lastupdated": "2019-10-18T07:07"
                }
            }
        ],
        "crs": {
            "type": "name",
            "properties": {
                "name": "urn:ogc:def:crs:EPSG::4326"
            }
        },
        "bbox": [
            -0.0908003,
            44.4601075426495,
            8.90056699186212,
            51.5272222
        ]
    },
    "users": [
```

```
        "lp"
    ],
    "owner": "testuser"
}
```

Create a new waypoints list

```
POST http://esb.heimdall.sp/services/rest/waypoints
```

```
{
   "scenarioId":137,
   "geoJson":{
      "type":"FeatureCollection",
      "features":[
         {
            "type":"Feature",
            "geometry":{
               "type":"Point",
               "coordinates":[
                  -0.0909003,
                  51.5282222
               ]
            },
            "properties":{
               "description" : "test"
            }
         }
      ]
   },
   "users":[
      "span",
      "angel"
   ]
```

```
}
```

scenarioid and **description** are optional

Add a point to a waypoints list

```
PUT http://esb.heimdall.sp/services/rest/waypoints
```

```
{
   "id" : 1,
   "geoJson":{
      "type":"FeatureCollection",
      "features":[
         {
            "type":"Feature",
            "geometry":{
               "type":"Point",
               "coordinates":[
                  -0.0908003,
                  51.5272222
               ]
            },
            "properties":{
               "description" : "test"
            }
         }
      ]
   }
}
```

Delete a waypoints list

```
DELETE http://esb.heimdall.sp/services/rest/waypoints?id={id}
```

Delete a point of a waypoints list

```
DELETE http://esb.heimdall.sp/services/rest/waypoints?id={id}&featureId={featureid}
```

5.12 Catalogue Service

The SP's catalogue service is a proxy service in front of Heimdall's platform catalogue service. It transforms the input from the GUI to the input required by the catalogue service by adding missing information, such as current Local Unit (LU) id.

Following are the available APIs

Naming structure.
`GET http://esb.heimdall.sp/services/catalog/tree/`
Response

```
{

"full naming structure in json format"

}
```

Publish data

This method is called if a data owner wants to share data with other entities. The name of the data shall be sent. The publications table in the database is updated.

`POST http://esb.heimdall.sp/services/catalog/pub/` with Content descriptor in JSON format in the input body;
an example:

```
{

"Access Rules":

    {

    "LU ID":

        [

        "LU ID 1",

        "LU ID 2"

        ],

    "Country":

        [

        "Country 1",

        "Country 2",

        "Country 3"

        ]

    },

    "ContentUri" : "THE Uri of the content to be published"
```

```
}
```

Response
on success: **HTTP code 200** with the PID:

{"PID" : 15 }

ContentUri : The Uri pointing to the actual content to be published. Examples:

SM : http://esb.heimdall.sp/services/sm/scenario/ID/
Fire Simulation : http://esb.heimdall.sp/services/rest/simulations?simulationId=ID
Landslide Simulation : http://esb.heimdall.sp/services/rest/landslidesimulations?id=ID
Impact Assessment : http://esb.heimdall.sp/services/rest/impactassessments?id=ID
ISAS : http://192.168.127.2/services/isas/result/?ID
EO product : namespace:layername , i.e. heimdall:Chiavari.EO.Flood.PLD.LE.2014.11.19.10.19.20

Edit data and add access rule

This method is called if a data owner wants to edit data in an existing publication with a particular PID. The name of the data shall be sent. The publications table in the database is updated. Any additional access rule is added.

`PATCH http://esb.heimdall.sp/services/catalog/pub/` with Content descriptor in JSON format in the input body; an example:

```
{
"PID":1,
"Access Rules":
    {
    "LU ID":
        [
        "LU ID 1",
        "LU ID 2"
        ],
    "Country":
        [
        "Country 1",
        "Country 2",
        "Country 3"
```

```
        ]
    }
}
```

Response
on success: **HTTP code 200**

Undo publication of data

This method is called if a data owner wants to stop sharing data with other entities. The PID shall be sent. The publications table in the database is updated. `DELETE http://esb.heimdall.sp/services/catalog/pub/PID/PIDNUMBER`
Response
on success: **HTTP code 200** and JSON with details

```
{
    "success" : "1 record deleted"
}
```

Query published data

This method can be used to check published data and the according access rights. If all data of a LU needs to be checked, only the LU_ID needs to be sent. Otherwise, the CD can be used to filter the publications. This method can only be used by the publisher. `GET http://esb.heimdall.sp/services/catalog/pub/LU ID/`
Response
on success: HTTP code 200 and JSON with results; an example:

```
[
    {
    "publications":
        [
        {
        "PID":"1",
        "Access Rules":
            {
            "LU ID":
                [
                    "SPH",
                    "L2"
                ],
```

```
                "Country":
                    [
                    "Fra11",
                    "Many",
                    "Ger"
                    ]
            },
            "LU ID":"DLR",
            "Language":"en-GB",
            "HEIMDALL Product":
                {
                "Scenario":
                    [
                    {
                    "Scenario ID":"Sce123"
                    },
                    {
                    "Hazard Type":
                        [
                        "Landslide"
                        ]
                    },
                    {
                    "Status":"Exercise"
                    },
                    {
                    "Urgency":"Immediate"
                    }
                    ]
                }
```

```
        },
        {
        "PID":"1",
        "Access Rules":
            {
            "LU ID":
                [
                "SPH",
                "L2"
                ],
            "Country":
                [
                "Fra11",
                "Many",
                "Ger"
                ]
            },
        "LU ID":"DLR",
        "Language":"en-GB",
        "HEIMDALL Product":
            {
            "EO":
                [
                {
                "Hazard Type":
                    [
                    "Forest fire"
                    ]
                }
                ]
```

```
            }
        }
        ]
    }
]
```

Subscribe to data

This method is called if a user wants to subscribe to content. `POST http://esb.heimdall.sp/services/catalog/sub/` with Content descriptor in JSON format in the input body; an example:

```
{
    "Role":"My Role",
    "Discipline":"My Discipline",
    "Country":"My Country",
    "Language":"en-GB",
    "HEIMDALL Product":"ISAS"
}
```

Response
on success: **HTTP code 200** and JSON with unique subscription ID and list of publications that are subscribed if available
List with publications where the access is granted.


```
[{"SID":"13"},{"publications":[{"PID":"17","LU ID":"9ee894d1-8588-4d72-983b-6a8b2b65c17b","Language":"en-GB","HEIMDALL Product":{"ISAS":[{"ISAS ID":"8e000041-824c-48fa-9ee4-bf27c1b594e5"},{"Date":"20190723"},{"Timestamp":"203111"},{"Status":"done"},{"Summary Products":["Total affected physical numbers","Total affected physical percentage","Max hazard category","Max physical damage in percentage","Total economic loss","Total affected nighttime population","Total affected daytime population"]},{"Area":{"Country":"Italy","State":"Liguria","Region":"Mendatica"}},{"List of damaged assets":{"Buildings":["Asset function","Day time population","Night time population","Physical damage in percentage","Location"]}}]}}]}]
```


on error: **HTTP code 400** and JSON with error details

```
{
```

```
    "error" : "error message as string"

}
```

or

```
{"error":"1. subscriptions with same inputs already exist. A duplication is not allowed.;"}
```

or
Successful in subscribing to a topic (indicated by returned field "SID") but no matching publication

```
{"Partial success":[{"SID":"2"}],"error":"1. No matching publications record exist;"}
```

Unsubscribe to data

This method is called if a user wants to unsubscribe content. No updates will be received anymore.
`DELETE http://esb.heimdall.sp/services/catalog/sub/SID/<SID-number>`
Response
on success: **HTTP code 200** and JSON with details

```
{

    "success" : "1 record deleted"

}
```

on error: **HTTP code 400** and JSON with error details

```
{

    "error" : "error message as string"

}
```

Check subscriptions by LU
This method can be used to check subscribed topics. If all data of a LU needs to be checked, only the LU_ID needs to be sent. Otherwise, the CD can be used to filter the subscriptions. This method can only be used by the subscriber.

HTTP GET http://esb.heimdall.sp/services/catalog/sub/LU ID/636a576f-dc8d-484b-8498-67a750047d00

```
[{"subscriptions":[{"SID":"1","LU ID":"636a576f-dc8d-484b-8498-67a750047d00"}]}]
```

on error: **HTTP code 400** and JSON with error details

```
{

    "error" : "error message as string"
```

```
}
```

Query

This method is used to query the network for data. It returns all the data the interested entity has access rights and that fit the query. `POST http://esb.heimdall.sp/services/catalog/query/` with Content descriptor in JSON format in the input body; an example:

```
{

    "Role":"My Role",

    "Discipline":"My Discipline",

    "Country":"My Country",

    "Language":"en-GB",

    "HEIMDALL Product":"ISAS"

}
```

Response
on success: **HTTP code 200** and JSON with list of publications where the access is granted.

```
[{"publications":[{"PID":"17","LU ID":"9ee894d1-8588-4d72-983b-
6a8b2b65c17b","Language":"en-GB","HEIMDALL Product":{"ISAS":[{"ISAS
ID":"8e000041-824c-48fa-9ee4-
bf27c1b594e5"},{"Date":"20190723"},{"Timestamp":"203111"},{"Status":"done"},
{"Summary Products":["Total affected physical numbers","Total affected
physical percentage","Max hazard category","Max physical damage in
percentage","Total economic loss","Total affected nighttime population","Total
affected daytime
population"]},{"Area":{"Country":"Italy","State":"Liguria","Region":"Mendati
ca"}},{"List of damaged assets":{"Buildings":["Asset function","Day time
population","Night time population","Physical damage in
percentage","Location"]}}]}}]}]
```

on error: HTTP code 400 and JSON with error details

```
{"error":"1. No matching publications record exist;"}
```

Create a new Workgroup

POST http://esb.heimdall.sp/services/catalog/wg/

Response
on success: **HTTP code 200** and id of the new workgroup:

```
{

  "WID" : "1"

}
```

Populate a workgroup with members

PATCH **http://esb.heimdall.sp/services/catalog/wg/**

```
{
   "WID" : "4",
   "members" :
   [

      { "LUID" : "77200cf6-6e38-4905-9a44-229efb3e88b4",
        "usernames" : [ "span", "angel" ]
      },
      { "LUID" : "636a576f-dc8d-484b-8498-67a750047d00",
        "usernames" : [ "ic" ]
      }
   ]
}
```

Broadcast message to a Workgroup

PATCH **http://esb.heimdall.sp/services/catalog/wg/**

```
{
    "WID":"4",
    "message":
     {
        "Scenario":"http://esb.heimdall.sp/services/sm/scenario/137/"
     }
}
```

Delete workgroup

```
DELETE http://esb.heimdall.sp/services/catalog/wg?id=
```

6 Test Plan and Report

This section contains the list of tests designed in order to verify the coverage of the relevant requirements described in Section 2. It is important to highlight that the tests documented in this deliverable are the ones for testing the functionalities of SP system modules individually and that the integration tests will be provided in the context of WP 2.

The tests are defined during the implementation of the various features and refined as the implementation matures. Then, two months before each release, the tests are performed, in collaboration with the HEIMDALL partners, the results are documented and updates are performed for each unsuccessful result.

For each technical requirement, suitable tests have been described and performed for assessing the fulfilment of each technical requirement. The template used for the documentation of the tests can be found in Table 6-1.

Table 6-1: Test template

Test ID	*Unique test identifier in the format "TS_SP_#"*
Requirements to be verified	*List of technical and system requirements that this test verifies in the form* • *TR_SP_#* ○ *Sys_<module>_#*
Test objective	*Short description of the test objective*
Test procedure	*Detailed steps to be followed in order to perform the test in the form* 1. *The user …* 2. *The user…* 3. *…*
Test prerequisites/ configuration	*List of pre-requisites which are mandatory to be fulfilled before the test starts; in the form* • *…*
Success criteria	*List or description of success criteria*
Results analysis	*Analysis of the test*
Success	PASSED / FAILED

6.1 Test Report

This section presents the testing campaign of the system, against solidly defined test cases. Each test case aims to validate one or more functional technical requirements of the system defined in Section 2.

Table 6-2: TS_SP_01: Access the database that stores GIS data through the GUI.

Test ID	*TS_SP_01*

Requirement to be verified	• *TR_SP_01*
	  ○ *Sys_IntData_1*
	  ○ *Sys_IntData_3*
	• *TR_SP_03*
	  ○ *Sys_IntData_1*
	  ○ *Sys_IntData_3*

Test objective	Verify that the SP provides a database to store GIS data. The operation can be performed by directly using the REST API.

Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user starts the web portal and logs in. 3. The user opens the main page and visualise the map with the different layers. 4. The user goes to the simulation window that displays the results of the simulation. 5. The user clicks on one of the sensor icon that opens a window to display the sensor information.
Test prerequisites/ configuration	• The web portal needs to be up and running. • The service platform can serve maps, at least one simulation has been started and information about at least one sensor has been entered in the service platform.
Success criteria	The user can visualise the following elements on the SP GeoServer: • Maps and layers • Simulation results • Sensor information
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-3: TS_SP_02: Access the database that stores GIS data and EO products through FTP.

Test ID	*TS_SP_02*

Requirement to be verified	• *TR_SP_01*
	  ○ *Sys_IntData_1*
	  ○ *Sys_IntData_3*
	• *TR_SP_02*
	  ○ *Sys_IntData_1*
	  ○ *Sys_IntData_2*
	• *TR_SP_03*
	  ○ *Sys_IntData_1*
	  ○ *Sys_IntData_3*

Test objective	Verify that the SP provides a database to store GIS data and EO products. The operation can be performed by using an FTP service.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user connects to the HEIMDALL FTP server. 3. The user uploads the files that have to follow the EPSG 4326 (WGS84 decimal degrees) coordinates reference system and the naming conventions, in order to be automatically registered in the SP GIS database.
Test prerequisites/ configuration	• The FTP server needs to be up and running. • The web portal needs to be up and running. • The service platform can serve maps.
Success criteria	The user can visualise the following elements on the UI: • Maps and layers • Simulation results

	• Sensors information
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-4: TS_SP_03: Storing and retrieving EO data/products.

Test ID	*TS_SP_03*

Requirement to be verified	• *TR_SP_02*
	  ○ *Sys_IntData_1*
	  ○ *Sys_IntData_2*

Test objective	Verify that the SP is able to store and retrieve EO data/products and make them available to the user through the GUI.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user connects to the HEIMDALL FTP server. 3. The users uploads EO products (layers and images)
Test prerequisites/ configuration	• The FTP server needs to be up and running. • The web portal needs to be up and running. • The service platform can serve maps.
Success criteria	The user can visualise the following elements on the UI: • Maps and layers
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-5: TS_SP_04: Receiving the position of a first responder (georeferenced information).

Test ID	*TS_SP_04*
Requirement to be verified	• *TR_SP_04* ○ *Sys_IntData_3*
Test objective	Validation of the capability of the SP to store and communicate geo-referenced data.
Test procedure	A user through the HEIMDALL mobile phone application shares his/her location to the HEIMDALL SP, by pressing the corresponding button. A first responder position (as an example of georeferenced data) is stored at the SP via an HTTP REST API call. Then this location can be displayed in the GUI.
Test prerequisites/ configuration	• The mobile phone should have access to the HEIMDALL VPN. • The web portal needs to be up and running. • The service platform can serve maps.
Success criteria	The user can visualise the following elements on the UI: • Maps and layers • User location (points)
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Test TS_SP_04 validates only the reception of the location from the responder through mobile app. More tests have been designed and performed during the development of the HEIMDALL mobile application.

Table 6-6: TS_SP_05: Access to historical data.

Test ID	*TS_SP_05*

Requirement to be verified	• *TR_SP_12*
	  ○ *Sys_IntUeMan_5*
	  ○ *Sys_IntUeMan_6*
	• *TR_SP_15*
	  ○ *Sys_IntUeMan_5*
	  ○ *Sys_IntUeMan_6*

Test objective	Verify access to historical data.
Test procedure	A request is sent to an SP service (e.g. the weather service) requiring data at a specific time in the past.
Test prerequisites/ configuration	The data needs to be already present in the SP.
Success criteria	The SP should respond with the requested historical data.

Results analysis	Request:
	http://esb.heimdall.sp/services/rest/weather/conditions/q/42.35,1.46.json?time=20180515:19&externaljson=true
	
	SP Response:
	{
	"type":"FeatureCollection",
	"totalFeatures":1,
	"features":[
	{
	"type":"Feature",
	"id":"conditions.180789",
	"geometry":{
	"type":"Point",
	"coordinates":[
	0.36,
	40.6
	]
	},
	"geometry_name":"geom",
	"properties":{
	"fid":180789,
	"temperature":19.4,
	"humidity":"53%",
	"winddirection":74,
	"windspeed":"6",

```
            "pressure":"30.04",
            "percipitation1hr":" 0",
            "percipitationtoday":"0",
            "date":"2018-05-16T08:49:16Z",
            "sky":null,
            "json":"\n{\n            \"response\":      {\n
\"version\":\"0.1\",\n
\"termsofService\":\"http://www.wunderground.com/weather/ap
i/d/terms.html\",\n  \"features\": {\n  \"conditions\": 1\n
}\n\t}\n     ,\t\"current_observation\":  {\n\t\t\"image\":
{\n\t\t\"url\":\"http://icons.wxug.com/graphics/wu2/logo_13
0x80.png\",\n\t\t\"title\":\"Weather
Underground\",\n\t\t\"link\":\"http://www.wunderground.com\
"\n\t\t},\n\t\t\"display_location\": {\n\t\t\"full\":\"Sant
Rafel del Maestrat, Spain\",\n\t\t\"city\":\"Sant Rafel del
Maestrat\",\n\t\t\"state\":\"CS\",\n\t\t\"state_name\":\"Sp
ain\",\n\t\t\"country\":\"SP\",\n\t\t\"country_iso3166\":\"
ES\",\n\t\t\"zip\":\"00000\",\n\t\t\"magic\":\"24\",\n\t\t\
"wmo\":\"08238\",\n\t\t\"latitude\":\"40.600000\",\n\t\t\"l
ongitude\":\"0.360000\",\n\t\t\"elevation\":\"242.0\"\n\t\t
},\n\t\t\"observation_location\":   {\n\t\t\"full\":\"Reus,
\",\n\t\t\"city\":\"Reus\",\n\t\t\"state\":\"\",\n\t\t\"cou
ntry\":\"SP\",\n\t\t\"country_iso3166\":\"ES\",\n\t\t\"lati
tude\":\"40.60\",\n\t\t\"longitude\":\"0.37\",\n\t\t\"eleva
tion\":\"711                ft\"\n\t\t},\n\t\t\"estimated\":
{\n\t\t},\n\t\t\"station_id\":\"ISANTRAF2\",\n\t\t\"observa
tion_time\":\"Last   Updated   on   May   16,   10:42   AM
CEST\",\n\t\t\"observation_time_rfc822\":\"Wed, 16 May 2018
10:42:44
+0200\",\n\t\t\"observation_epoch\":\"1526460164\",\n\t\t\"
local_time_rfc822\":\"Wed,    16    May    2018    10:52:15
+0200\",\n\t\t\"local_epoch\":\"1526460735\",\n\t\t\"local_
tz_short\":\"CEST\",\n\t\t\"local_tz_long\":\"Europe/Madrid
\",\n\t\t\"local_tz_offset\":\"+0200\",\n\t\t\"weather\":\"
Clear\",\n\t\t\"temperature_string\":\"66.9     F     (19.4
C)\",\n\t\t\"temp_f\":66.9,\n\t\t\"temp_c\":19.4,\n\t\t\"re
lative_humidity\":\"53%\",\n\t\t\"wind_string\":\"From  the
ENE      at      3.7      MPH      Gusting      to      6.8
MPH\",\n\t\t\"wind_dir\":\"ENE\",\n\t\t\"wind_degrees\":74,
\n\t\t\"wind_mph\":3.7,\n\t\t\"wind_gust_mph\":\"6.8\",\n\t
\t\"wind_kph\":6.0,\n\t\t\"wind_gust_kph\":\"10.9\",\n\t\t\
"pressure_mb\":\"1017\",\n\t\t\"pressure_in\":\"30.04\",\n\
t\t\"pressure_trend\":\"0\",\n\t\t\"dewpoint_string\":\"50
F                                                    (10
C)\",\n\t\t\"dewpoint_f\":50,\n\t\t\"dewpoint_c\":10,\n\t\t
\"heat_index_string\":\"NA\",\n\t\t\"heat_index_f\":\"NA\",
\n\t\t\"heat_index_c\":\"NA\",\n\t\t\"windchill_string\":\"
NA\",\n\t\t\"windchill_f\":\"NA\",\n\t\t\"windchill_c\":\"N
A\",\n\t\t\"feelslike_string\":\"66.9        F        (19.4
C)\",\n\t\t\"feelslike_f\":\"66.9\",\n\t\t\"feelslike_c\":\
"19.4\",\n\t\t\"visibility_mi\":\"N/A\",\n\t\t\"visibility_
km\":\"N/A\",\n\t\t\"solarradiation\":\"--
\",\n\t\t\"UV\":\"-1\",\"precip_1hr_string\":\"-999.00 in (
```

	0 mm)\",\n\t\t\"precip_1hr_in\":\"-999.00\",\n\t\t\"precip_1hr_metric\":\" 0\",\n\t\t\"precip_today_string\":\"0.00 in (0 mm)\",\n\t\t\"precip_today_in\":\"0.00\",\n\t\t\"precip_today_metric\":\"0\",\n\t\t\"icon\":\"clear\",\n\t\t\"icon_url\":\"http://icons.wxug.com/i/c/k/clear.gif\",\n\t\t\"forecast_url\":\"http://www.wunderground.com/global/stations/08238.html\",\n\t\t\"history_url\":\"http://www.wunderground.com/weatherstation/WXDailyHistory.asp?ID=ISANTRAF2\",\n\t\t\"ob_url\":\"http://www.wunderground.com/cgi-bin/findweather/getForecast?query=40.595425,0.369418\",\n\t\t\"nowcast\":\"\"\n\t}\n}\n",
	"weather":"Clear",
	"temperature_string":"66.9 F (19.4 C)",
	"wind_string":"From the ENE at 3.7 MPH Gusting to 6.8 MPH",
	"display_full_location":"Sant Rafel del Maestrat, Spain",
	"observation_time":"Last Updated on May 16, 10:42 AM CEST",
	"weather_icon":"http://icons.wxug.com/i/c/k/clear.gif"
	}
	}
	],
	"crs":{
	"type":"name",
	"properties":{
	"name":"urn:ogc:def:crs:EPSG::4326"
	}
	}
	}

Success	PASSED

Table 6-7: TS_SP_06: The SP running on virtualised infrastructure.

Test ID	*TS_SP_06*
Requirement to be verified	• *TR_SP_09* ○ *Sys_Gen_17*
Test objective	Validation of the capability of the SP run on virtualised infrastructure
Test procedure	The SP administrator logs in the HEMIDALL server infrastructure and inspects the containers running. He/she is able to stop/start them.
Test prerequisites/ configuration	• These operations can be performed only through the HEIMDALL VPN.
Success criteria	The administrator is able to modify the status of the containers.

Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-8: TS_SP_07: Providing chat functionality.

Test ID	*TS_SP_07*

Requirement to be verified	• *TR_SP_13*
	  ○ *Sys_Int_3*
	• *TR_SP_14*
	  ○ *Sys_Int_3*

Test objective	To verify that the first responder and an operator of the GUI are able to exchange written messages through the HEIMDALL platform

Test procedure	1. The operator from the top left menu of the GUI selects a first responder user from the list
	  a. then clicks on the chat button
	  b. composes a message and presses the send button
	2. The first responder receives the message in his/her mobile application, writes and response and hits the send button.

Test prerequisites/ configuration	• The web portal needs to be up and running. • The operator has successfully logged in the GUI • The first responder is logged in the mobile application • These operations can be performed only through the HEIMDALL VPN.
Success criteria	Exchange of messages is observed among the GUI operator and the mobile application.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

6.2 Test Summary

The matrix in Table 6-9 summarizes the test coverage of technical requirements.

Table 6-9: Test coverage matrix

Requirement ID	Test ID	Result
TR_SP_01	TS_SP_01 TS_SP_02	PASSED PASSED
TR_SP_02	TS_SP_02	PASSED
TR_SP_03	TS_SP_01 TS_SP_02	PASSED PASSED
TR_SP_04	TS_SP_04	PASSED
TR_SP_05	N/A	
TR_SP_06	N/A	
TR_SP_07	N/A	
TR_SP_08	N/A	
TR_SP_09	TS_SP_06	PASSED
TR_SP_10	N/A	

TR_SP_11	N/A	
TR_SP_12	TS_SP_05	PASSED
TR_SP_13	TS_SP_07	PASSED
TR_SP_14	TS_SP_03 TS_SP_05 TS_SP_07	PASSED PASSED PASSED
TR_SP_15	N/A	
TR_SP_16	N/A	
TR_SP_17	N/A	

7 Conclusion

This deliverable presented the implementation status of the final release of the core Service Platform of HEIMDALL. The implemented SP has followed the user and system requirements.

The HEIMDALL SP has been extensively tested in lab trials and end-user workshops. During these phases, it showed adequate stability and scalability. The SP is currently operational and available for the final demo. Minor issues, if any, will be addressed over the next months.

8 References

[1] Mulero Chaves, J. et al. (2018). HEIMDALL D2.12: HEIMDALL System Architecture

[2] Bartzas, A. et al. (2018) HEIMDALL D4.1: Service Platform Design and Specification - Draft

[3] Pantazis, S. et al. (2020) HEIMDALL D4.5: Users and Roles Management Specifications – Final

[4] Mathew, D. et al. (2018) HEIMDALL D4.7: User Interface Design –Draft

[5] Mathew, D. et al. (2018) HEIMDALL D4.9: User interfaces – Draft

[6] Barth, B. et al. (2020) HEIMDALL D4.14: Communications and Information Sharing – Final

[7] Mathew, D. et al. (2020) HEIMDALL D4.17: Communications to Remote Areas – Design and Specifications – Final

[8] Friedemann, M. et al. (2020) HEIMDALL D5.2: EO Tools and Products – Specifications – Draft

[9] Barth, B. et al. (2020) HEIMDALL D5.5: In-Situ Sensors – Specifications – Draft

[10] To be released on M38 (2020) HEIMDALL D5.7: First Responders Data Module Design

[11] To be released on M38 (2020) HEIMDALL D5.8: Smartphone/Tablet Device Application for First Responders

[12] Pantazis, S., et al, (2020) HEIMDALL D5.10: Interfaces for External and Existing Systems – Specifications – Final

[13] Mendes, M. et al, (2020) HEIMDALL D5.13: Modelling and Simulation Services – Specifications – Final

[14] Friedemann, M. et al. (2020) HEIMDALL D6.3: Validated Risk Analysis and Emergency Response Methods which have been Coordinated with Product Development – Final

[15] Mendes, M. et al. (2020) HEIMDALL D6.5: Concept on Hazard, Scale and User-Specific Risk Assessment Information, Products and Service Workflows - Final

[16] Friedemann, M. et al. (2020) HEIMDALL D6.8: Situation Assessment, Impact Summary Generation and sCOP/SITREP Specification and Implementation Report – Final

[17] Friedemann, M. et al. (2020) HEIMDALL D6.11: Decision Support Specification and Implementation Report - Final

[18] Friedemann, M. et al. (2020) HEIMDALL D6.15: Scenario Specification, Scenario Management Specification and Scenario and Situation Metrics – Final

[19] PostgreSQL: The world's most advanced open-source database, available at: http://www.postgresql.org/.

[20] PostGIS, Spatial and Geographic objects for PostgreSQL, available at: http://www.postgis.net.

[21] GeoServer, an open-source geospatial server, available at: http://www.geoserver.org.

[22] Apache ActiveMQ message broker, available at: http://activemq.apache.org/.

[23] OAuth, an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications, available at: http://oauth.net.

[24] Rancher, A simplified Linux distribution built from containers, for containers, available at: https://rancher.com/rancher-os/