D4.4

Users and Roles Management Specifications - Draft

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	M20
Actual Delivery Date	21/12/2018
Editor	Alexandros Bartzas (SPH)
Contributors	Spyros Pantazis, George Vamvakas, Alexandros Bartzas (SPH)

Document History			
Version	Date	Modifications	Source
0.1	02/03/18	First draft	SPH
0.2	12/03/18	Draft technical specifications	SPH
0.3	12/04/18	Updated technical specifications	SPH
0.4	21/05/18	Draft APIs	SPH
0.5	18/07/18	Updated APIs (JWT implementation)	SPH
0.6	04/10/18	Release A updated document	SPH
0.7	03/12/18	Updated API documentation, technical specifications and testing and validation report	SPH
0.8	18/12/18	QA ready version	SPH
0.9	21/12/18	QA-reviewed version	DLR
1.0.D	21/12/18	Final draft version	SPH
1.0.F	21/12/18	Final version for submission	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
GIS	Geographic Information System
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
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
SP	Service Platform
SPH	SPACE Hellas S.A.
TOC	Table of Contents
UeRM	User and Role Management
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 first version of the technical requirements, architecture and functionality of the User and Role Management (UeRM) 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 UeRM.

The main task contributing to this deliverable is T4.2 – User and Role Management. However, contributions regarding the interfaces were made by the other technical tasks of WP4, WP5 and WP6 where the other technical components of HEIMDALL are being developed. Furthermore, T2.4 – Service Concept Specification and System Architecture defined the scope of the UeRM in the overall HEIMDALL system. The UeRM is deployed as a Virtual Machine (VM) 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 for Releases A and B, has been planned and partially executed, following the timing of the corresponding releases. The development, integration and testing activities will continue following the schedule of the upcoming releases contributing to the evolution of the HEIMDALL system towards its final release and demonstration.

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 UeRM as the component that will facilitate authentication and access control, as well as role management. This document describes WP4/T4.2 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 UeRM has to satisfy and provide.

This document focuses on providing an initial component design with a basic technical specification. Deliverables D4.5 and D4.6 due in M38 will present the final design and specifications of the UeRM and its interfaces, as well as release the software prototype.

In particular, this document is organised as follows:

- Section 2defines the technical requirements for the UeRM.
- Section 3 describes the UeRM in the context of the overall HEIMDALL system, inputs and outputs and interfaces with the HEIMDALL SP.
- Section 4 focusses on the UeRM functionalities.
- Section 5 presents the technical specification.
- Section 6 presents the internal technical testing scenarios and their results.
- Finally, Section 7 summarizes the work carried out so far and gives an outlook on the work to be performed for the completion of the UeRM design and implementation and the release of its software prototype.

2 Technical Requirements

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

2.1 Interface Requirements

2.1.1 **Hardware Interfaces**

The UeRM 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 UeRM 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), as described in D4.1 [3]. More specifically, UeRM is deployed in a VM with 4 Cores, 8 GB RAM and 256 GB HDD. OS is Windows 2012 Server.

These requirements relate to Sys_IntData_4, Sys_IntUeMan_*

2.1.3 **Communication Interfaces**

The UeRM is part of the SP and shall use either HTTP or HTTPS for secured connection, to connect to the HEIMDALL network and the internet.

These requirements relate to Sys_Int_3 and Sys_Int_4.

2.2 Functional Technical Requirements

2.2.1 **Short Term Requirements**

Table 2-1: Technical Requirement TR_UeRM_01

Requirement ID:	TR_UeRM_01
Related SR(s):	• Sys_IntData_4 • Sys_Gui_8 • Sys_Gui_10 • Sys_Gui_20 • Sys_Gui_116 • Sys_IntUeMan_12

	• Sys_IntUeMan_18
Description: The UeRM shall store the preferences of the users in their private user profile. The stored preferences shall be: • Language of the UI • Symbology • A list of the default areas of interest • Default active role • A list of the roles available to the user • Notifications	
Rational: The user preferences is an integral part of the platform, easing the usage of HEIMDALL and its wider adoption	
Stimulus: 1. Request to store (create/modify) user preferences/settings 2. Request to retrieve user preferences	
Response: 1. The UeRM receives the preferences from the GUI (via the SP) and stores them in the user preferences DB. 2. The UeRM retrieves the user preferences from the DB and forwards them to the SP, which in turn forwards them to the requesting HEIMDALL component.	
Verification Criterion: Perform multiple read and write operations in the user preferences DB and check that the data is correctly read/written.	
Notes: none	

Table 2-2: Technical Requirement TR_UeRM_02

Requirement ID:	TR_UeRM_02
Related SR(s):	• Sys_IntUeMan_1 • Sys_IntData_4
Description: The UeRM shall allow the system administrator to manage the configuration of the UeRM, concerning; • The roles/groups • The role/group permissions • Password policies • The list of preferences	
Rational: The system administrator should be able to configure the UeRM in order to match the user requirements.	
Stimulus: The administrator modifies the corresponding settings of the UeRM.	
Response: The settings are stored in the HEIMDALL platform.	

Verification Criterion: Multiple modifications of the UeRM settings are performed and validated through read operations and testing.
Notes: none

Table 2-3: Technical Requirement TR_UeRM_03

Requirement ID:	TR_UeRM_03
Related SR(s):	• Sys_IntUeMan_1 • Sys_IntUeMan_2 • Sys_IntUeMan_3 • Sys_IntData_4
Description: The UeRM shall allow the system administrator to enable and disable features, concerning: • Access rights of roles/groups to HEIMDALL products and services • Default user preferences	
Rational: The system administrator should be able to modify the corresponding features.	
Stimulus: The administrator modifies the corresponding features of the UeRM.	
Response: The settings are stored in the HEIMDALL platform.	
Verification Criterion: Multiple modifications of the UeRM features are performed and validated through read operations and testing.	
Notes: none	

Table 2-4: Technical Requirement TR_UeRM_04

Requirement ID:	TR_UeRM_04
Related SR(s):	• Sys_IntUeMan_1 • Sys_IntUeMan_2 • Sys_IntUeMan_3 • Sys_IntData_4
Description: The UeRM shall allow the system administrator to manage roles and permissions assigned to roles (create, delete and modify roles).	
Rational: The system administrator should be able to manage the roles/groups and assignment of users to roles.	
Stimulus: 1. The administrator modifies the role access rights 2. The administrator modifies the assignment of users to roles/groups	
Response: Upon successful operation, the modified roles are stored in the UeRM system.	
Verification Criterion: Multiple modifications of the roles are performed and validated through	

read operations and testing.
Notes: none

Table 2-5: Technical Requirement TR_UeRM_05

Requirement ID:	TR_UeRM_05
Related SR(s):	• Sys_IntUeMan_1 • Sys_IntUeMan_3 • Sys_IntUeMan_4 • Sys_IntData_4
Description: The UeRM shall allow the system administrator to manage users by creating, deleting and modifying (activate, deactivate and assigning roles to) users.	
Rational: The administrator should have full flexibility in managing the users and their roles.	
Stimulus: The administrator send the corresponding requests to the UeRM components	
Response: The user accounts are created/deleted/modified based on the administrator operation.	
Verification Criterion: Multiple operations on user accounts are performed and validated through read operations and testing.	
Notes: none	

Table 2-6: Technical Requirement TR_UeRM_06

Requirement ID:	TR_UeRM_06
Related SR(s):	• Sys_IntUeMan_1 • Sys_IntUeMan_2 • Sys_IntUeMan_3 • Sys_IntUeMan_4 • Sys_IntData_4
Description: The UeRM shall provide an admin console where administrators shall be able to: • Centrally manage the configuration of the UeRM • Enable and disable features • Manage roles and permissions assigned to roles (create, delete and modify roles) • Manage users (create, delete and modify (activate, deactivate and assign roles to) users)	
Rational: The administrator should be able to modify the UeRM (configuration, users, roles, etc.) though the UeRM API	
Stimulus: API calls to the UeRM	
Response: The requested operations are performed	

Verification Criterion: Multiple calls of the UeRM API
Notes: none

Table 2-7: Technical Requirement TR_UeRM_07

Requirement ID:	TR_UeRM_07
Related SR(s):	• Sys_IntUeMan_4 • Sys_IntUeMan_9 • Sys_IntUeMan_15 • Sys_IntData_4
Description: The UeRM shall provide an account management console to the users, where they shall be able to manage their own accounts. The users shall be able to (indicative): • Change their own passwords • Manage sessions • View history of the account • Modify their profile (user preferences).	
Rational: The users should be able to modify their profiles though the UeRM API	
Stimulus: API calls to the UeRM	
Response: The requested operations are performed	
Verification Criterion: Multiple calls of the UeRM API	
Notes: none	

Table 2-8: Technical Requirement TR_UeRM_8

Requirement ID:	TR_UeRM_8
Related SR(s):	• Sys_IntUeMan_5 • Sys_IntUeMan_6 • Sys_IntUeMan_7
Description: The UeRM shall allow the user to grant access to other users for the specific data he/she has permission to do so.	
Rational: The HEIMDALL platform shall enable information sharing with other users of the platform.	
Stimulus: Request to modify the access permissions of selected products/data	
Response: The UeRM forwards this to the SP that hosts the data.	
Verification Criterion: Multiple operations are performed and validated through read operations and testing.	
Notes: none	

Table 2-9: Technical Requirement TR_UeRM_9

Requirement ID:	TR_UeRM_9
Related SR(s):	• Sys_IntUeMan_1
Description: The UeRM shall support standard protocols, namely: • JWT (JSON-based open standard (RFC 7519)	
Rational: The utilisation of standards increases the maturity of the platform and makes its adoption easier from the users.	
Stimulus: A login operation triggers the generation of the JWT token	
Response: The generation of a valid JWT token	
Verification Criterion: The generation of a valid JWT token	
Notes: none	

Table 2-10: Technical Requirement TR_UeRM_10

Requirement ID:	TR_UeRM_10
Related SR(s):	• Sys_IntData_4
Description: The UeRM shall store the roles, the users, their roles and profiles.	
Rational: All operations that the administrators and users perform on the system profiles and roles should be stored in the platform.	
Stimulus: Any operation from a platform user requesting the modification of a parameter of their profiles and/or groups.	
Response: The modified parameter is stored in the UeRM database.	
Verification Criterion: Multiple operations are performed and validated through read operations and testing.	
Notes: none	

2.2.2 **Mid-Term Requirements**

Table 2-11: Technical Requirement TR_UeRM_11

Requirement ID:	TR_UeRM_11
Related SR(s):	• Sys_IntUeMan_9
Description: The UeRM shall maintain a list of login and logout operations.	
Rational: The users should know which persons have accessed the platform during specific incidents.	

Stimulus: The user with proper privileges (most probably an administrator) would request to see the list of login and logout operations for a specific period.
Response: The list of login and logout operations
Verification Criterion: Perform multiple request to retrieve the list for different periods.
Notes: none

Table 2-12: Technical Requirement TR_UeRM_12

Requirement ID:	TR_UeRM_12
Related SR(s):	• Sys_intUeMan_09 • Sys_IntUeMan_15
Description: The UeRM shall enable single sign-on and single sign-off.	
Rational: The users shall authenticate with the UeRM/HEIMDALL and not with the individual applications. This means that once signed in the users shall be able to access all applications/products/services they have access to instead of having to login again to access additional material.	
Stimulus: The user enters the login credentials in the HEIMDALL GUI.	
Response: A valid authentication token is returned by the system. Then it can be passed to other components.	
Verification Criterion: The user is able to use multiple HEIMDALL components without entering his/her credentials. Once the user is signed off, he/she cannot access HEIMDALL without entering valid credentials.	
Notes: none	

Table 2-13: Technical Requirement TR_UeRM_13

Requirement ID:	TR_UeRM_13
Related SR(s):	• Sys_IntUeMan_8
Description: Deletion of scenarios should only be allowed to users with the role of incident commander.	
Rational: Only Incident Commander should be authorised to delete scenarios from the system.	
Stimulus: A delete scenario command is send from a user	
Response: The scenario is deleted.	
Verification Criterion: The scenario is deleted only if the user is an incident commander. All other scenario deletion requests from users without this role are not executed.	
Notes: none	

Table 2-14: Technical Requirement TR_UeRM_14

Requirement ID:	TR_UeRM_14
Related SR(s):	• Sys_IntUeMan_2 • Sys_IntUeMan_10
Description: Deletion of scenario and lessons learnt templates is allowed to users with appropriate access rights.	
Rational: Only authorised users should be able to define scenario and lessons learnt templates.	
Stimulus: A template creation action is performed	
Response: The template is instantiated and the user is able to define its parameters	
Verification Criterion: Scenario and lessons learnt templates are created by authorised users. All other template creation requests from users without this authorisation are not executed.	
Notes: none	

Table 2-15: Technical Requirement TR_UeRM_15

Requirement ID:	TR_UeRM_15
Related SR(s):	• Sys_IntUeMan_2 • Sys_IntUeMan_11
Description: Modification of scenario information is allowed to users with appropriate access rights.	
Rational: Only authorised users should be able to modify scenario information	
Stimulus: A modification request to a scenario is performed.	
Response: The scenario information is updated and stored in the appropriate database.	
Verification Criterion: Multiple modification requests to various scenarios will be performed. Only the ones from authorised users will be executed.	
Notes: none	

Table 2-16: Technical Requirement TR_UeRM_16

Requirement ID:	TR_UeRM_16
Related SR(s):	• Sys_IntUeMan_2 • Sys_IntUeMan_12
Description: Modification of map symbology is allowed to users with appropriate access rights.	
Rational: Only authorised users should be able to modify the symbology	
Stimulus: A modification request of the symbology is performed.	
Response: The map symbology is modified and stored in user preferences	
Verification Criterion: Multiple map symbology modification requests will be performed. Only	

the ones from authorised users will be executed.
Notes: none

Table 2-17: Technical Requirement TR_UeRM_17

Requirement ID:	TR_UeRM_17
Related SR(s):	• Sys_IntUeMan_2 • Sys_IntUeMan_13
Description: Creation of map layers is allowed to users with appropriate access rights.	
Rational: Only authorised users should be able to create map layers	
Stimulus: The creation of map layer is requested (registration of a new layer in the system).	
Response: The new layer is registered in the system.	
Verification Criterion: Map layer creation requests to various scenarios will be performed. Only the ones from authorised users will be executed.	
Notes: none	

Table 2-18: Technical Requirement TR_UeRM_18

Requirement ID:	TR_UeRM_18
Related SR(s):	• Sys_IntUeMan_16
Description: The system shall allow access to the information gateway functionality of sending alert messages to only authorised users.	
Rational: Only authorised users should be able to send alert messages	
Stimulus: The creation of an alert message	
Response: The alert message is dispatched to the selected audience.	
Verification Criterion: Multiple alert messages are composed and dispatched. Only the ones from authorised users pass through the information gateway and reach the intended recipients.	
Notes: none	

2.2.3 **Long-Term Requirements**

Table 2-19: Technical Requirement TR_UeRM_19

Requirement ID:	TR_UeRM_19
Related SR(s):	• Sys_IntUeMan_17
Description:	

The system shall allow only authorised users to request assistance
Rational: Only authorised users should be able to request assistance following national/international agreements.
Stimulus: The user send an assistance request.
Response: The receiving party acknowledges the receipt of the request.
Verification Criterion: Multiple requests are sent, which their reception is acknowledged by the receiving party. The system blocks bequests sent by unauthorised users.
Notes: none

2.3 Other Requirements

2.3.1 **Short Term Requirements**

N/A based on Section 7.2 of D2.7.

2.3.2 **Mid-Term Requirements**

N/A based on Section 7.2 of D2.7.

2.3.3 **Long-Term Requirements**

N/A based on Section 7.2 of D2.7.

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 [2].

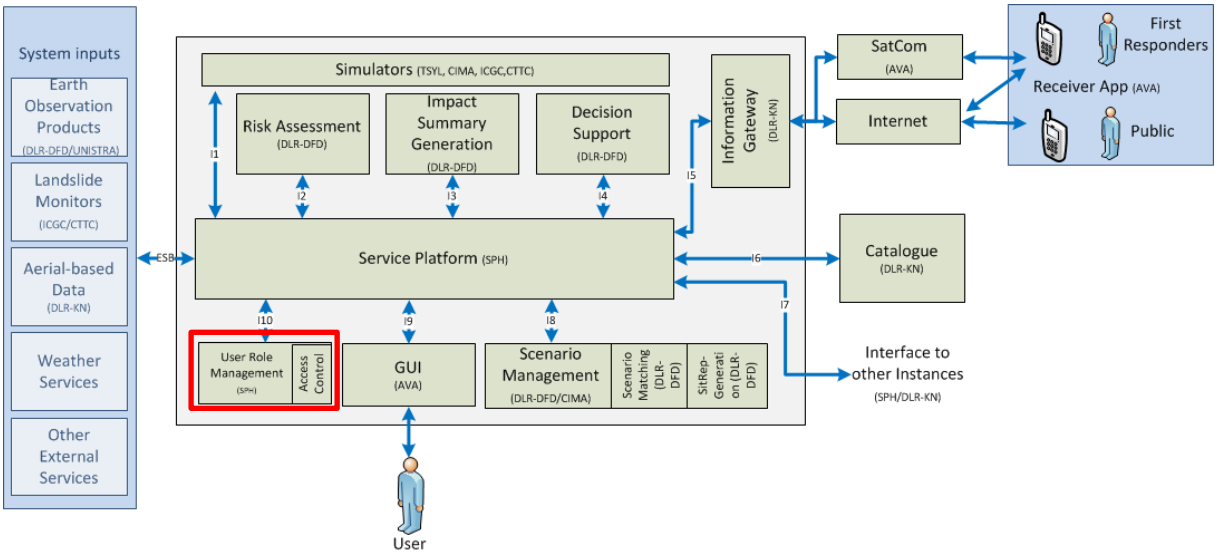

Figure 3-1: Local unit architecture.

The User and Role Management component, which includes the access control functionality, connects to the HEIMDALL system through the Service Platform (SP). As described in D4.1 [3], the core element of the HEIMDALL architecture is the Service Platform (SP) offered to each individual authority for response planning and scenario building. As shown in Figure 3-2, the UeRM consists of the following internal components:

- The policy enforcement component allows a user or an application/service to access the system based on the credentials provided, forwarding this information to the authorisation component.
- The authorisation component applies selective restriction to HEIMDALL resources (services/products and actions on them) based on (active) group the user belongs.
- The administration module allows the HEIMDALL administrators to manage all aspects of the UeRM service.
- The storage component holds the user preferences/profile.
- The policies components holds the group access policies.

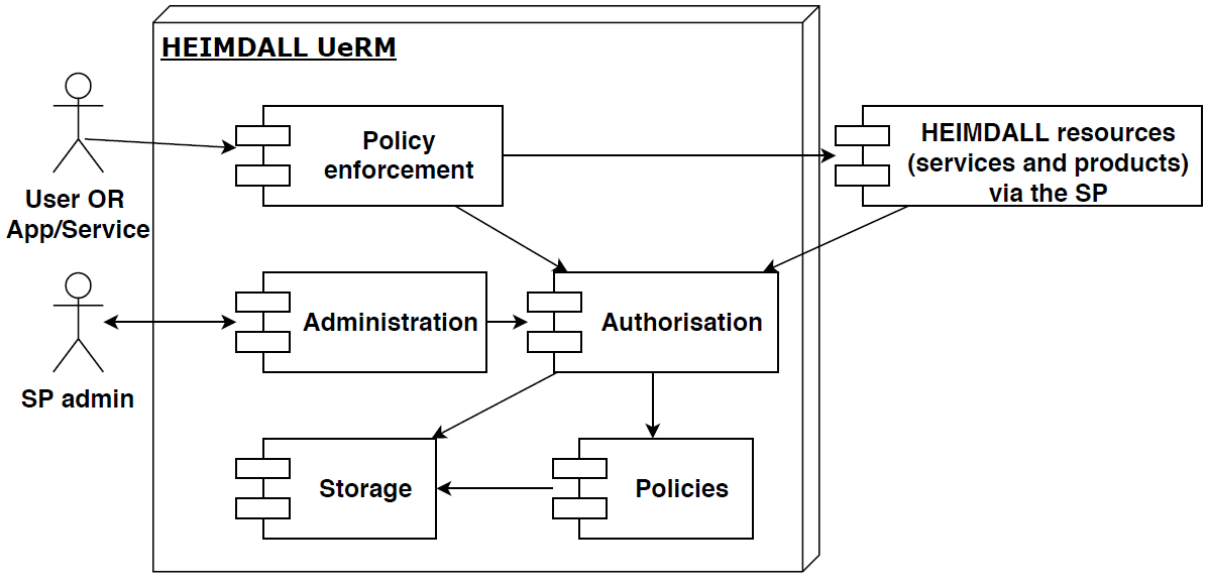

Figure 3-2: UeRM internal architecture.

3.2 Interface with the Service Platform

The UeRM interacts with the Service Platform for two purposes – a) for sending and retrieving data (login information, etc.) and b) for managing the user preferences. The SP provides access to other HEIMRALL data resources and functionality by use of different RESTful web services. Table 3-1 shows I10 as the interface connecting the UeRM with the SP.

Table 3-1: Interfaces with other components.

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

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

4 Module Functionality

The main entities that the UeRM uses in its access control mechanism are ***Group***, ***User***, ***Role***, ***Permission***, ***ResourceAccessRights*** and ***AccessRights***.

- ***Group*** Contains ***User(s)***.
- A ***User*** has a ***Role***.
- Each ***Role*** is a set of ***Permission(s)***.
- ***Permissions*** have *Types*.
- The platform consists of services and resources.
- Every resource/service has a ***ResourceAccessRight***, which specifies the owner of the resource (***User***/***Group***) and a set of ***AccessRights*** (READ/WRITE) to the resource for owning group's users and other system's users.

The permission types can be expressed by numerical values. Below (Table 4-1) is a list of valid permission and their corresponding numerical values.

Table 4-1: Association of numerical values to permission types.

Permission type	Numerical value
CreateGroup	0
CreateUser	1
UpdateGroup	2
UpdateUser	3
DeleteUser	5
AssignUserToGroup	6
AssignPermissiontoUser	7
DeletePermission	8

At first, it was considered to utilise the access control features of the GIS engine (Geoserver) as they were already available. However, they were proven not sufficient:

1) to fulfil the whole set of security-related system requirements; and
2) to cover all information exchange (e.g. data publication, exchange of non-geospatial data).

Upon uploading/publishing to the data repository, again, the user has to provide his/her ID. In this case, the publisher can also control the access of the other users to the published data. In order to do so, the user must append an extra parameter, which defines the access policy for the resource, to the URL. Hence, the UeRM module shall allow a user or an application/service to access the system based on the credentials provided (specifying access rights/privileges to resources). Only users/applications/services with valid credentials will be allowed to access the system. The access control module shall apply selective restriction to HEIMDALL resources (services/products and actions on them). Valid users will have access to the resources based on their role and access rights. The read, marked as "R", and write (create, update and delete), marked as "W", access of the various HEIMDALL roles to the system components is presented in Table 4-2.

Table 4-2: Access rights of roles to HEIMDALL modules.

	GUI [1]		Mobile app [2]		Simulator [3]		Decision support		Scenario management		Impact assessment		External systems [4]		UeRM [5]		Catalogue [6]		Information gateway [7]	
	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W
Control room chief	Y		Y		Y	N	Y		Y		Y		Y		Y		Y		Y	
Incident commander	Y		Y		Y	N	Y		Y		Y		Y		Y		Y		Y	
Fire analyst	Y		N		Y		Y	N	Y		Y		Y		Y		Y		Y	N
Landslide analyst	Y		N		Y		Y	N	Y		Y		Y		Y		Y		Y	N
Flood analyst	Y		N		Y		Y	N	Y		Y		Y		Y		Y		Y	N
First responder (field)[8]	N		Y		Y	N	Y	N	Y	N	Y	N	Y	N	Y		Y	N	Y	N
System	Y		Y		Y		Y		Y		Y		Y		Y		Y		Y	

[1] A read/write access to the GUI allows the user to access the GUI and modify the way the information is presented (linked with the UeRM component).

[2] This application is focused on the responders, fire and rescue services and police, deployed in the field.

[3] Only personnel with these roles, having the capable scientific and engineering skills, will be allowed to trigger simulations (e.g., initiate new simulations, modify their parameters, etc.). The rest of the HEIMDALL users will be able to see the simulation outcomes.

[4] All users are able to see information coming from external systems (e.g., weather updates, EO products, etc.), however only the ones with write access will be able to request new resources (e.g., request a new weather update).

[5] Through this component the users are able to modify their settings, and other aspects of their accounts.

[6] The users are able to share information through the catalogue (write access) and read information from there, if this is share to their role and group they belong to.

[7] The control room chief and the incident commanders are the ones authorized to create and dispatch information messages through the GUI and dispatched to personnel in (selected) areas through the information gateway component.

[8] The personnel deployed in the field will mainly use the HEIMDALL application, hence they have full read/write access, whereas they will be able to read the information coming from the other components of the system.

administrator [9]										

To the purpose of achieving the above mentioned features, the UeRM has been designed and implemented as a "layer". In this framework, each user belongs to a specific Group and is provided with a unique ID. Each request to the HEIMDALL system, through the SP, should be accompanied with the corresponding user ID. The UeRM, receives this through the SP, and decides whether the user has access or not to the requested resource. Table 4-3 summarises the services provided by UeRM.

Table 4-3: UeRM 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***
Authentication	• Username and password of the user • A valid token, in the case a token-based method is used	• GUI (such action is triggered by the GUI)	• UeRM • The token is received by other HEIMDALL components
Access control	• Valid login credentials (successful authentication) • Active role of the user (selected by the UI or provided by the UeRM)	• GUI (such action is triggered by the GUI) • UeRM	• UeRM • SP

Through the admin console, the UeRM administrators can centrally manage all aspects of the user management server, whereas through the account management console, users can manage their own accounts. In addition, the user profile shall hold their preferences, as shown in D2.11, facilitating a smother operation from the user perspective. Table 4-4 summarises the UeRM management services.

Table 4-4: UeRM 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***
Admin console	Valid admin credentials	• GUI (such action is triggered by the GUI)	• UeRM

[9] The system administrator has full access to the HEIMDALL platform, only for administrative and maintenance functions, not interfering to the operational aspects of the various workflows.

Account management console	Valid user credentials	• GUI (such action is triggered by the GUI)	• UeRM
User profile	The user preferences as shown in D2.11.	• UeRM	• GUI • The other HEIMDALL components

5 Technical Specification

The entire HEIMDALL platform as well as the UeRM 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.

5.1 User login service API

In order for any user or application to be able to interact with the HEIMDALL system and the UeRM, 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.eyJ1bmlxdWVfbmFtZSI6ImNyYyIsImh0dHA6Ly9zY2hlbWFzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvaWRlbnRpdHkvY2xhaW1zL3NpZCI6IjM2MDQ4NjA0LTQzNzUtNDRjZC04M2E2LTVjZTIwMzE3NzViNiIsInJvbGUiOiJDb250cm9sIFJvb20gQ2hpZWYiLCJwcmltYXJ5c2lkIjoiNTdkYTlhMjgtMzdmMy00NDZjLTk2M2MtNzZkZGUzMDY1NmUzIiwibmJmIjoxNTQ1MDUyNDQyLCJleHAiOjE1NDUxMzg4NDIsImlhdCI6MTU0NTA1MjQ0Mn0.JIK5y_UVT7ofNhNrwwPsAxp8uDH8De30XdfpZrTjZjs",
    "expires": "20181218T131202"
```

```
}
```

5.2 Retrieve operations

In the following sections the retrieve operations are presented.

5.2.1 **Retrieve all users**

The user is able to retrieve the list of users through the service presented in Table 5-2.

Table 5-2: Retrieve users service.

Service ID	**UeRM_retrieve_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	List of users
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

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

A part of the response is as follows:

```
[
    {
        "UserId": "00000000-0000-0000-0000-000000000000",
        "Group": {
            "GroupId": "00000000-0000-0000-0000-000000000000",
            "Name": "AVANTI",
            "Description": "Avanti",
            "GroupOwner": null,
            "ByteVersion": null,
            "Id": 2
        },
        "Name": "Demo R2",
        "UserName": "demor2",
```

```
        "EMail": "demor2@avanti.com",
        "Password": null,
        "Photo": null,
        "SessionId": "cddc9463-40fe-4e6e-bb97-f86acf61a538",
        "SessionValidUntil": "2018-09-11T13:34:16",
        "Role": null,
        "IsFirstResponder": true,
        "LastSeen": "2016-02-22T11:57:47",
        "Longitude": -0.1024384,
        "Latitude": 51.5131884,
        "DeviceId": null,
        "ByteVersion": null,
        "Id": 41
    },
…
    {
        "UserId": "00000000-0000-0000-0000-000000000000",
        "Group": {
            "GroupId": "00000000-0000-0000-0000-000000000000",
            "Name": "TEST GROUP",
            "Description": "Group of test users",
            "GroupOwner": null,
            "ByteVersion": null,
            "Id": 3
        },
        "Name": "Test User",
        "UserName": "testuser",
        "EMail": "test@space.gr",
        "Password": null,
        "Photo": null,
        "SessionId": "bb1b64f7-833e-4714-bb32-2a053e5b3e17",
        "SessionValidUntil": "2018-11-30T14:56:53",
        "Role": null,
        "IsFirstResponder": false,
        "LastSeen": "2018-09-06T14:22:10",
        "Longitude": 2.234,
        "Latitude": 40.313,
```

```
        "DeviceId": null,
        "ByteVersion": null,
        "Id": 3
    },


    {
        "UserId": "00000000-0000-0000-0000-000000000000",
        "Group": {
            "GroupId": "00000000-0000-0000-0000-000000000000",
            "Name": "FIRE-1",
            "Description": "Group of FIRE-1 Run",
            "GroupOwner": null,
            "ByteVersion": null,
            "Id": 10
        },
        "Name": "Angel Grablev",
        "UserName": "angel",
        "EMail": "Angel.Grablev@avantiplc.com",
        "Password": null,
        "Photo": null,
        "SessionId": "b8233c57-c87a-48e5-b324-4a390bc96b6e",
        "SessionValidUntil": "2018-10-01T12:25:04",
        "Role": null,
        "IsFirstResponder": true,
        "LastSeen": "2018-07-24T09:02:00",
        "Longitude": 2.8241983,
        "Latitude": 47.6758983,
        "DeviceId": null,
        "ByteVersion": null,
        "Id": 2
    }
]
```


5.2.2 **Retrieve information about user**

The HEIMDALL user is able to retrieve information about his/her own account through the service summarised in Table 5-3.

Table 5-3: Retrieve user information service.

Service ID	UeRM_retrieve_02
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Information about the user
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

```
GET http://esb.heimdall.sp/services/rest/users/me
```

Below is a sample response:

```
{
    "UserId": "36048604-4375-44cd-83a6-5ce2031775b6",
    "Group": null,
    "Name": "Control Room Chief",
    "UserName": "crc",
    "EMail": "crc@shrd.com.gr",
    "Password": "crc",
    "Photo": null,
    "SessionId": "57da9a28-37f3-446c-963c-76dde30656e3",
    "SessionValidUntil": "2018-12-18T13:14:02",
    "Role": {
        "Name": "Control Room Chief",
        "HomePage": null,
        "Permissions": [],
        "ByteVersion": null,
        "Id": 46
    },
    "IsFirstResponder": false,
    "LastSeen": "0001-01-01T00:00:00",
```

```
    "Longitude": 0,
    "Latitude": 0,
    "DeviceId": null,
    "ByteVersion": null,
    "Id": 54
}
```

5.2.3 **Retrieve all groups**

The HEIMDALL user is able to retrieve the existing groups through the service presented in Table 5-4.

Table 5-4: Retrieve all groups service.

Service ID	**UeRM_retrieve_03**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	List of the groups
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

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

A sample of the response is as follows:

```
[
    {
        "GroupId": "00000000-0000-0000-0000-000000000000",
        "Name": "TEST GROUP",
        "Description": "Group of test users",
        "GroupOwner": null,
        "ByteVersion": null,
        "Id": 3
    },
    {
```

```
        "GroupId": "00000000-0000-0000-0000-000000000000",
        "Name": "Another Test Group",
        "Description": null,
        "GroupOwner": null,
        "ByteVersion": null,
        "Id": 4
    },
    {
        "GroupId": "00000000-0000-0000-0000-000000000000",
        "Name": "atest",
        "Description": "A-Test",
        "GroupOwner": null,
        "ByteVersion": null,
        "Id": 5
    },
    {
        "GroupId": "00000000-0000-0000-0000-000000000000",
        "Name": "TSYL",
        "Description": "TechnoSylva",
        "GroupOwner": null,
        "ByteVersion": null,
        "Id": 6
    },
    {
        "GroupId": "00000000-0000-0000-0000-000000000000",
        "Name": "DLR-DFD",
        "Description": "DLR",
        "GroupOwner": null,
        "ByteVersion": null,
        "Id": 7
    },
    {
        "GroupId": "00000000-0000-0000-0000-000000000000",
        "Name": "SPMM",
        "Description": "Spmm.org",
        "GroupOwner": null,
        "ByteVersion": null,
        "Id": 8
```

```
    },
…
]
```

5.2.4 **Retrieve a single group**

The HEIMDALL user is able to retrieve the existing groups through the service presented in Table 5-5.

Table 5-5: Retrieve a single group service.

Service ID	**UeRM_retrieve_04**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Group information
Operations	N/A
Main parameters	Numerical ID of the group of GUID
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

```
GET http://esb.heimdall.sp/services/rest/groups/<Numerical Id>
```

Where `Numerical Id` is the Id of the group, or

```
GET http://esb.heimdall.sp/services/rest/groups?groupId=<GUID>
```

The response for the group with the Id=2 is as follows:

```
{
    "GroupId": "00000000-0000-0000-0000-000000000000",
    "Name": "AVANTI",
    "Description": "Avanti",
    "GroupOwner": null,
    "ByteVersion": null,
    "Id": 2
}
```

5.2.5 **Retrieve user's owned groups**

With the call, summarized in Table 5-6, a user can retrieve a list of all the groups that he owns.

Table 5-6: Retrieve own groups service.

Service ID	**UeRM_retrieve_05**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Group information
Operations	N/A
Main parameters	Numerical ID of the user
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a "401 Unauthorised" response.

Through the following `GET` request the user can retrieve the groups he/she is member of:

```
GET http://esb.heimdall.sp/services/rest/groups/userId=<GUID>
```

5.2.6 **Retrieve Access Rights for a single user**

With the following call, summarized in Table 5-6, a user can retrieve a list of all the groups that he/she owns.

Table 5-7: Retrieve access rights service.

Service ID	**UeRM_retrieve_06**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Access rights
Operations	N/A
Main parameters	Numerical ID of the user and (if needed) the access type
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a "401 Unauthorised" response.

Through the following `GET` request the user can retrieve the groups he/she is member of:

```
GET  http://esb.heimdall.sp/services/rest/access?userId=f0a19e04-301d-47af-a5bc-bed50d17d254
```

A sample of the response is:


```
[
    {
        "Id": 77,
        "ByteVersion": null,
        "Group": null,
        "User": null,
        "ResourceUrn": "pharos:sm_04a9ffcd-1a84-4f6a-8cf7-dbf955f5715d",
        "Rights": {
            "GroupCanRead": true,
            "GroupCanWrite": false,
            "OtherCanRead": true,
            "OtherCanWrite": false,
            "ByteVersion": null,
            "Id": 64
        },
        "IsOwner": false
    },
…
    {
        "Id": 693,
        "ByteVersion": null,
        "Group": null,
        "User": null,
        "ResourceUrn": "heimdall:ffs_LadvnZ4uUiJLOWQ9JmTQ_simflamelength",
        "Rights": {
            "GroupCanRead": true,
            "GroupCanWrite": false,
            "OtherCanRead": true,
            "OtherCanWrite": false,
            "ByteVersion": null,
            "Id": 675
        },
```

```
        "IsOwner": false
    },
    {
        "Id": 694,
        "ByteVersion": null,
        "Group": null,
        "User": null,
        "ResourceUrn":
"heimdall:ffs_LadvnZ4uUiJLOWQ9JmTQ_firebehaviourindex",
        "Rights": {
            "GroupCanRead": true,
            "GroupCanWrite": false,
            "OtherCanRead": true,
            "OtherCanWrite": false,
            "ByteVersion": null,
            "Id": 676
        },
        "IsOwner": false
    }
]
```


Or

GET http://esb.heimdall.sp/services/rest/access?userId=f0a19e04-301d-47af-a5bc-bed50d17d254&access=3

{access=1 returns the resources with read access, where access=3 returns the resources with write access. If *access* is omitted it defaults to 1}

Please find below a sample response.


```
[
    {
        "Label": "Municipalities",
        "WmsUrl": "http://esb.heimdall.sp/services/ogc/pharos/wms",
        "Group": null,
        "User": null,
        "ResourceUrn": "pharos:municipis",
        "Rights": {
            "GroupCanRead": true,
            "GroupCanWrite": true,
```

```
            "OtherCanRead": true,
            "OtherCanWrite": true,
            "ByteVersion": null,
            "Id": 14
        },
        "Type": 1,
        "IsAvailable": true,
        "IsExternal": false,
        "IsBaseLayer": false,
        "Metadata": {
            "__interceptor": {
                "persistentClass":
"Space.AccessControl.Entities.ResourceMetadata,
Space.AccessControl.Entities,       Version=1.0.0.0,       Culture=neutral,
PublicKeyToken=null",
                "getIdentifierMethod": {
                    "Name": "get_Id",
                    "AssemblyName":          "Space.AccessControl.Entities,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=null",
                    "ClassName":
"Space.AccessControl.Entities.ResourceMetadata",
                    "Signature":           "System.Nullable`1[System.Int32]
get_Id()",
                    "Signature2":         "System.Nullable`1[[System.Int32,
mscorlib,                Version=4.0.0.0,                Culture=neutral,
PublicKeyToken=b77a5c561934e089]] get_Id()",
                    "MemberType": 8,
                    "GenericArguments": null
                },
                "setIdentifierMethod": {
                    "Name": "set_Id",
                    "AssemblyName":          "Space.AccessControl.Entities,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=null",
                    "ClassName":
"Space.AccessControl.Entities.ResourceMetadata",
                    "Signature":                                        "Void
set_Id(System.Nullable`1[System.Int32])",
                    "Signature2":                                   "System.Void
set_Id(System.Nullable`1[[System.Int32,     mscorlib,     Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089]])",
                    "MemberType": 8,
                    "GenericArguments": null
```

```
                },
                "overridesEquals": false,
                "componentIdType": null,
                "_target": null,
                "initialized": false,
                "_id": 14,
                "unwrap": false,
                "_entityName":
"Space.AccessControl.Entities.ResourceMetadata",
                "readOnly": false,
                "readOnlyBeforeAttachedToSession": null
            },
            "__baseType":   "Space.AccessControl.Entities.ResourceMetadata,
Space.AccessControl.Entities,       Version=1.0.0.0,       Culture=neutral,
PublicKeyToken=null",
            "__baseInterfaceCount": 1,
            "__baseInterface0":         "NHibernate.Proxy.INHibernateProxy,
NHibernate,               Version=4.1.0.4000,               Culture=neutral,
PublicKeyToken=aa95f207798dfdb4"
        },
        "ByteVersion": null,
        "Id": 14
    },
…
    {
        "Label": "IG Alert Areas",
        "WmsUrl": "http://esb.heimdall.sp/services/ogc/heimdall/wms",
        "Group": null,
        "User": null,
        "ResourceUrn": "heimdall:alertarea",
        "Rights": {
            "GroupCanRead": true,
            "GroupCanWrite": true,
            "OtherCanRead": true,
            "OtherCanWrite": true,
            "ByteVersion": null,
            "Id": 445
        },
        "Type": 1,
```

```
        "IsAvailable": true,
        "IsExternal": false,
        "IsBaseLayer": false,
        "Metadata": {
            "__interceptor": {
                "persistentClass":
"Space.AccessControl.Entities.ResourceMetadata,
Space.AccessControl.Entities,       Version=1.0.0.0,       Culture=neutral,
PublicKeyToken=null",
                "getIdentifierMethod": {
                    "Name": "get_Id",
                    "AssemblyName":          "Space.AccessControl.Entities,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=null",
                    "ClassName":
"Space.AccessControl.Entities.ResourceMetadata",
                    "Signature":           "System.Nullable`1[System.Int32]
get_Id()",
                    "Signature2":         "System.Nullable`1[[System.Int32,
mscorlib,                 Version=4.0.0.0,                 Culture=neutral,
PublicKeyToken=b77a5c561934e089]] get_Id()",
                    "MemberType": 8,
                    "GenericArguments": null
                },
                "setIdentifierMethod": {
                    "Name": "set_Id",
                    "AssemblyName":          "Space.AccessControl.Entities,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=null",
                    "ClassName":
"Space.AccessControl.Entities.ResourceMetadata",
                    "Signature":                                        "Void
set_Id(System.Nullable`1[System.Int32])",
                    "Signature2":                                "System.Void
set_Id(System.Nullable`1[[System.Int32,     mscorlib,     Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089]])",
                    "MemberType": 8,
                    "GenericArguments": null
                },
                "overridesEquals": false,
                "componentIdType": null,
                "_target": null,
                "initialized": false,
                "_id": 34,
```

```
                "unwrap": false,
                "_entityName":
"Space.AccessControl.Entities.ResourceMetadata",
                "readOnly": false,
                "readOnlyBeforeAttachedToSession": null
            },
            "__baseType":   "Space.AccessControl.Entities.ResourceMetadata,
Space.AccessControl.Entities,       Version=1.0.0.0,       Culture=neutral,
PublicKeyToken=null",
            "__baseInterfaceCount": 1,
            "__baseInterface0":         "NHibernate.Proxy.INHibernateProxy,
NHibernate,              Version=4.1.0.4000,              Culture=neutral,
PublicKeyToken=aa95f207798dfdb4"
        },
        "ByteVersion": null,
        "Id": 464
    }
]
```


5.3 Create, Update and Delete operations

In the following sections the create, update and delete operations are presented.

5.3.1 **Create Group**

With the following call, summarized in Table 5-8, a user can create a new group.

Table 5-8: Create group service.

Service ID	**UeRM_create_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User id and new group information
Operations	N/A
Main parameters	Numerical id of the user group information (JSON file)
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	JSON file holding the GroupID
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

A group can be created by a user that has the necessary permissions. By default, the user that creates the group becomes the owner of the Group. Through the following `POST` operation the user can create a group:

```
POST http://esb.heimdall.sp/services/rest/groups?userId={The id of the user trying to create a new group}

{
   "Name":"TEST",
   "Description":"A Test Group"
}

Return Value: If success, the new GroupId will be returned.
```

5.3.2 **Create User – No group assignment**

With the following call, summarized in Table 5-9, a user can create another user.

Table 5-9: Create user service.

Service ID	**UeRM_create_02**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User id
Operations	N/A
Main parameters	Numerical id of the user
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	JSON holding the user ID
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

A user can be created by another user that has the necessary permissions:

```
POST http://esb.heimdall.sp/services/rest/users?userId={The id of the user trying to create a new user}

{
   "Name":"A new test user",
   "UserName":"testusr",
   "EMail":"testusr@space.gr",
   "Password":"password",
   "Role":{
      "Name":"testusr Role",
```

```
      "Permissions":[
         {
            "Type":0
         },
         {
            "Type":1
         }
      ]
   }
}
```

Return Value: If success, the new UserId will be returned.

Comments: UserName and EMail are mandatory unique properties.

5.3.3 **Create User – Group assignment**

A User can be created by a group owner and assigned to that group with a single REST calls. With the following call, summarized in Table 5-10, a user can create another user and assign him/her to an existing group.

Table 5-10: Create user service.

Service ID	**UeRM_create_03**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User and group id
Operations	N/A
Main parameters	Numerical id of the user group information (JSON file)
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	JSON holding the user ID
Notes	Without a successful login operation the UeRM 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/users?userId={The id of the user trying to create a new user}&groupId={The group owned by userId}

{
   "Name":"A new test user",
   "UserName":"testusr",
   "EMail":"testusr@space.gr",
```

```
   "Password":"password",
   "Role":{
      "Name":"testusr Role",
      "Permissions":[
         {
            "Type":0
         },
         {
            "Type":1
         }
      ]
   }
}

Return Value: If success, the new UserId will be returned.
Comments: UserName and EMail are mandatory unique properties. Permissions and other properties can be omitted.
```

5.3.4 **Assign user to group**

With the following call, summarized in Table 5-11, a user can assign another user to an existing group.

Table 5-11: Assign user to group service.

Service ID	**UeRM_assign_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Ids of the users and the target group
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP 200 upon successful completion
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

A user that has the permission type AssignUserToGroup=7 and is the owner of a group can assign a user to a group through the following `POST` operation.

```
POST http://esb.heimdall.sp/services/rest/groups?userId{GUID of the assigner}&groupId{GUID of the destination group}&joinUserId={GUID of the assignee}


Return Value: If success, HTTP 200 OK
```

5.3.5 **Grant permission to user**

With the following call, summarized in Table 5-12, a user can assign another user to an existing group.

Table 5-12: Grant permissions to user service.

Service ID	**UeRM_grant_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User id and permission type numerical values
Operations	N/A
Main parameters	Numerical id of the user and permission type
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP 200 upon successful completion
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

In order to give permission to another user, the user must have the appropriate permissions and be the owner of the group that the assignee belongs to. This operation can be performed through the following `POST` operation.

```
POST http://esb.heimdall.sp/services/rest/permissions?userId&assignedUserId


<PermissionType Numerical Value, e.g. 0 for CreateGroup>
```

5.3.6 **Revoke permission from user**

With the following call, summarized in Table 5-13, a user can assign another user to an existing group.

Table 5-13: Revoke permissions service.

Service ID	**UeRM_revoke_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User id and permission type numerical values
Operations	N/A
Main parameters	Numerical id of the user and permission type

Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP 200 upon successful completion
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

With the REST call below a specific permission is removed (revoked) from the assignedUserId. This operation can be performed through the following `DELETE` operation.

```
DELETE
http://esb.heimdall.sp/services/rest/permissions?userId&assignedUserId

<PermissionType Numerical Value, e.g. 0 for CreateGroup>
```

5.3.7 **Delete user**

With the following call, summarized in Table 5-14, a user can assign another user to an existing group.

Table 5-14: Delete user service.

Service ID	**UeRM_delete_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	User id and permission type numerical values
Operations	N/A
Main parameters	Numerical id of the user and permission type
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP 200 upon successful completion
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

This operation can be performed through the following `DELETE` operation.

```
DELETE http://esb.heimdall.sp/services/rest/users?userId&deleteUserId
```

5.4 User settings

The user can access the settings functionality, read and write operations, through the REST API presented in the following sections.

5.4.1 **Fetch Settings**

In order to fetch all settings of a user (Global, Group and User scope settings), the API presented in Table 5-15 is used.

Table 5-15: Fetch all user settings service.

Service ID	**UeRM_settings_01**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	N/A
Operations	N/A
Main parameters	N/A
Data representation protocol	JSON
Communication protocol	HTTP (GET)
Response	JSON holding the list of settings
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

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

A successful call, returns a list of settings applying override rules, i.e. if the same setting exists for Group and User scopes the list will contain only the User setting. A sample of the return is provided below:

```
[
    {
        "Id": 5,
        "Name": "logoUrl",
        "Value":                                "http://heimdall-h2020.eu/wp-
content/uploads/2017/11/cropped-01_HEIMDALL_Logo_w-1.png",
        "Scope": "User",
        "OverriddenByScope": null
    },
    {
        "Id": 4,
        "Name": "secondaryUrl",
        "Value":                                "http://heimdall-h2020.eu/wp-
content/uploads/2017/11/cropped-01_HEIMDALL_Logo_w-1.png",
        "Scope": "Group",
        "OverriddenByScope": "User"
```

```
    }
]
```

5.4.2 **Add Setting**

In order to add a new setting the API presented in Table 5-16 is used.

Table 5-16: Add a new setting service.

Service ID	**UeRM_settings_02**
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Setting information (JSON)
Operations	N/A
Main parameters	Name, value and scope of setting
Data representation protocol	JSON
Communication protocol	HTTP (POST)
Response	HTTP 200 upon successful operation
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

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


{
  "name" : "logoUrl",
  "value"  :  "http://heimdall-h2020.eu/wp-content/uploads/2017/11/cropped-01_HEIMDALL_Logo_w-1.png" ,
  "scope" : "Group",
}
name: Name of the setting
value: Value of the setting
scope: Scope of the setting. Can be Global/Group/User. Only sysadmin account can POST Global-scope settings and Group owner Group-scope settings.
overriddenbyscope (optional) : If not set (null) the setting cannot be overridden. Can be set to Group/User for an existing Global-scoped setting and to User for Group-scoped setting.
```

5.4.3 **Update Setting**

In order to update an existing setting the API presented in Table 5-17 is used.

Table 5-17: Update existing setting service.

Service ID	UeRM_settings_03
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	Setting information (JSON)
Operations	N/A
Main parameters	ID, name, value and scope of setting
Data representation protocol	JSON
Communication protocol	HTTP (PUT)
Response	HTTP 200 upon successful operation
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

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

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


{
  "Id" : 6
  "name" : "logoUrl",
  "value"  :  "http://heimdall-h2020.eu/wp-content/uploads/2017/11/cropped-01_HEIMDALL_Logo_w-4.png",
  "scope" : "Group",
  "overriddenbyscope": "User"
}
```

Update works if user is owner of the setting (for user-scope settings), owner of the group (for group-scope settings), sysadmin for global settings.

5.4.4 **Delete Setting**

In order to delete an existing setting the API presented in Table 5-18 is used.

Table 5-18: Delete setting service.

Service ID	UeRM_settings_04
Assumed consumers (via reference point)	All modules of HEIMDALL
Data exchanged	N/A
Operations	N/A
Main parameters	ID
Data representation protocol	JSON

Communication protocol	HTTP (PUT)
Response	HTTP 200 upon successful operation
Notes	Without a successful login operation the UeRM will not accept the incoming request; they will be rejected and the user will get a “401 Unauthorised” response.

The following example shows a `DELETE` settings request.

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

6 Test Plan and Report

This section contains the list of tests designed and performed targeting the necessary features 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 UeRM 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, developing the modules that interact with the UeRM, 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_UeRM_#"*
Requirements to be verified	*List of technical and system requirements that this test verifies in the form* • *TR_UeRM_#* ○ *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 / PARTIAL / NOT_PERFORMED

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 UeRM defined in Section 2. The list presented here captures the status of the "Release A"-ready HEIMDALL UeRM, and during the project evolution is will be enriched and updated. In this way we will test all features of the UeRM moving towards the integrated prototype and final demonstration.

Table 6-2: TS_UeRM_01: The user is able to login.

Test ID	*TS_UeRM_01*

Requirement to be verified	• *TR_UeRM_9*
	  ○ *Sys_IntUeMan_1*
	• *TR_UeRM_11*
	  ○ *Sys_IntUeMan_9*

	• *TR_UeRM_12*
	  ○ *Sys_intUeMan_9*
	  ○ *Sys_IntUeMan_15*

Test objective	User is able to login
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user starts the web portal and logs in.
Test prerequisites/ configuration	• The web portal needs to be up and running.
Success criteria	The system returns a valid authentication token.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-3: TS_UeRM_02: The user is able to retrieve the list of login and logout operations.

Test ID	*TS_UeRM_02*
Requirement to be verified	• *TR_UeRM_11* ○ *Sys_IntUeMan_9*
Test objective	The user is able to retrieve the list of login and logout operations.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user starts the web portal and logs in. 3. The user requests the list of login and logout operations
Test prerequisites/ configuration	• The web portal needs to be up and running.
Success criteria	The list of login and logout operations are displayed in the user's screen
Results analysis	*N/A*
Success	NOT_PERFORMED

Table 6-4: TS_UeRM_03: The user is able to store his/her own preferences/settings.

Test ID	*TS_UeRM_03*

Requirement to be verified	• *TR_UeRM_01*
	  ○ *Sys_IntData_4*
	  ○ *Sys_Gui_8*
	  ○ *Sys_Gui_10*
	  ○ *Sys_Gui_20*
	  ○ *Sys_Gui_116*
	  ○ *Sys_IntUeMan_12*
	  ○ *Sys_IntUeMan_18*

Test objective	Store and retrieve the settings of a user
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user stores his/her own settings 3. The user retrieves the settings for validation of the prior action.
Test	• The SP should be operational.

prerequisites/ configuration	
Success criteria	The settings retrieved should be the ones set by the user.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-5: TS_UeRM_04: The system administrator should be able to create and modify groups.

Test ID	*TS_UeRM_04*

Requirement to be verified	• *TR_UeRM_02*
	  ○ *Sys_IntUeMan_1*
	  ○ *Sys_IntData_4*

Test objective	Create and modify user groups
Test procedure	1. The system administrator connects to the HEIMDALL VPN. 2. The system administrator creates a group, then retrieves the group information for validation of the prior action. 3. The system administrator modifies a group, then retrieves the group information for validation of the prior action.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The group information retrieved should be the ones set by the system administrator.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-6: TS_UeRM_05: The system administrator should be able to assign users to groups.

Test ID	*TS_UeRM_05*

Requirement to be verified	• *TR_UeRM_03*
	  ○ *Sys_IntUeMan_1*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_3*
	  ○ *Sys_IntData_4*

Test objective	Assign users to group
Test procedure	1. The system administrator connects to the HEIMDALL VPN. 2. The system administrator modifies the user(s) to group(s) assignments.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The group information retrieved should be the ones set by the system administrator.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-7: TS_UeRM_06: The system administrator should be able to create and modify roles.

Test ID	*TS_UeRM_06*

Requirement to be verified	• *TR_UeRM_04*
	  ○ *Sys_IntUeMan_1*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_3*
	  ○ *Sys_IntData_4*

Test objective	Create and modify roles
Test procedure	1. The system administrator connects to the HEIMDALL VPN. 2. The system administrator modifies the system roles.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The role information retrieved should be the ones set by the system administrator.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-8: TS_UeRM_07: The system administrator should be able to assign roles to users.

Test ID	*TS_UeRM_07*

Requirement to be verified	• *TR_UeRM_05*
	  ○ *Sys_IntUeMan_1*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_3*
	  ○ *Sys_IntData_4*

Test objective	Assign roles to users
Test procedure	1. The system administrator connects to the HEIMDALL VPN. 2. The system administrator modifies the user(s) to role(s) assignments.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The role information retrieved should be the ones set by the system administrator.
Results analysis	*The test has been performed and passed according to the success criteria. The role configuration has not been finalised at this stage of the project. The roles as well the user assignment will be refined based on the feedback collected during the project activities and finalised for D4.5.*
Success	PARTIAL

Table 6-9: TS_UeRM_08: The system administrator has access to the administration console.

Test ID	*TS_UeRM_08*

Requirement to be verified	• *TR_UeRM_06*
	  ○ *Sys_IntUeMan_1*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_3*

	○ *Sys_IntUeMan_4* ○ *Sys_IntData_4*
Test objective	Assess the functionality of the administration console
Test procedure	1. The system administrator connects to the HEIMDALL VPN. 2. The system administrator manages the configuration of the UeRM, managing users, roles and groups.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The system administrator is able to modify the operational parameters and overall configuration of the UeRM
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-10: TS_UeRM_09: The user has access to the user account console.

Test ID	*TS_UeRM_09*
Requirement to be verified	• *TR_UeRM_07* ○ *Sys_IntUeMan_4* ○ *Sys_IntUeMan_9* ○ *Sys_IntUeMan_15* ○ *Sys_IntData_4*
Test objective	Assess the functionality of the user account console
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user is able to: ○ Change their own passwords ○ Manage sessions ○ View history of the account ○ Modify their profile (user preferences).
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The user is able to modify the aforementioned parameters and overall configuration of his/her account
Results analysis	*At the current stage the user is able to modify his/her preferences/settings*
Success	PARTIAL

Table 6-11: TS_UeRM_10: The user is able to grant access to other users.

Test ID	*TS_UeRM_10*
Requirement to be verified	• *TR_UeRM_08* ○ *Sys_IntUeMan_5* ○ *Sys_IntUeMan_6* ○ *Sys_IntUeMan_7*
Test objective	Grant access to other users, for the data/products the user has permission.
Test procedure	1. The user connects to the HEIMDALL VPN.

	2. The user grants access to other users for the specific data he/she has permission to do so.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The user is able to modify the access level of data and products, enabling other users to access it.
Results analysis	
Success	NOT_PERFORMED

Table 6-12: TS_UeRM_11: The UeRM stores users, roles and their profiles.

Test ID	*TS_UeRM_11*

Requirement to be verified	• *TR_UeRM_06*
	  ○ *Sys_IntUeMan_1*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_3*
	  ○ *Sys_IntUeMan_4*
	  ○ *Sys_IntData_4*
	• *TR_UeRM_07*
	  ○ *Sys_IntUeMan_4*
	  ○ *Sys_IntUeMan_9*
	  ○ *Sys_IntUeMan_15*
	  ○ *Sys_IntData_4*
	• *TR_UeRM_10*
	  ○ *Sys_IntData_4*

Test objective	Validate the capability of UeRM to store the users, their roles and profiles/settings.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user retrieves the list of users, their roles and profiles/settings.
Test prerequisites/ configuration	• The SP should be operational.
Success criteria	The user is able to retrieve and view the list of users, their profiles and the roles they are assigned to.
Results analysis	*The test has been performed and passed according to the success criteria.*
Success	PASSED

Table 6-13: TS_UeRM_12: Scenario deletion.

Test ID	*TS_UeRM_12*
Requirement to be verified	• *TR_UeRM_13* ○ *Sys_IntUeMan_8*
Test objective	Allow only incident commanders to delete scenarios.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user logs in to the HEIMDALL platform through the GUI, with valid credentials associated with the role of incident commander.

	3. The user retrieves the list of scenarios. 4. The user deletes the scenario he/she wants.
Test prerequisites/ configuration	• The GUI should be operational. • The SP should be operational. • The scenario management module should be operational.
Success criteria	The user is able to delete scenarios. Only users with the role of incident commander are able to perform that action.
Results analysis	*N/A*
Success	NOT_PERFORMED

Table 6-14: TS_UeRM_13: Deletion of scenario and lessons learnt templates.

Test ID	*TS_UeRM_13*

Requirement to be verified	• *TR_UeRM_14*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_10*

Test objective	Allow only authorised users to delete templates of scenarios and lessons learnt.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user logs in to the HEIMDALL platform through the GUI, with valid credentials associated with the appropriate roles(s) 3. The user retrieves the list of scenarios or lessons learnt templates. 4. The user deletes the templates he/she wants.
Test prerequisites/ configuration	• The GUI should be operational. • The SP should be operational. • The scenario management module should be operational
Success criteria	The user is able to delete the corresponding templates. Only users with the appropriate role(s) are able to perform that action.
Results analysis	*N/A*
Success	NOT_PERFORMED

Table 6-15: TS_UeRM_14: Modification of scenario information.

Test ID	*TS_UeRM_14*

Requirement to be verified	• *TR_UeRM_15*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_11*

Test objective	Allow only authorised users to modify scenario information.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user logins to the HEIMDALL platform through the GUI, with valid credentials associated with the appropriate roles(s) 3. The user retrieves the list of scenarios. 4. The user modifies the scenario information that he/she wants.
Test prerequisites/	• The GUI should be operational.

configuration	• The SP should be operational. • The scenario management module should be operational
Success criteria	The user is able to modify the scenario. Only users with the appropriate role(s) are able to perform that action.
Results analysis	*N/A*
Success	NOT_PERFORMED

Success	NOT_PERFORMED

Table 6-16: TS_UeRM_15: Modification of map symbology.

Test ID	*TS_UeRM_15*

Requirement to be verified	• *TR_UeRM_16*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_12*

Test objective	Allow only authorised users to modify map symbology.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user logs in to the HEIMDALL platform through the GUI, with valid credentials associated with the appropriate roles(s) 3. The user modifies the map symbology.
Test prerequisites/ configuration	• The GUI should be operational. • The SP should be operational. • The scenario management module should be operational
Success criteria	The user is able to modify the map symbology. Only users with the appropriate role(s) are able to perform that action.
Results analysis	*N/A*
Success	NOT_PERFORMED

Table 6-17: TS_UeRM_16: Modification of map symbology.

Test ID	*TS_UeRM_16*

Requirement to be verified	• *TR_UeRM_17*
	  ○ *Sys_IntUeMan_2*
	  ○ *Sys_IntUeMan_13*

Test objective	Allow only authorised users to create map layers
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user logins to the HEIMDALL platform through the GUI, with valid credentials associated with the appropriate roles(s) 3. The user creates map layers.
Test prerequisites/ configuration	• The GUI should be operational. • The SP should be operational. • The scenario management module should be operational
Success criteria	The user is able to create map layers. Only users with the appropriate role(s) are able to perform that action.
Results analysis	*N/A*

Success	NOT_PERFORMED

Table 6-18: TS_UeRM_17: Modification of map symbology.

Test ID	*TS_UeRM_17*
Requirement to be verified	• *TR_UeRM_18* ○ *Sys_IntUeMan_16*
Test objective	Allow only authorised users to create and send alert messages through the information gateway.
Test procedure	1. The user connects to the HEIMDALL VPN. 2. The user logs in to the HEIMDALL platform through the GUI, with valid credentials associated with the appropriate roles(s) 3. The user creates an alert message. 4. The user dispatches the alert message through the information gateway to the intended recipients.
Test prerequisites/ configuration	• The GUI should be operational. • The SP should be operational. • The IG should be operational.
Success criteria	The user is able to create and dispatch alert messages. Only users with the appropriate role(s) are able to perform that action.
Results analysis	*N/A*
Success	NOT_PERFORMED

6.2 Test Summary

The matrix in Table 6-19 summarizes the test coverage of technical requirements.

Table 6-19: Test coverage matrix

Requirement ID	Test ID	Result
TR_UeRM_01	*TS_UeRM_03*	PASSED
TR_UeRM_02	*TS_UeRM_04* *TS_UeRM_05*	PASSED PASSED
TR_UeRM_03	*TS_UeRM_05*	PASSED
TR_UeRM_04	*TS_UeRM_06*	PASSED
TR_UeRM_05	*TS_UeRM_07*	PARTIAL
TR_UeRM_06	*TS_UeRM_03* *TS_UeRM_04* *TS_UeRM_05* *TS_UeRM_08*	PASSED PASSED PASSED PASSED
TR_UeRM_07	*TS_UeRM_03* *TS_UeRM_09*	PASSED PARTIAL
TR_UeRM_08	*TS_UeRM_10*	NOT_PERFORMED
TR_UeRM_09	*TS_UeRM_01*	PASSED
TR_UeRM_10	*TS_UeRM_07* *TS_UeRM_08*	PARTIAL PASSED

	TS_UeRM_11	PASSED
TR_UeRM_11	*TS_UeRM_01* *TS_UeRM_02*	NOT_PERFORMED NOT_PERFORMED
TR_UeRM_12	*TS_UeRM_01*	PASSED
TR_UeRM_13	*TS_UeRM_12*	NOT_PERFORMED
TR_UeRM_14	*TS_UeRM_13*	NOT_PERFORMED
TR_UeRM_15	*TS_UeRM_14*	NOT_PERFORMED
TR_UeRM_16	*TS_UeRM_03* *TS_UeRM_15*	PASSED NOT_PERFORMED
TR_UeRM_17	*TS_UeRM_16*	NOT_PERFORMED
TR_UeRM_18	*TS_UeRM_17*	NOT_PERFORMED
TR_UeRM_19	N/A	N/A

7 Conclusion

This report presented the implementation status of the UeRM of HEIMDALL. The implemented component has followed the user and system requirements. The first technical activity was the generation of the technical specifications and the creation of the first prototype of the UeRM, to facilitate the implementation and integration of the HEIMDALL system.

An early version of the HEIMDALL UeRM has been tested in lab trials towards the Release A and the EUW2. During these phases, it showed adequate stability and scalability. Additional tests have been performed as the user and system requirements were maturing and the component design and implementation was evolving in order to meet the corresponding requirements.

At the moment the UeRM is being actively developed in order to cover the additional technical requirements, undergo lab testing and validation and participate in the integrated prototype of the Release B and EUW3. The UeRM is operational and available for the following phases of development and integration.

8 References

[1] Barth, B., et al. (2019). HEIMDALL D2.7: HEIMDALL Requirements Report – Issue 2

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

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

[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. (2018) HEIMDALL D4.13: Communications and Information Sharing – Specifications

[7] Mathew, D. et al. (2018) HEIMDALL D4.16: Communications to Remote Areas – Design and Specifications – Draft

[8] Friedemann, M. et al. (2018) HEIMDALL D5.1: EO Tools and Products – Specifications – Draft

[9] Barth, B. et al. (2018) HEIMDALL D5.4: 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] To be released on M22 (2019) HEIMDALL D5.9: Interfaces for External and Existing Systems – Specifications – Draft

[13] To be released on M22 (2019) HEIMDALL D5.12: Modelling and Simulation Services – Specifications – Draft

[14] Friedemann, M. et al. (2018) HEIMDALL D6.1: Concept design for risk analysis methods and components – Detailed concept design and documentation of methods on risk analysis

[15] Mendes, M. et al. (2018) HEIMDALL D6.4: Technical Specifications on Hazard, Scale and User-Specific Risk Assessment Information, Products and Service Workflows

[16] Friedemann, M. et al. (2018) HEIMDALL D6.7: Situation Assessment, Impact Summary Generation and sCOP/SITREP Specification and Implementation Report – Draft

[17] Friedemann, M. et al. (2018) HEIMDALL D6.10: Decision Support Specification and Implementation Report - Draft

[18] Friedemann, M. et al. (2018) HEIMDALL D6.14: Scenario Specification, Scenario Management Specification and Scenario and Situation Metrics – Draft