



## D4.17

# Communications to Remote Areas – Design and Specifications – FINAL

<b>Instrument</b>	Collaborative Project
<b>Call / Topic</b>	H2020-SEC-2016-2017/H2020-SEC-2016-2017-1
<b>Project Title</b>	Multi-Hazard Cooperative Management Tool for Data Exchange, Response Planning and Scenario Building
<b>Project Number</b>	740689
<b>Project Acronym</b>	HEIMDALL
<b>Project Start Date</b>	01/05/2017
<b>Project Duration</b>	42 months
<b>Contributing WP</b>	WP 4
<b>Dissemination Level</b>	PU
<b>Contractual Delivery Date</b>	M45
<b>Actual Delivery Date</b>	14/01/2021
<b>Editor</b>	Joseph Muna (AVA)
<b>Contributors</b>	Diana Mathew, Joseph Muna (AVA), Benjamin Barth, Govinda Kabbinahithilu (DLR)

<b>Document History</b>			
Version	Date	Modifications	Source
0.1	11/01/2021	ToC and first draft	AVA
0.2	13/01/2021	Draft for QA to DLR	AVA
1.0.F	14/01/2021	Final version approved for submission	DLR

# Table of Contents

- List of Figures..... iii
- List of Tables..... iv
- List of Acronyms..... v
- Executive Summary ..... 8
- 1 Introduction ..... 9
- 2 Technical Requirements.....10
  - 2.1 Information Gateway .....10
    - 2.1.1 Interface Requirements .....10
    - 2.1.2 Functional Requirements .....11
    - 2.1.3 Non-functional Requirements .....16
  - 2.2 Satellite Communication.....17
    - 2.2.1 Interface Requirements .....17
    - 2.2.2 Functional Requirements .....17
    - 2.2.3 Non-Functional Requirements.....18
- 3 Reference Architecture.....19
  - 3.1 Overall HEIMDALL architecture.....19
  - 3.2 Information Gateway .....19
  - 3.3 Satellite Communication.....20
- 4 Module Functionality .....21
  - 4.1 Information Gateway .....21
  - 4.2 Satellite Communication.....22
- 5 Technical Specification.....27
  - 5.1 Alerting service.....27
    - 5.1.1 Database.....27
    - 5.1.2 Alerting Services Web Services Specification.....28
  - 5.2 First Responders Information Web Services Specification.....34
    - 5.2.1 POST methods.....34
    - 5.2.2 GET methods.....35
    - 5.2.3 DELETE method .....38
- 6 Test Plan.....39
  - 6.1 IG Verification.....39
- 7 Conclusion .....43
- 8 References.....44

# List of Figures

Figure 3-1: HEIMDALL Architecture .....19

Figure 4-1: Information Gateway (IG) components .....21

Figure 4-2: Schematic architecture diagram of the end-end communication link in HEIMDALL with rapid deploy satellite communication.....23

Figure 4-3: Service Parameters from Avant VMAP Tool .....25

# List of Tables

- Table 2-1: Technical Requirement TR\_Com\_1 .....11
- Table 2-2: Technical Requirement TR\_Com\_2.....11
- Table 2-3: Technical Requirement TR\_Com\_3.....11
- Table 2-4: Technical Requirement TR\_Com\_14.....12
- Table 2-5: Technical Requirement TR\_Com\_15.....12
- Table 2-6: Technical Requirement TR\_Com\_16.....12
- Table 2-7: Technical Requirement TR\_Com\_21.....13
- Table 2-8: Technical Requirement TR\_Com\_5.....13
- Table 2-9: Technical Requirement TR\_Com\_20.....14
- Table 2-10: Technical Requirement TR\_Com\_22.....14
- Table 2-11: Technical Requirement TR\_Com\_23.....14
- Table 2-12: Technical Requirement TR\_Com\_17.....15
- Table 2-13: Technical Requirement TR\_Com\_18.....15
- Table 2-14: Technical Requirement TR\_Com\_19.....16
- Table 2-15: Technical Requirement TR\_Com\_11.....16
- Table 2-16: Technical Requirement TR\_Com\_12.....16
- Table 2-17: Technical Requirement TR\_Com\_13.....17
- Table 3-1: Information gateway inputs/outputs .....20
- Table 3-2: Interface to the Service Platform (SP).....20
- Table 3-3: Communication to Remote areas inputs/outputs.....20
- Table 4-1: Component Description and Exemplars.....23
- Table 4-2: Satellite Service Parameters .....25
- Table 4-3: Key Service Installation Stakeholders Roles and Descriptions.....25
- Table 4-4: Service Test Summary .....26
- Table 5-1: IG database tables .....27
- Table 5-2: HTTP POST commands of the IG web services .....28
- Table 5-3: IG POST body values for the set command.....29
- Table 5-4: IG POST body values for the add file command .....32
- Table 5-5: IG POST body values for the dispatch command .....32
- Table 5-6 IG POST body values for the reset command.....32
- Table 5-7: HTTP GET commands of the IG web services .....33
- Table 5-8: HTTP POST commands of the Information web services .....34

## List of Acronyms

ACAP	Alerting Channels Access Points
AG	Alerting Gateway
AOI	Area of Interest
AVA	Avanti Communications LTD
BUC	Block Up Converter
CAP	Common Alerting Protocol
CPE	Customer Premises Equipment
CTTC	Centre Tecnològic de Telecomunicacions de Catalunya (Catalan Technological Telecommunications Centre)
C&C	Command & Control
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center)
EC	European Commission
EUW	End Users Workshop
FCP	Forward Command Post
FWD	Forward
FR	First Responder
GHz	Gigahertz
GUI	Graphical User Interface
HTS	High Throughput Satellite
HYLAS	Highly Adaptable Satellite
IG	Information Gateway
IPR	Intellectual Property Right
IV&V	Integration, Verification & Validation
JSON	JavaScript Object Notation
LNB	Low Noise Block
LU	Local Unit
MCE	Message Composition Engine
Mbps	Megabits per second
NOC	Network Operations Centre
OSS	Operations Support System

RTN	Return
SatCom	Satellite Communication
SitRep	Situation Report
SOC	Satellite Operations Centre
SP	Service Platform
SR	System Requirement
TR	Technical Requirement
UDP	User Datagram Protocol
UR	User Requirement
URL	Uniform Resource Locator
VSAT	Very Small Aperture Terminal
Wi-Fi	Wireless Fidelity

**Intentionally blank**



## Executive Summary

This deliverable presents the work carried out as part of Task 4.5 (Communication to Remote Areas) of the HEIMDALL project. The main objective of this document is to provide details about the design requirements and specifications of the Information Gateway (IG) and SatCom modules of the HEIMDALL system, which together make the 'communication to remote area' functionality possible. This deliverable thus includes the technical requirements derived from the user and system requirements identified in D2.9 [4] and D2.10 [5]. More importantly, this document is the final version of D4.16, which has been here completed with a few details related to the testing methodology. The present deliverable has been also revised with respect to the architecture and module description in order to provide a comprehensive and clear description of the satellite subsystem object of Task 4.5. Finally, specific details on how the system has been implemented in terms of equipment and overall end-to-end integration is provided in the companion deliverable D.4.18 "Communication to remote areas (demonstrator).

The Information Gateway (IG) module facilitates communication with the population and information sharing between the first responder (FR) users and the web-based HEIMDALL users. The IG, via the smartphone application, allows the FRs receive information available on the HEIMDALL system and messages from the command and control centre (C&C). Additionally, the IG offers a multi-language capable alerting service for the public. The IG is an augmented version of the alerting gateway (AG) messaging service developed as part of the Alert4All [1] and PHAROS [2] projects. On one end, the IG is connected to the rest of the modules in the HEIMDALL network via the service platform (SP) and on the other end it connects to the HEIMDALL mobile application via internet.

As hazard events usually take place in remote locations with no/limited internet connectivity or result in disrupting the communication infrastructure of the location, a satellite-based broadband connection the most viable technology in such circumstances. A portable, easily deployable, Ka-band based Very Small Aperture Terminal (VSAT) kit with Wireless Fidelity (Wi-Fi) capability is used to provision an on-demand high-throughput satellite-based broadband connection for use by the first responders at the disaster site.

In terms of IG development, as main differences to the previous issue of this deliverable, the first responder's information service has been specified, implemented and evaluated. For the alerting service a feature that alerts can be sent to another HEIMDALL instance was integrated into the system.

A mobile application has been developed for HEIMDALL as part of Task 5.3 (Crowdsourced and First Responders data).

# 1 Introduction

To address the issue of providing a reliable communication link to first responders in the field in remote areas with no/limited service, HEIMDALL is proposing a satellite-based connection. Communication to remote areas or areas experiencing connectivity issues during an event is envisaged to be served by the Information Gateway (IG).

To this end, the document discusses on the identified technical requirements taken as baseline for the design and development of the satellite subsystem as well as its integration with rest of the HEIMDALL system. As part of the overall system engineering methodology, testing is also provided and discussed, in order to show the successful implementation of the proposed system with respect to the identified technical requirements.

This document is then organised as follows:

- Section 2 describes the technical specifications captured so far for the IG module, the satellite-based broadband connection and the smartphone application, along with their interface requirements.
- Section 3 describes the context of the IG module, the Satellite Communication (SatCom) connection and the mobile-app within the overall HEIMDALL architecture
- Section 4 provides a brief description of the functionalities offered by the IG and the mobile application at the end of Release-A phase.
- Section 5 provides the IV&V test procedures and results of the IG and the application for the Release- A phase.
- Finally, Section 6 summarises and concludes the document. References are provided in Section 7.

## 2 Technical Requirements

This section describes the technical requirements (TRs) for the information gateway (IG) and the satellite communication (SatCom) modules of the HEIMDALL platform. The TRs have been derived from the user requirements (URs) and system requirements (SRs) identified during the end user workshops (EUWs) and progress meetings in the initial 18 months of the project. These URs and SRs have been documented in D2.9 [4] and D2.10 [5].

Technical requirements for the smartphone application for first responders are also included here in the interim to make it easier to keep track of the requirements definition of the functionalities in the initial implementation and integration stages. These requirements will be later moved to a standalone document D5.7 [6] due M38.

### 2.1 Information Gateway

The alerting gateway (AG) developed as part of the Alert4All [1] and PHAROS [2] projects to improve population awareness will be used to send information messages to first responders. HEIMDALL will see augmentation of the AG into a secure information gateway to grant access for first responders to the information available on the platform.

#### 2.1.1 Interface Requirements

The Information Gateway (IG) is connected on one hand to the HEIMDALL network via the Service Platform (SP) and on the other hand to the Internet for providing the information to the first responders in the field and to model a channel to the public. Via the Internet connection the Satellite Network can be accessed for information exchange. For verification of the interface requirements, it shall be tested whether or not the specified interface features of the IG are working properly.

##### 2.1.1.1 Hardware Interfaces

The IG should be connected to the internet and HEIMDALL network via Ethernet port.

##### 2.1.1.2 Software Interfaces

The IG shall interface via RESTful Web services to:

- The SP to access the user and scenario management as well as the situation awareness tools;
- The receiver application;
- The Graphical User Interface (GUI) to let a user control the IG.

These requirements relate to: Sys\_DSC\_12, Sys\_DSC\_10, Sys\_DSC\_11, Sys\_DSC\_5, Sys\_DSC\_1, Sys\_DSC\_2.

##### 2.1.1.3 Communication Interfaces

The IG shall use either HTTP or, for secured connection, HTTPS to connect to the HEIMDALL network and the internet.

These requirements relate to: Sys\_DSC\_26.

## 2.1.2 Functional Requirements

### 2.1.2.1 Short Term Features

Table 2-1: Technical Requirement TR\_Com\_1

Requirement ID:	TR_Com_1
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_8</li> </ul>
<b>Description:</b>	
The IG shall offer means to keep the population informed.	
Rational: In order to mitigate people at risk and casualties the population must be informed. The IG is the technical module that shall include this functionality.	
Stimulus: The user starts a process to inform the public.	
Response: Information is distributed via dedicated channels.	
Verification Criterion: Information can be distributed using the IG	
Notes: Standards should be used.	

Table 2-2: Technical Requirement TR\_Com\_2

Requirement ID:	TR_Com_2
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_5</li> <li>• Sys_DSC_8</li> </ul>
<b>Description:</b>	
The IG shall be able to transmit alert messages to the population.	
Rational: The population shall be informed about risks, hazards and disasters. Alert messages are common and successful mean to inform the public.	
Stimulus: The user creates an alert message at the IG using the GUI and triggers the transmission of it.	
Response: IG dispatches an alert to the connected networks.	
Verification Criterion: An alert is created at the IG and dispatched. The alert is successfully received and displayed at the receiver application.	
Notes: none	

Table 2-3: Technical Requirement TR\_Com\_3

Requirement ID:	TR_Com_3
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_5</li> <li>• Sys_DSC_8</li> </ul>
<b>Description:</b>	
The IG shall be able to create alert messages.	

Rational: In order to transmit messages to the public they need to be created.
Stimulus: The user uses to GUI to create an alert message at the IG.
Response: The IG informs the user about the status of the message under creation.
Verification Criterion: Using the GUI an alert message can be create at the IG.
Notes: Standards should be used.

Table 2-4: Technical Requirement TR\_Com\_14

Requirement ID:	TR_Com_14
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_31</li> </ul>
<b>Description:</b>	
The IG shall be able to consider predefined areas on different levels (municipality/region/state).	
Rational: Different user profiles look at different scales of the area. The IG must be able to cope with all area sizes.	
Stimulus: GUI sends the selected area.	
Response: The area is updated by the IG.	
Verification Criterion: The area shown in the alert message is the one selected by the user.	
Notes: None	

Table 2-5: Technical Requirement TR\_Com\_15

Requirement ID:	TR_Com_15
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_31</li> </ul>
<b>Description:</b>	
The IG shall be able to consider different types of areas:	
<ul style="list-style-type: none"> <li>• Circles</li> <li>• Polygons</li> <li>• Predefined areas</li> <li>• Address</li> </ul>	
Rational: Offering multiple options to set the area is very flexible and allows the user to select the best possible way for the current situation.	
Stimulus: GUI sends the selected area.	
Response: The area is updated by the IG.	
Verification Criterion: The area shown in the alert message is the one selected by the user.	
Notes: None	

Table 2-6: Technical Requirement TR\_Com\_16

Requirement ID:	TR_Com_16
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_33</li> <li>• Sys_DSC_32</li> </ul>
<b>Description:</b>	
The IG shall be able to consider different types of users as recipients within the private message.	
Rational: The user wants to address sub-groups within the organisation.	
Stimulus: User selects a sub-group for which the message is intended.	
Response: The message is send to the corresponding addresses of the sub-groups.	
Verification Criterion: If a sub-group is selected as recipient, the IG sends the addresses of the sub-group to the address field of the Common Alerting Protocol (CAP) message.	
Notes: Address field is used instead of sub-categories in the CAP standard.	

Table 2-7: Technical Requirement TR\_Com\_21

Requirement ID:	TR_Com_21
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_42</li> </ul>
<b>Description:</b>	
The IG shall be able to share pictures.	
Rational: Maps are provided in digital format as pictures.	
Stimulus: A picture is set as an attachment to an alert message.	
Response: The IG attaches the picture to the alert message.	
Verification Criterion: The user in the field can access the data available at the local unit (LU).	
Notes: none	

### 2.1.2.2 Mid-Term Features

Table 2-8: Technical Requirement TR\_Com\_5

Requirement ID:	TR_Com_5
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_16</li> </ul>
<b>Description:</b>	
The IG shall consider access rights for each user/profile.	
Rational: In order to design a secure system the IG must be able to interact with the access control methods of the system to ensure that only authorized personal can access the system via the IG.	
Stimulus: Credentials for the access to the system are transferred for login.	
Response: The system grants access upon successful login.	

Verification Criterion: User with an account and the corresponding rights can access information available for their profile. Attackers with invalid credentials cannot access the system via the IG.

Notes: The IG is relying on the user rights provided by the overall HEIMDALL system here. No specific measures are taken at the IG.

Table 2-9: Technical Requirement TR\_Com\_20

Requirement ID:	TR_Com_20
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_36</li> <li>• Sys_DSC_18</li> </ul>
<b>Description:</b>	
The IG shall be able to share data and data sets with users in the field.	
Rational: In order to share the information with users in the field the IG is the interface.	
Stimulus: Data is requested by the user in the field.	
Response: The IG forwards the information tailored to the channel quality.	
Verification Criterion: The user in the field can access the data available at the LU.	
Notes: none	

Table 2-10: Technical Requirement TR\_Com\_22

Requirement ID:	TR_Com_22
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_27</li> </ul>
<b>Description:</b>	
The IG shall use standard formats where possible	
Rational: To be able to share the information with the population fostering social inclusion and considering people with special needs standard receivers that consider special needs can be used.	
Stimulus: Content is created at the IG	
Response: Content is provided in standard compliant format	
Verification Criterion: standard receivers/validators can consume IG generated data.	
Notes: none	

Table 2-11: Technical Requirement TR\_Com\_23

Requirement ID:	TR_Com_23
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_35</li> </ul>
<b>Description:</b>	
The IG shall ask a confirmation before dispatching the alert.	

Rational: in order to request a confirmation by a superior (communication chief) the IG needs to ask for a confirmation.
Stimulus: Alert is created and prepared to send
Response: Confirmation is requested from IG before dispatching
Verification Criterion: Confirmation is requested
Notes: none

### 2.1.2.3 Long-term requirements

Table 2-12: Technical Requirement TR\_Com\_17

Requirement ID:	TR_Com_17
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_34</li> </ul>
<b>Description:</b>	
The IG shall be able to allow the selection among multiple channels.	
Rational: In order to allow the user the selection of dedicated channels the IG must consider this.	
Stimulus: User selects/deselects a channel for alerting.	
Response: The IG updates the configuration for the dissemination.	
Verification Criterion: Different channels are selected/ deselected and the IG updates the config files.	
Notes: IG is already capable of this and was evaluated in previous projects on this aspect. Its is out of scope for HEIMDALL	

Table 2-13: Technical Requirement TR\_Com\_18

Requirement ID:	TR_Com_18
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_36</li> </ul>
<b>Description:</b>	
The IG shall be able to provide information about who has access to which data.	
Rational: In order to allow the user to have a good picture of the situation it is necessary that he knows who has access to the data.	
Stimulus: User requests an overview of a data set about who has the access rights.	
Response: The IG answers with the access rights.	
Verification Criterion: The correct access rights for the data set are shown.	
Notes: none	



Table 2-14: Technical Requirement TR\_Com\_19

Requirement ID:	TR_Com_19
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_36</li> </ul>
<b>Description:</b>	
The IG shall be able to show the incident commander who has accessed the data already	
Rational: In this way the incident commander can have an overview if there is a lack of information at some point.	
Stimulus: The incident commander asks about the status of distribution for some information.	
Response: System returns if someone with access rights already has accessed the data.	
Verification Criterion: The incident commander can identify if a data set has been accessed by another user.	
Notes: none	

### 2.1.3 Non-functional Requirements

Table 2-15: Technical Requirement TR\_Com\_11

Requirement ID:	TR_Com_11
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_6</li> </ul>
<b>Description:</b>	
The IG shall be able to connect to communication networks.	
Rational: In order to forward messages and connect first responders the IG must be able to connect to communication networks.	
Verification Criterion: The IG can establish a connection to other system components according to the overall system architecture.	
Notes: Standards should be used.	

Table 2-16: Technical Requirement TR\_Com\_12

Requirement ID:	TR_Com_12
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_26</li> </ul>
<b>Description:</b>	
The IG shall be connected to secure communications links.	
Rational: The links to and from the IG must be secured to ensure security of the whole system.	
Verification Criterion: An attacker outside of the secured network tries to access the IG which shall not be successful.	
Notes: Standards should be used.	

## 2.2 Satellite Communication

Natural disasters like wildfires, floods, landslides and so on, usually occur in remote areas and additionally often result in destroying the existing communication infrastructure of the area. These factors present difficulties such as low quality/ congested/ non-existent communication channels for emergency response operation teams. HEIMDALL proposes to overcome this situation by using satellite-based internet connectivity. The proposed solution will make use of a rapidly deployable, lightweight and portable VSAT operating in Ka-Band. In addition, a Wi-Fi access point will be set-up, allowing all authorised first responders to communicate with the Forward Command Post (FCP) and the C&C via the HEIMDALL web-platform and/or the mobile application. The satellite broadband connection will also allow real-time monitoring and communication with the drones<sup>1</sup> system by means of Wi-Fi connectivity. The VSAT employed will be a Sematron Flyaway terminal consisting of a VSAT antenna, modem, Wi-Fi router and battery. Specifications of the VSAT kit will be detailed in the final issue of this document.

### 2.2.1 Interface Requirements

#### 2.2.1.1 Hardware Interface

The VSAT is a standalone unit. The smartphones will connect to the portable VSAT terminal via an integrated Wi-Fi router.

#### 2.2.1.2 Software Interface

Not applicable.

#### 2.2.1.3 Communication Interfaces

Any standard IP data protocol such as HTTP or HTTPS can be used to connect to remote servers.

### 2.2.2 Functional Requirements

#### 2.2.2.1 Mid-term Features

Table 2-17: Technical Requirement TR\_Com\_13

Requirement ID:	TR_Com_13
Related SR(s):	<ul style="list-style-type: none"> <li>• Sys_DSC_6</li> <li>• Sys_DSC_14</li> <li>• Sys_DSC_26</li> <li>• Sys_DSC_30</li> </ul>
<b>Description:</b>	
An alternative secure communication system that is easily deployable by the user in a disaster area shall be possible. Ka-band satellite broadband connectivity shall be made possible using a rapidly deployable, lightweight and portable Ka-band VSAT.	

<sup>1</sup> More technical details about the reference drone system used throughout the course of the project can be found in D5.4 "Drone Sensors and In-Situ Sensors"

Rational: The location of some incidents is sometimes out of range of terrestrial communications networks and/or communication networks can be damaged in the aftermath of an event. Therefore, the system must provide an alternative way of communications.
Stimulus: An incident takes place in an area with no connectivity via terrestrial communications networks.
Response: A Ka-band satellite terminal shall be deployed providing an alternative communication link to the first responders and the FCP.
Verification Criterion: A Ka-band satellite terminal is installed in the incident area. The FR/FCP is able to connect to the satellite broadband network using a mobile phone/laptop and check that they can reach the internet.
Notes: none

### 2.2.3 Non-Functional Requirements

No non-functional requirements have yet been identified as part of the requirements definition exercise.

## 3 Reference Architecture

### 3.1 Overall HEIMDALL architecture

Figure 3-1 shows the overall architecture of the HEIMDALL system. The Intelligent Gateway (IG) module, SatCom module and the Receiver App are highlighted in red boxes.

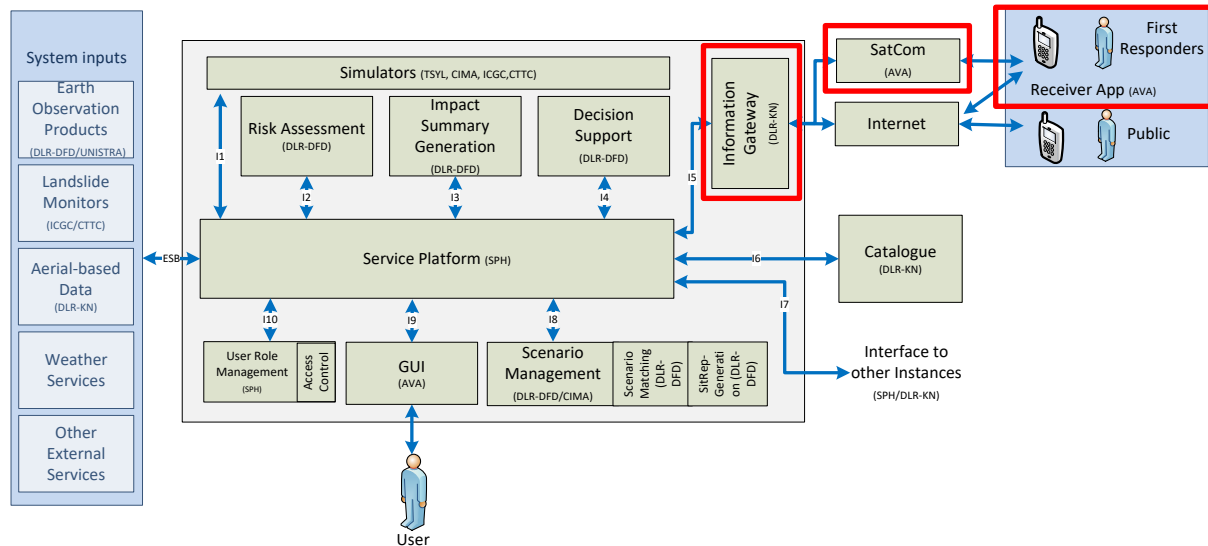


Figure 3-1: HEIMDALL Architecture

### 3.2 Information Gateway

As depicted in Figure 3-1, the central element between the HEIMDALL system and the first responders (FRs) in the field and the public is the information gateway (IG). It is connected to the HEIMDALL platform via the service platform (SP) and receives input from the scenario management module and the GUI. It offers an information services to the users in the field (as summarized in Table 3-1) by forwarding a situation report (SitRep). The report is generated at the scenario management module [7] and includes the necessary information for the FRs. The EDXL-SitRep format [8] is used for generating the report. Before transmitting EDXL-Distribution Element (DE) [9] information (e.g. classification, receiver area) is added that the IG receives from the GUI. EDXL-DE can be seen as kind of envelope with the necessary information for distributing the content and finding the right recipient; the content in this case is the EDXL-SitRep. The HEIMDALL smartphone application [10] acts as a receiver for the information.

The second service offered by the IG is the alerting service for FR and public awareness. Alert messages are composed by the user using the web-based GUI and can be forwarded to the smartphone application that can also be used as alerts receiver. The alerting service is based on the approaches of the PHAROS and Alert4All projects and offers multi-language capabilities and easy message composition features and enables the possibility to use a variety of communication channels as shown during these projects. For HEIMDALL we focus on terrestrial internet access and satellite-based broadband, only.

The HEIMDALL system has been extended so that alerts can directly be received by another local unit. The system recognizes if a LU ID is selected as address and forwards it to the dedicated SP. The SP notifies the GUI which shows an notification to the user about a new alert and uploads it to the IG of the dedicated LU.

Table 3-1: Information gateway inputs/outputs

<b>Products and/or Services</b>	<b>Inputs needed</b>	<b>Provided by</b>	<b>Used by</b>
FR information service	Situation Report (SitRep) Scenario User Role Management (UeRM)	Situation Assessment Scenario Management	Smartphone application
Alerting service	CAP message	GUI	Smartphone application

Table 3-2: Interface to the Service Platform (SP)

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

### **3.3 Satellite Communication**

As shown in the overall architecture diagram, the satellite communication module forms a data link between the core HEIMDALL platform and the HEIMDALL mobile application, facilitating data transfer in the form of voice, video and messaging. The SatCom module is a standalone entity with no hardware interface to any of the other HEIMDALL modules.

Table 3-3: Communication to Remote areas inputs/outputs

<b>Products and/or Services</b>	<b>Inputs needed</b>	<b>Provided by</b>	<b>Used by</b>
Ka-band satellite communications	Successful configuration and installation	N/A	First Responders

## 4 Module Functionality

This section shall describe the different building blocks within the IG and SatCom modules.

### 4.1 Information Gateway

The Information Gateway (IG) design has been based on the Alerting Gateway (AG) which has been developed in the framework of the EU FP7 projects Alert4All [1] and PHAROS [2] and therein represented the communication hub responsible for dispatching alert messages composed by the operator via the web-based GUI. It is further extended in HEIMDALL in order to process and forward situation awareness information and scenarios for FRs in the field.

The FR information service for situation awareness has been designed together with the end user partners of HEIMDALL. Important parameters for end users in the field of the HEIMDALL scenario model have been identified by the end users and are included in an EDXL-SitRep. The SitRep is transmitted via EDXL-DE with the standardized mandatory information to the app which shows only the important data set to the responders. This approach should prevent an overload if information of user in the field which are in a stressful situation anyway.

Components of the IG are shown in Figure 4-1. The FR information service is included in the web server of the alerting service with the difference that it uses the same server for forwarding the information as well. It has access to the local database and configuration files. The alerting service makes use of some more components, inherited from previous designs and explained in detail in the following. This includes the functions that implement the interface with the SP, the AG main processor; the message composition engine (MCE), the interface with the alerting channels access points (ACAPs), the local database and the configuration files.

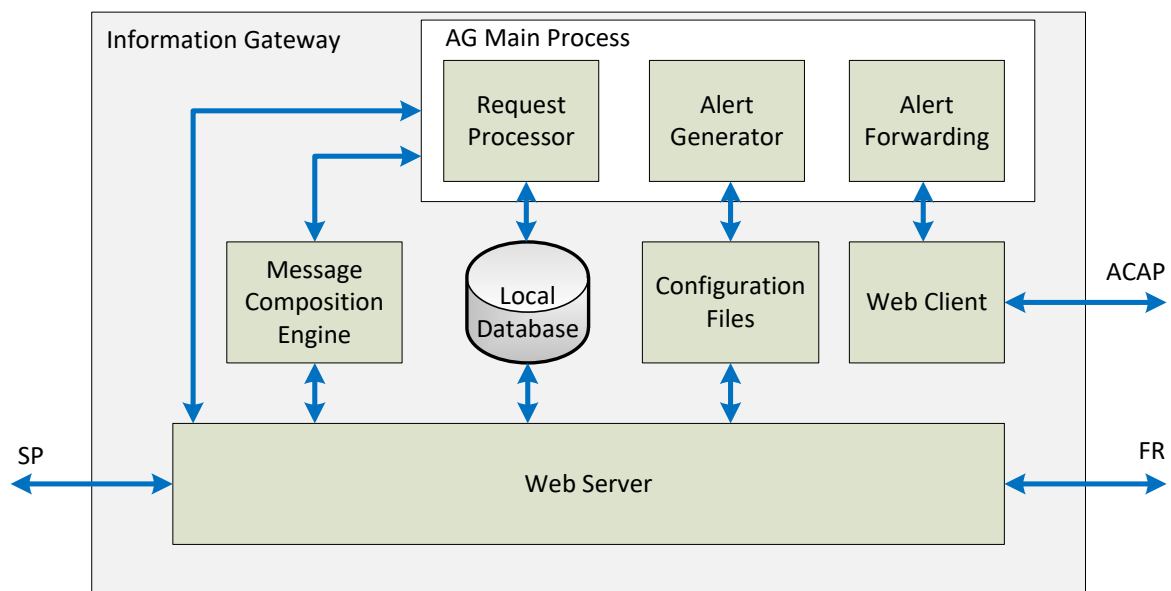


Figure 4-1: Information Gateway (IG) components

The interface between the IG and the SP is based on web services and, hence, it consists of web server, able to receive and process the corresponding web requests. The web server implements the following functions for the alerting service:

- Receive and process the web requests from the SP and send responses;
- Generate common alerting protocol (CAP) [3] files out of the SP inputs for internal use;
- Trigger the MCE;

- Update the local database with the requests for documentation purposes;
- Trigger the process in the 'AG Main Process' module that correspond to the received requests;

The web server provides the web services in a PHP script described in more detail in section 5.1.2. As already mentioned, the functionality of the web server for the FR information service is extended since it will also receive requests to and from FRs as well as process situation awareness and scenario information.

In order to create a human readable version of the alert message to be distributed, the MCE is called by the web server functions which has been developed and enhanced in PHAROS and Alert4All. The MCE is provided as a Java executable. The IG uses the MCE to set the text in the description, the instruction and the headline fields of the CAP, using the information in the other CAP fields to formulate a human-readable version of the alert message.

The *AG main process* module in the IG is structured in three main parts:

- The request processor is the module which communicates with the web server in the SP interface and triggers the corresponding internal processes that can either be the alert generator or the alert forwarding;
- The alert generator takes care of generating the alert message in the appropriate format to be transmitted and decoded;
- Finally, the alert forwarding module takes care of intelligent forwarding of alert messages (and report requests) towards the ACAPs.

The *AG main process* and the interface to the ACAPs are combined in a multi-thread C++ program. The main process is triggered by the web server at the SP interface. If a dispatch command is initiated by the user at the GUI, the web server sends the corresponding CAP file and the list of the selected channels via User Datagram Protocol (UDP) socket to the main process. It is received by the request processor, which parses the CAP file and sets the channels accordingly at the alert forwarding entity. The alert generator converts the parsed CAP to different formats. It is possible to convert it to the A4A-protocol and JavaScript Object Notation (JSON) (see section 5.1.2).

## **4.2 Satellite Communication**

The satellite-based broadband connectivity for the users on the field will be provided using a light weight, robust, quick-deployable Sematron Flyaway terminal. A high level schematic of the end-end connection facilitated by the satellite terminal is shown in Figure 4-2.

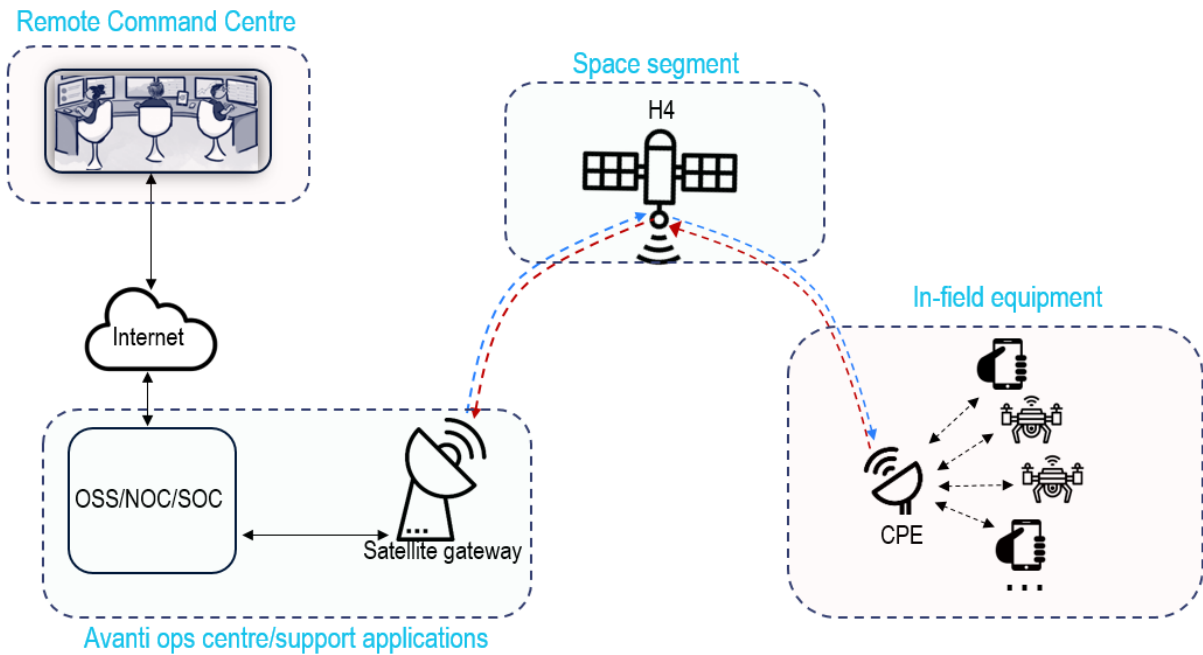




Figure 4-2: Schematic architecture diagram of the end-end communication link in HEIMDALL with rapid deploy satellite communication

Table 4-1 below summarises the equipment that needs to be installed and configured enable the service.

Table 4-1: Component Description and Exemplars

Segment	Equipment	Type	Description	Exemplar
In-field equipment	Portable terminal	VSAT	Sematron Lightweight portable all-in one ruggedized fly away terminal (router, dish, LNB, Wi-Fi access point)	
	Drones	Any	Wi-Fi enabled drone	
	Smartphones	Standard Android smartphones	Wi-Fi enabled smartphone running Nougat or above with HEIMDALL	Any modern smartphone fitting preceding description.






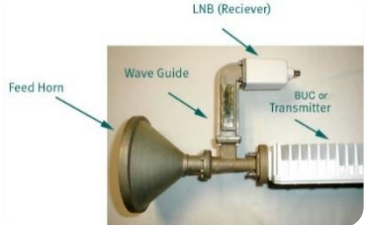


Segment	Equipment	Type	Description	Exemplar
			receiver app installed.	
<b>Remote command centre</b>	Fixed terminal	VSAT	Hughes Network Systems 74 cm, 98 cm VSAT dish	
	VSAT router	Hughes Network Systems	HT2000 high throughput (50+Mbps) GigE LAN port router. Supports all antenna sizes (74 cm, 98 cm, or 120 cm).	
	Wi-Fi router	Buffalo	MR9600 Dual-Band Mesh with up to 6 Gbps throughput.	
	VSAT accessories	Hughes Network Systems	Low Noise Block (LNB), 3W Block Up Converter (BUC) and feed horn.	
<b>Satellite gateway</b>	Hub	Hughes Network Systems	Hughes Network Systems HTS Jupiter Hub	
<b>Space Segment</b>	Geosynchronous satellite at 31°E	Ka-band HTS 26.64 GHz with 64 fixed beams.	HYLSA 4 satellite owned and operated by Avanti.	

Table 4-2 and Table 4-3 summarise the key service parameters.

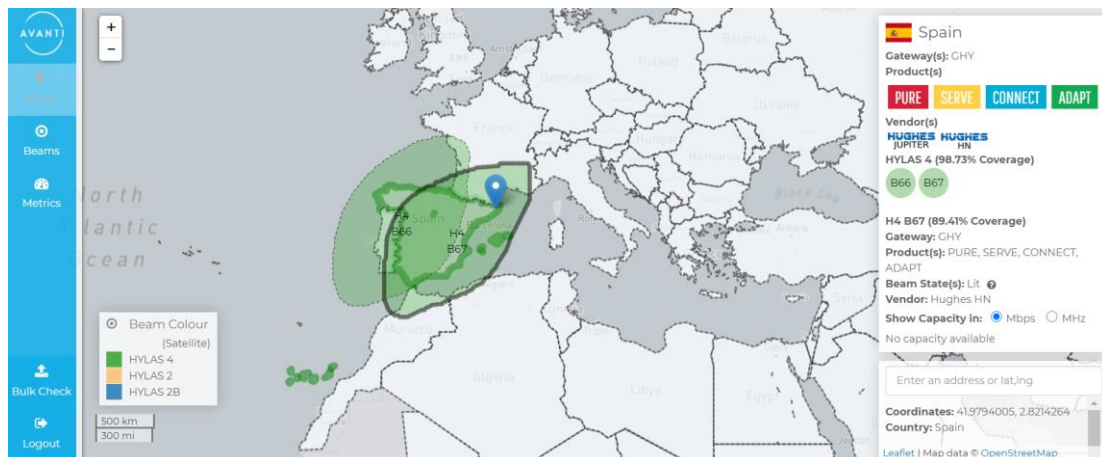


Figure 4-3: Service Parameters from Avanti VMAP Tool

Table 4-2: Satellite Service Parameters

Element	Description
Service type	Avanti Adapt service.  ADAPT customers operate as Virtual Network Operators, whereby Avanti provides wholesale dedicated bandwidth (in Mbps) to them to design and configure their own connectivity solutions using Avanti's Operational Support Systems (OSS) to configure and manage the service.
Satellite and beam	H4 beam 67
Site location	Girona, Spain (41.9794005, 2.8214264)
Service Level Agreement (SLA)	10Mbps FWD and 2Mbps RTN Committed Information Rate (CIR) for downloading high definition imagery and video streams.
Partner Name in Avanti OSS	HEIMDALL_Girona_Adapt_Demo

The key service installation stakeholders roles and descriptions are summarised in Table 4-3 below.

Table 4-3: Key Service Installation Stakeholders Roles and Descriptions

Stakeholder	Responsibility
Avanti Partner Support staff	Create install job, device, bandwidth plan and schedule job on Operations Support System (OSS).
VSAT install engineer	Install and testing of on-filed VSAT equipment via Avanti Mobile Assist.
Satellite and Network Operations Engineer	Allocate bandwidth on HYLAS 4 and configure carrier and hub.
Network Operations Engineer	Network configuration and IP addressing
L2 Technical Support	Service maintenance and technical helpdesk.

The end-to-end service elements that need to be tested prior to in-service operation are summarised in Table 4-4 below.

Table 4-4: Service Test Summary

Test element	Description	Outcome
Modem test	Transmit and receive tests.	Must pass
Terminal	Terminal availability and modem power to ascertain whether there are any dropped packets identified by the device.	Must pass
Latency test	Examine device to ascertain exact timing between the device and the satellite whilst it is dealing with data.	Must pass
Satellite signal	Ascertain actual signal achieved on the device	Must pass
Speed	Complete service speed tests over the service using dedicated speed tester	Must pass
Wi-Fi router to VSAT modem	Test and configure connection between Wi-Fi router to VSAT modem.	Must pass
Smartphone receiver app	Test connection to local model Wi-Fi router.	Must pass
Drones	Test connection to local model Wi-Fi router.	Must pass

## 5 Technical Specification

As already introduced in the previous section, the IG provides two services, the alerting service and the FR information service.

### 5.1 Alerting service

For the alerting service, alert messages are created through the user interface. The internal format to create and store locally the alert messages is the Common Alerting Protocol (CAP) [3]. Users will create the message using the different menus provided in the UI (using pre-defined alerting libraries) or manually introducing text in the given text boxes. The provided data items will be gathered at the IG and used to populate a CAP file containing the relevant alert information. Once all the mandatory data items have been inserted by the user, the possibility to disseminate the alert message will be available at the UI.

When receiving the request to disseminate the message, the IG triggers the internal creation of the message according to the selected alerting channels. The mapping between the alerting channels and the format used for each of them is stored in the dedicated configuration files, for the pre-operational version. In a general case, the following formats are provided by the alert generator module:

- **Common Alerting Protocol (CAP)** is an XML-based message format which is used by a wide number of alerting systems, thus allowing interoperability between already existing systems. By providing CAP compatibility, the system is able to input alerts in already operating systems based on the CAP format.
- **A4A Protocol:** The Alert4All protocol was designed and first implemented within the Alert4All project [1] in order to overcome the limitations that the transmission of CAP messages presented in narrowband channels. The A4A protocol is based on the use of extensible headers which include the encoded version of the message as well as the verbose version, if needed. The principle used for encoding the alert message information is based on the use of alerting libraries. These alerting libraries provide, for each information item to be included in the alert message according to the CAP protocol, a series of options to build the message. According to the chosen options, an encoded version of the alert message will be created and disseminated.

The alert forwarding then dispatches the alerts in the proper format to the alerting channels. Therefore, its Uniform Resource Locator (URL) encodes information of the message for the ACAPs. The URL used by the alert forwarding module has the following structure:

```
http://{user}:{password}@{ACAP-IP}/{path}/{reqType}/{IG-ID}/{incidentId:msgId}
```

If no authentication is required, the user and password fields are empty. The ACAP-IP describes the IP address of the ACAP. The IG-ID is an identifier for the IG used in case there are multiple instances; it is fixed for a given IG and is used by the corresponding ACAP to identify the source of the message. The “reqType” field holds information on the message format used. The web client then dispatches the alert via HTTP.

#### 5.1.1 Database

The web server and the IG main process have both accesses to a PostgreSQL database. The database includes five tables which are described in Table 5-1.

Table 5-1: IG database tables

Table name	Description	Fields of the table
area	Holds information on the area. Is used to map geo-	<ul style="list-style-type: none"> <li>• Geocode</li> </ul>

	codes to polygons and vice versa. The database is filled with data from all municipalities of the Catalan region (Spain) as preparation for the pilot demonstration.	<ul style="list-style-type: none"> <li>• Area description (human readable version)</li> <li>• Type (e.g. polygon)</li> <li>• Coordinates</li> <li>• Altitude</li> <li>• Ceiling</li> </ul>
channels	Holds information on the channels available.	<ul style="list-style-type: none"> <li>• Channel Identifier (ID)</li> <li>• Channel description (human readable version)</li> <li>• Sent (number of sent messages)</li> <li>• Successful (number of successfully transmitted messages)</li> <li>• Rating (good, low, medium)</li> </ul>
message_content	This table stores the sent message in human-readable version. To identify the corresponding CAP file, the incident and message ID is also stored.	<ul style="list-style-type: none"> <li>• Incident ID</li> <li>• Message ID</li> <li>• Content</li> </ul>
message_log	Stores all relevant data for an alert message.	<ul style="list-style-type: none"> <li>• Incident ID</li> <li>• Message ID</li> <li>• Timestamp</li> <li>• Status (true if alert was dispatched)</li> <li>• Headline</li> <li>• Channels (list of acknowledgements)</li> <li>• Expires (expire field of the CAP file)</li> </ul>
Subscriber_data	Holds the information of subscribers.	<ul style="list-style-type: none"> <li>• ID</li> <li>• Name</li> <li>• Email address</li> </ul>

## 5.1.2 Alerting Services Web Services Specification

This section gives the specification of the web services between the GUI and the IG.

The structuring of the methods is divided into HTTP-POST requests and HTTP-GET requests.

The web services provide the GUI with methods to create and manipulate alert messages at the IG. It is implemented by a PHP7 script running on the Linux machine with Apache2.

### 5.1.2.1 POST Methods

The POST methods URLs have the following structure:

```
http://<<IP>>/IG/<<postCommand>>
```

where <<IP>> is to be replaced by the IP address of the dedicated IG. Table 5-2 shows the commands that can be used with a POST request.

Table 5-2: HTTP POST commands of the IG web services

postCommand	Description
new	The command creates a new message ID.

set	With set commands a CAP can be generated and values in this CAP can be set.
dispatch	Dispatches an alert message to the ACAPs
reset	Reset of different values in the CAP

Output is either an error message with HTTP status code 400 or 404 in the form of:

```
<<error message>>
```

Or upon success a message with HTTP status code 200 with the JSON encoded content described in the output field in the tables below. If the output is an alert message it is defined as:

```
{
  "header": <<headline of the alert message>>,
  "content": <<content of the alert message>>,
  "complete": <<array with all cap fields that are missing>>,
  "freeText": <<flag that indicates if there is free text in the message>>
}
```

The free text flag is needed at the receiver to determine whether a message can be automatically translated using the MCE or not. Input variables are sent in the HTTP POST message body in JSON format. For all POST commands the same body structure is used. It consists of the incident ID, the message ID, a field called *selectedItem* and a field called *itemValue*. An example for the JSON body is:

```
{
  "incidentId": "INC201529100941",
  "messageId": "1234567890",
  "selectedItem": "event",
  "itemValue": "Forest Fire",
  "language": "en-GB"
}
```

For the incidentId and the messageId field any value is possible except “null” which throws an error message. With the selectedItem field it is determined which part of the CAP is changed. The itemValue field is the value that shall be set. In some commands the itemValue field can be empty. In Table 5-3 and Table 5-4 the possible values of the selectedItem field are presented for the set command, in Table 5-5 for the dispatch command, and in Table 5-6 the ones for the rest command, respectively. In Table 5-3 the **M** column shows the mandatory fields that have to be set in order to prepare a complete CAP message. If the check box is selected it means it is a mandatory field. The language of the input JSON can be any value but if the language is not available at the IG it is set to “en-GB” by default.

Table 5-3: IG POST body values for the set command

Service ID	M	Description	Input		Output
	<input type="checkbox"/>		selectedItem	itemValue	

IG_Set_1	<input checked="" type="checkbox"/>	Set the message type	msgType	“Standard” “Cancellation” “Release”	or or	Alert message
IG_Set_2	<input checked="" type="checkbox"/>	Set the event field	event	Earthquake, Heavy rain, Flood, Explosion, Terrorist attack, Forest Fire, Train crash, Traffic Accident , Toxic Cloud, “Free Text” as string		Alert message
IG_Set_3	<input checked="" type="checkbox"/>	Set the severity field	severity	Extreme, Severe, Moderate, Minor, Unknown, “Free Text” as string		Alert message
IG_Set_4	<input checked="" type="checkbox"/>	Set the certainty field	certainty	Observed, Likely, Possible, Unlikely, Unknown		Alert message
IG_Set_5	<input type="checkbox"/>	Set the instruction field	instruction	Free text		Alert message
IG_Set_6	<input checked="" type="checkbox"/>	Set the scope of the message	scope	“Public” or “Private”		Alert message
IG_Set_7	<input checked="" type="checkbox"/>	Set the response type	responseType	Shelter, Evacuate, Prepare, Execute, Monitor, None		Alert message
IG_Set_8	<input checked="" type="checkbox"/>	Set the urgency field	urgency	Immediate, Expected, Future, Past, Unknown, None		Alert message
IG_Set_9	<input type="checkbox"/>	Add a Language	language	{ “language”: “STRING”, “headline”: “STRING” “content”: “STRING” }		Alert message
IG_Set_10	<input type="checkbox"/>	Add a URI	uri	URI as string		Alert message

IG_Set_11	<input checked="" type="checkbox"/>	Set the date	onsetDate	"DD:MM:YYYY;hh:mm"	Alert message
IG_Set_12	<input type="checkbox"/>	Set the expiration date of the message	expirationDate	"DD:MM:YYYY;hh:mm"	Alert message
IG_Set_13	<input checked="" type="checkbox"/>	Sets the area of to alert.	area	<pre>{   "geocodes": "STRING separated by comma if multiple" } or {   "circleDesc": "STRING",   "circle": "WGS84 coordinate pair followed by a space character and a radius value in kilometres" } or {   "polyDesc": "STRING",   "polygon": " whitespace-delimited list of WGS 84 coordinate pairs" } } Or any combination of the above three options.</pre>	Alert message
IG_Set_14	<input type="checkbox"/>	Set the addresses in case of a private message this is mandatory.	addresses	Addresses as string. Multiple space-delimited addresses MAY be included. Addresses including whitespace MUST be enclosed in double-quotes.	Alert message
IG_Set_15	<input checked="" type="checkbox"/>	Identifies the sender of the message	sender	sender as string	Alert message
IG_Set_16	<input type="checkbox"/>	Sets the web page	web	String if not set default page is used	Alert message
IG_Set_17	<input type="checkbox"/>	Channels for the transmission can be selected. All available are used as default	channels	Channels as string separated by comma  Possible values TBD	Alert message

One exception from the POST request structure above is the Service IG\_Set\_18. This service is for uploading a file. Since files can be received from different sources the base64 formatting was shifted to the IG. Encoded files are sent to the IG. With this, the HTTP body is occupied by the file which is sent in content type multipart/form. The URL for this function is given by:

<http://<<IP>>/IG/addFile/<<lang>>/<<incidentId>>:<<messageId>>>



Table 5-4: IG POST body values for the add file command

Service ID	Description	Input	Output
IG_Set_18	A file is attached	File in post body	"ack": added file

Table 5-5: IG POST body values for the dispatch command

Service ID	Description	Input		Output
		selectedItem	itemValue	
IG_Send_1	The message is dispatched via the ACAPs	empty	empty	Alert message

Table 5-6 IG POST body values for the reset command

Service ID	Description	Input		Output
		selectedItem	itemValue	
IG_Rst_1	Resets all fields besides: identifier, incidents, status, msgType, areaDesc	reset	empty	Alert message
IG_Rst_2	A, in a previous attached step, additional language field is removed.	removeLanguage	Language code	"ack": Acknowledgement (message that can be shown to the user)
IG_Rst_3	A, in a previous attached step, resource is removed.	removeResource	"file"	"ack": Acknowledgement (message that can be shown to the user)
IG_Rst_4	A list with message IDs given for an incident ID is send. A combined cancellation message is dispatched.	cancelmessageID	message ID	"ack": Acknowledgement (message that can be shown to the user with the list of cancelled messages)

### 5.1.2.2 GET Methods

In GET requests the variables required are URL encoded. They are either the incident ID and the message ID or only the incident ID. Like the POST commands, GET requests have always a main command word in the URL. For the GET requests these commands are shown in Table 5-7. Below, GET commands for filtering are presented. These can be used to access the IG alerting database. The URL of GET requests are the following:

```
http://<<IP>>/IG/<<getCommand>>/<<lang>>/<<id>>
```

where <<id>> in the URL shall be set to <<incidentId:messageId>> for commands all commands not including the filter keyword and to <<incidentId>> for commands including the filter keyword. The <<getCommand>> needs to be replaced with a value from Table 5-7 in the column getCommand in order to call the dedicated method.

Table 5-7: HTTP GET commands of the IG web services

Service ID	Description	getCommand	Output
IG_Show_3	Returns the message and the indicators if the message is complete or contains free text	checkMessage	Alert message
IG_Show_4	Returns an JSON encoded array with n messages that fit the input incident ID	filter/all	<pre>{   "incidentId": incidentId   "message"::[n] }</pre> <p>Where a "message" array is:  "messageId": messageId,  "date": timestamp,  "time": timestamp,  "status": status,  "headline": headline</p> <p>The status field hold if the message has be sent or only stored</p>
IG_Show_5	Returns a JSON encoded array with n messages that fit the input incident ID and have been sent.	filter/sent	<pre>{   "incidentId": incidentId   "message"::[n] }</pre> <p>Where a "message" array is:  "messageId": messageId,  "date": timestamp,  "time": timestamp,  "status": status,  "headline": headline</p> <p>The status field hold if the message has be sent or only stored</p>
IG_Show_6	Returns a JSON encoded array with n messages that fit the input incident ID and that have not sent but are stored by the system.	filter/stored	<pre>{   "incidentId": incidentId   "message"::[n] }</pre> <p>Where a "message" array is:  "messageId": messageId,  "date": timestamp,  "time": timestamp,  "status": status,  "headline": headline</p> <p>The status field hold if the message has be sent or only stored</p>

IG_Show_7	Returns a JSON encoded array with <i>n</i> messages that fit the input incident ID or were cancelled	filter/cancelled	<pre>{   "incidentId": incidentId   "message": [n] }</pre> <p>where a "message" array is:  "messageId": messageId,  "date": timestamp,  "time": timestamp,  "status": status,  "headline": headline</p> <p>The status field hold if the message has be sent or only stored</p>
IG_Show_8	Returns a JSON encoded array with <i>n</i> messages that fit the input incident ID or were not cancelled	filter/notCancelled	<pre>{   "incidentId": incidentId   "message": [n] }</pre> <p>Where a "message" array is:  "messageId": messageId,  "date": timestamp,  "time": timestamp,  "status": status,  "headline": headline</p> <p>The status field hold if the message has be sent or only stored</p>
IG_Show_9	Returns the cap message (xml), for testing.	getCap	CAP message in XML

## 5.2 First Responders Information Web Services Specification

The first responder information service is based on output from deliverable D6.7 Situation report and accordingly will be based on EDXL-SitRep. The users creates a SitRep of the scenario and adds necessary information for the EDXL-DE. The GUI forwards all information to the IG which combines the information for the EDXL-DE with the EDXL-SitRep.

### 5.2.1 POST methods

The POST methods URLs have the following structure:

```
http://<<IP>>/IG/sitrep/<<postCommand>>
```

where <<IP>> is to be replaced by the IP address of the dedicated IG. Table 5-2 Table 5-2 shows the commands that can be used with a POST request.

Table 5-8: HTTP POST commands of the Information web services

postCommand	Description
set/incidentID	With set commands an information message can be generated and values in this message can be set.

dispatch/incidentID:messageID	Dispatches an information message to the FRs.
-------------------------------	---

For error, output is either an error message with HTTP status code 400 or 404. For set/incidentID, body should be in the below format:

```
{
  "senderID": "actor@domain-name",
  "dateTimeExpires": "yyyy-mm-ddThh:mm",
  "distributionStatus": "Actual",
  "confidentiality": "Classified",
  "Language": "en-GB",
  "recipientRole": "Incident commander",
  "address": "Polizia Municipale",
  "area": {
    "geocodes": "geocodes from SP, separated by comma",
    "circleDesc": "Name of the circle area",
    "circle": "WGS84 coordinate pair followed by a space character and a radius value in kilometers",
    "polyDesc": "Name of the polygon area",
    "polygon": "Whitespace-delimited list of WGS 84 coordinate pairs"
  },
  "sitrep": <input sitrep in base64 format>
}
```

On success, HTTP code 200 will be returned along with created message ID in the body, for example,

```
INC20190828110834
```

For dispatch/incidentID:messageID, body is ignored. Upon success, HTTP 200 code will return along with message in the body:

```
dispatched message
```

## 5.2.2 GET methods

The command to see all possible options in the input is:

```
http://<<IP>>/IG/sitrep/options
```

and the output is,

```
{
  "set": {
    "senderID": {
      "type": "string",
      "itemValue": "actor@domain-name",
      "default": false,
      "required": true
    }
  }
}
```

```
    },
    "dateTimeExpires": {
      "type": "string",
      "itemValue": "yyyy-mm-ddThh:mm",
      "default": false,
      "required": true
    },
    "distributionStatus": {
      "type": "enumeration",
      "itemValue": ["Actual",
        "Exercise",
        "System",
        "Test",
        "Unknown"],
      "default": false,
      "required": true
    },
    "confidentiality": {
      "type": "enumeration",
      "itemValue": ["Unclassified",
        "Classified"],
      "default": false,
      "required": true
    },
    "Language": {
      "type": "string",
      "itemValue": ["en-GB",
        "es-ES"],
      "default": "en-GB",
      "required": false
    },
    "recipientRole": {
      "type": "enumeration",
      "itemValue": ["First responder",
        "Incident commander",
        "Crisis manager",
        "Communications officer",
        "Administration and finance",
```

```
        "Sector commander",
        "Command and control",
        "Training coordinator",
        "Resilience manager"],
        "default": false,
        "required": true
    },
    "address": {
        "type": "enumeration",
        "itemValue": ["Polizia Municipale",
        "Croce Rossa Italiana",
        "CNR - Istituto di Ricerca per la Protezione Idrogeologica",
        "Free Text"],
        "default": false,
        "required": true
    },
    "area": {
        "type": "JSON",
        "itemValue": {
            "geocodes": {
                "type": "string",
                "itemValue": "geocodes from SP, separated by comma",
                "default": false,
                "required": false
            },
            "circleDesc": {
                "type": "string",
                "itemValue": "Name of the circle area",
                "default": false,
                "required": "if circle is set"
            },
            "circle": {
                "type": "string",
                "itemValue": "WGS84 coordinate pair followed by a space
character and a radius value in kilometers",
                "default": false,
                "required": false
            }
        },
    },

```

```

        "polyDesc": {
            "type": "string",
            "itemValue": "Name of the polygon area",
            "default": false,
            "required": "if polygon is set"
        },
        "polygon": {
            "type": "string",
            "itemValue": "Whitespace-delimited list of WGS 84
coordinate pairs",
            "default": false,
            "required": false
        }
    },
    "default": false,
    "required": true
},
"sitrep": {
    "type": "string",
    "itemValue": "EDXL-SitRep (XML)",
    "default": false,
    "required": true
}
}
}

```

To see status of information messages in an incident ID, filter methods can be used in the same way as mentioned in Section 5.1.2.2. The only difference is that, the headline field will not be used here.

### 5.2.3 DELETE method

The command to delete an information message is,

```
http://<<IP>>/IG/sitrep/rmv/incidentID:messageID:LUID
```

Upon success, HTTP 200 will be returned.

## 6 Test Plan

### 6.1 IG Verification

The IG can be tested by connecting it to the web-based GUI or by directly calling its web services with a REST client as test receiver to verify the correct transmission of alert messages. For all web services defined in Section 5.1.2 a test was performed using a correct and an incorrect input. In the *correct* case, the IG shall perform the specified action and return the specified reply. In the *incorrect* case, the IG shall reply with the specified error message.

A receiver app from previous project is used for the first tests. The HEIMDALL receiver app will be used as soon as the alert receiver is implemented.

<b>Test ID</b>	TS_IG_01
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>TR_Com_1</li> <li>TR_Com_2</li> </ul>
<b>Test objective</b>	To verify that the IG is able to keep the population informed.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>The user creates an alert message using the GUI</li> <li>The user dispatches the created alert message</li> <li>Alert is shown at the receiver application</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>IG connected to the HEIMDALL system</li> <li>Receiver application connected to the HEIMDALL system</li> <li>GUI connected to the HEIMDALL system</li> </ul>
<b>Success criteria</b>	Correct alert message received at the receiver application
<b>Results analysis</b>	<i>The test has been performed using a receiver application in a lab environment. Nevertheless, the app has the potential to be used as a receiver for the public.</i>
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_02
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>TR_Com_3</li> </ul>
<b>Test objective</b>	To verify that the IG is able to create alert messages.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>The user starts the alert creating wizard using the GUI</li> <li>The user creates an alert message following the wizard</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>IG connected to the HEIMDALL system</li> <li>GUI connected to the HEIMDALL system</li> </ul>
<b>Success criteria</b>	Correct alert message according to CAP is created.
<b>Results analysis</b>	<i>The created CAP file has been verified using an online CAP verification tool.</i>
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_06
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>TR_Com_5</li> </ul>



<b>Test objective</b>	To verify that the IG is able to give access to users in the field.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. A user connects to the system using the smartphone application using credentials that give him only partial access.</li> <li>2. Web GUI creates an alert or FR information message.</li> <li>3. The user can view specific data sets or an overview shared by the Web GUI to the app user</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG connected to the HEIMDALL system</li> <li>• Smartphone with HEIMDALL App installed connected to the HEIMDALL system.</li> <li>• Web GUI connected to the HEIMDALL system.</li> </ul>
<b>Success criteria</b>	The user can access the data when a notification arrives.
<b>Results analysis</b>	-
<b>Success</b>	Passed.

<b>Test ID</b>	TS_IG_07
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_14</li> </ul>
<b>Test objective</b>	To verify that the IG is able to consider predefined areas of different levels
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. An alert message is created and different levels of areas are selected at the GUI</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG connected to the HEIMDALL system</li> <li>• GUI connected to the HEIMDALL system</li> </ul>
<b>Success criteria</b>	Areas of all levels are updated accordingly at the alert message
<b>Results analysis</b>	-
<b>Success</b>	Passed.

<b>Test ID</b>	TS_IG_08
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_15</li> </ul>
<b>Test objective</b>	To verify that the IG is able to consider predefined areas of different levels
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. An alert message is created and types of areas are selected at the GUI</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG connected to the HEIMDALL system</li> <li>• GUI connected to the HEIMDALL system</li> </ul>
<b>Success criteria</b>	Areas of different types are updated accordingly at the alert message
<b>Results analysis</b>	-
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_09
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_16</li> </ul>

<b>Test objective</b>	To verify that the IG is able to consider different types of users as recipients within the private message
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. An alert message is created and as scope private is selected.</li> <li>2. Using the set addresses method, different user groups according to CAP are set</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG connected to the HEIMDALL system</li> <li>• GUI connected to the HEIMDALL system</li> </ul>
<b>Success criteria</b>	The groups are updated accordingly at the alert message and the message is set to private
<b>Results analysis</b>	<i>The tested fields are free text so all kind of users can be included as per the API. The GUI includes a set of predefined values for usability.</i>
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_10
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_11</li> <li>• TR_Com_12</li> </ul>
<b>Test objective</b>	To verify that the IG is able to connect communication networks.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. The machine of the IG connects to the HEIMDALL VPN</li> <li>2. Test requests are sent from SP side to the IG.</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• Machine of the IG server is up and running</li> </ul>
<b>Success criteria</b>	The SP and the IG can exchange data.
<b>Results analysis</b>	<i>The machine can connect to communication networks</i>
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_11
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_21</li> </ul>
<b>Test objective</b>	To verify that the IG is able to share pictures
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. An alert message is created</li> <li>2. Using the add file method a picture file is attached to the message</li> <li>3. The message is dispatched</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG is up and running</li> </ul>
<b>Success criteria</b>	The message is forwarded including the picture
<b>Results analysis</b>	<i>The file was added using a test function.</i>
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_13
----------------	----------

<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_20</li> </ul>
<b>Test objective</b>	To verify that the IG is able to share data sets to users in the field
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. An information message is created with required data sets.</li> <li>2. The message is dispatched</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG is up and running</li> </ul>
<b>Success criteria</b>	The message is forwarded to the mobile app and displayed.
<b>Results analysis</b>	
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_14
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_22</li> </ul>
<b>Test objective</b>	The IG shall use standard formats where possible
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. Create an alert message</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG is up and running</li> </ul>
<b>Success criteria</b>	The results is stored in CAP compliant format
<b>Results analysis</b>	<i>The CAP file generated at the IG has been validated using external tools.</i>
<b>Success</b>	Passed

<b>Test ID</b>	TS_IG_15
<b>Requirements to be verified</b>	<ul style="list-style-type: none"> <li>• TR_Com_23</li> </ul>
<b>Test objective</b>	The IG shall ask a confirmation before dispatching the alert.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. Create an alert message</li> <li>2. Check the alert message before dispatching</li> <li>3. Confirm dispatching</li> </ol>
<b>Test prerequisites/configuration</b>	<ul style="list-style-type: none"> <li>• IG is up and running</li> </ul>
<b>Success criteria</b>	The message is only dispatched after a confirmation
<b>Results analysis</b>	<i>The IG is requesting the confirmation before dispatching. Who actually will be responsible for the confirmation is part of the GUI and access management.</i>
<b>Success</b>	Passed

## 7 Conclusion

This document presented the design and implementation of the information gateway and satellite communication modules developed for the HEIMDALL platform. Requirements and implementation of the HEIMDALL smartphone application is presented. The TRs presented here include the user feedback and suggestions collected in the course of the project. The IG functionalities developed presented herein were successfully integrated and tested as part of the second end user workshop (EUW2, M18). A description of the satellite telecommunication solution that can be used to provide connectivity to FRs at a disaster site and for communication with remote command centres is also presented.

## 8 References

- [1] Alert4All: Alert for all, available at: <https://cordis.europa.eu/project/rcn/98427/factsheet/en> [last accessed Feb. 2019]
- [2] PHAROS: *Project on a multi-hazard open platform for satellite based downstream services* available at <https://cordis.europa.eu/project/rcn/188829/reporting/en> [last accessed Feb. 2019]
- [3] OASIS Standard, Common Alerting Protocol Version 1.2 available at <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html> [last accessed Feb. 2019]
- [4] Barth, B. (2020), HEIMDALL D2.9: HEIMDALL Requirements Report – Issue 4
- [5] To be released, Barth, B. (2021), HEIMDALL D2.10: HEIMDALL Requirements Report – Issue 5
- [6] To be released Barth, B. et al. (2020), HEIMDALL D5.7: First Responders Data Module Design
- [7] Friedemann, M. et al. (2020), HEIMDALL D6.15: “Scenario Specification, Scenario Management Specification and Scenario and Situation Metrics - Final“
- [8] OASIS Standard, Emergency Data Exchange Language Situation Reporting (EDXL-SitRep) Version 1.0 available at <http://docs.oasis-open.org/emergency/edxl-sitrep/v1.0/csprd03/edxl-sitrep-v1.0-csprd03.html> [last accessed Oct. 2020]
- [9] OASIS Standard, Emergency Data Exchange Language (EDXL) Distribution Element Version 2.0 available at <http://docs.oasis-open.org/emergency/edxl-de/v2.0/edxl-de-v2.0.html> [last accessed Oct. 2020]
- [10] To be released in M38 (2020), HEIMDALL D4.17: Communication to Remote Areas – Design and Specifications – Final

**End of document**