Functional/Technical requirements & System architecture

Deliverable 1.2

Work package: **WP1**
Dissemination level: **PU**
Lead partner: ENG
Authors: **Nicoletti Tommaso, Giuseppe Vella**
Due date: **31/05/2023**
Submission date: **02/06/2023**

The OVERWATCH project has received funding from the Horizon Europe call "HORIZON-EUSPA-2021", topic HORIZON-EUSPA-2021-SPACE-02-52, under agreement No. 101082320

Deliverable	**Functional/Technical requirements & System architecture**
Deliverable No.	D1.2
Work Package	1
Dissemination Level	PU
Nature[1]	Report
Author(s)	Tommaso Nicoletti (ENG), Giuseppe Vella (ENG),
Co-Author(s)	Edoardo Arnaudo (LINKS), Brendan David Polidori (LINKS), Hugo Silva (INESCTEC), Helder Fontes (INESCTEC), Ruben Queiros (INESCTEC), Iuliu Novac (ROB), Anna Nałęcz-Kobierzycka (CBK), Anna Foks -Ryznar (CBK), Emil Wrzosek (CBK)
Date	02/06/2023
Status	Final
Revision	0.5
Reviewed by (if applicable)	LINKS, CBK, CINAMIL
Information to be used for citations of this report	Tommaso Nicoletti (ENG), Giuseppe Vella (ENG), Edoardo Arnaudo (LINKS), Brendan David Polidori (LINKS), Hugo Silva (INESCTEC), Helder Fontes (INESCTEC), Ruben Queiros (INESCTEC), Iuliu Novac (ROB), Anna Nałęcz-Kobierzycka (CBK), Anna Foks -Ryznar (CBK), Emil Wrzosek (CBK) (2023): *Functional/Technical requirements & System architecture*, D1.2, OVERWATCH. Horizon Euspa Space 2021 Grant Agreement No 101082320, XX pp.

Deliverable abstract	This deliverable focuses on the System Architecture, highlighting the alignment of functional requirements with end user needs across the constituent modules. The initial architectural draft and its envisioned service layers are introduced. Detailed descriptions of each module encompass their functional requirements, technical specifications, and the anticipated data exchange interfaces with expected users. Furthermore, the interactions between modules are explored, emphasizing key requirements related to technical aspects, testing, performance, reliability, and availability. The connectivity assumptions for different use cases are addressed to

[1] Nature of the deliverable: **R** = Report, **P** = Prototype, **D** = Demonstrator, **O** = Other

	identify optimal solutions. Lastly, the deliverable concludes with final remarks summarizing the key findings.
Keywords	functional requirements, technical requirements, system architecture, data formats, dataflow, interfaces, Disaster management, end-user needs

Table of Content

Table of Content 4

Figures 8

Tables 8

Document revision history 10

List of authors, contributors and reviewers 10

Abbreviations 10

Executive Summary 14

Introduction 14

Goal of the deliverable 14

1. The Architecture in a nutshell 15

2. Technical modules description and functional requirements 16

2.1. Management Backend 16

2.1.1. Brief description of the tool/module 16

2.1.2. Expected users of the system 17

2.1.3. Description of capabilities of the tool/module 17

2.1.4. Expected technical inputs and expected outputs 17

2.1.5. Available interfaces 20

2.1.6. Mapping of the functional requirements and the end user requirements within the technical module 20

2.2. Web-based dashboard 23

2.2.1. Brief description of the tool/module 23

2.2.2. Expected users of the system 23

2.2.3. Description of capabilities of the tool/module 23

2.2.4. Expected technical inputs and expected outputs 24

2.2.5. Available interfaces 27

2.2.6. Mapping of the functional requirements and the end user requirements within the technical module 27

2.3. Mapping from EO 29

2.3.1. Brief description of the tool/module 29

2.3.2. Expected users of the system 30

2.3.2.1. Input required from the end users 30

2.3.3. Description of capabilities of the module 31

2.3.4. Expected technical inputs and expected outputs 31

2.3.5. Available interfaces 34

2.3.6. Mapping of the functional requirements and the end user requirements within the technical module 34
2.4. AR Backend 36
2.4.1. Brief description of the tool/module 36
2.4.2. Expected users of the system 36
2.4.3. Description of capabilities of the tool/module 36
2.4.4. Expected technical inputs and expected outputs 37
2.4.5. Available interfaces 39
2.4.6. Mapping of the functional requirements and the end user requirements within the technical module 39
2.5. AR Front end 41
2.5.1. Brief description of the tool/module 41
2.5.2. Expected users of the system 41
2.5.3. Description of capabilities of the tool/module 42
2.5.4. Expected technical inputs and expected outputs 42
2.5.5. Available interfaces 42
2.5.6. Mapping of the functional requirements and the end user requirements within the technical module 42
2.6. Accurate and secure drone navigation 43
2.6.1. Brief description of the tool/module 43
2.6.2. Expected users of the system 43
2.6.3. Description of capabilities of the tool/module 43
2.6.4. Expected technical inputs and expected outputs 43
2.6.5. Available interfaces 44
2.6.6. Mapping of the functional requirements and the end user requirements within the technical module 44
2.7. Drone terrain mapping 44
2.7.1. Brief description of the tool/module 44
2.7.2. Expected users of the system 44
2.7.3. Description of capabilities of the tool/module 45
2.7.4. Expected technical inputs and expected outputs 45
2.7.5. Available interfaces 47
2.7.6. Mapping of the functional requirements and the end user requirements within the technical module 47
2.8. Fallback communication system 48
2.8.1. Brief description of the tool/module 48
2.8.2. Expected users of the system 49
2.8.3. Description of capabilities of the tool/module 49

2.8.4. Expected technical inputs and expected outputs ... 49
2.8.5. Available interfaces ... 49
2.8.6. Mapping of the functional requirements and the end user requirements within the technical module ... 50
3. Data flow diagrams and interactions with the other modules ... 51
3.1. Management backend ... 51
3.1.1. Workflow diagram of the tool and operational concept ... 51
3.1.2. Technical requirements ... 51
3.1.2.1. Testing requirements ... 51
3.1.2.2. Performance requirements ... 52
3.1.2.3. Reliability requirements ... 52
3.1.2.4. Availability requirements ... 53
3.1.3. Interactions with the other modules ... 53
3.1.4. Roadmap of the release of the module ... 53
3.2. Web-based dashboard ... 54
3.2.1. Workflow diagram of the tool and operational concept ... 54
3.2.2. Technical requirements ... 54
3.2.2.1. Testing requirements ... 54
3.2.2.2. Performance requirements ... 54
3.2.2.3. Reliability requirements (any suggestions?) ... 55
3.2.2.4. Availability requirements ... 55
3.2.3. Interactions with the other modules ... 55
3.2.4. Roadmap of the release of the module ... 56
3.3. Mapping from EO ... 56
3.3.1. Workflow diagram of the tool and operational concept ... 56
3.3.2. Technical requirements ... 57
3.3.2.1. Testing requirements ... 57
3.3.2.2. Performance requirements ... 57
3.3.2.3. Reliability requirements (any suggestions?) ... 58
3.3.2.4. Availability requirements ... 58
3.3.3. Interactions with the other modules ... 58
3.3.4. Roadmap of the release of the module ... 59
3.4. AR backend ... 59
3.4.1. Workflow diagram of the tool and operational concept ... 59
3.4.2. Technical requirements ... 60
3.4.2.1. Testing requirements ... 61
3.4.2.2. Performance requirements ... 62

3.4.2.3. Reliability requirements ... 63
3.4.2.4. Availability requirements ... 63
3.4.3. Interactions with the other modules ... 64
3.4.4. Roadmap of the release of the module ... 64
3.5. AR front end ... 64
3.5.1. Workflow diagram of the tool and operational concept ... 64
3.5.2. Technical requirements ... 65
3.5.2.1. Testing requirements ... 65
3.5.2.2. Performance requirements ... 66
3.5.2.3. Reliability requirements ... 66
3.5.2.4. Availability requirements ... 66
3.5.3. Interactions with the other modules ... 67
3.5.4. Roadmap of the release of the module ... 67
3.6. Accurate and secure drone navigation ... 67
3.6.1. Testing requirements ... 67
3.6.2. Performance requirements ... 68
3.6.3. Reliability requirements ... 68
3.6.4. Availability requirements ... 68
3.6.5. Roadmap of the release of the module ... 68
3.7. Drone terrain mapping ... 69
3.7.1. Workflow diagram of the tool and operational concept ... 69
3.7.2. Technical requirements ... 70
3.7.2.1. Testing requirements ... 70
3.7.2.2. Performance requirements ... 71
3.7.2.3. Reliability requirements ... 71
3.7.2.4. Availability requirements ... 72
3.7.3. Interactions with the other modules ... 72
3.7.4. Roadmap of the release of the module ... 72
3.8. Fallback connectivity ... 72
3.8.1. Workflow diagram of the tool and operational concept ... 72
3.8.2. Technical requirements ... 73
3.8.2.1. Testing requirements ... 73
3.8.2.2. Performance requirements ... 74
3.8.2.3. Reliability requirements ... 74
3.8.2.4. Availability requirements ... 75
3.8.3. Interactions with the other modules ... 75
3.8.4. Roadmap of the release of the module ... 75

4. Connectivity assumptions 76
5. Conclusions 78

Figures

Figure 1 - The OVERWATCH architecture in a nutshell. 15
Figure 2 - Workflow diagram of the module and integration logic of the Management backend 51
Figure 3 - sample integration of the backend with the AI Mapping from EO and the AR backend 53
Figure 4 - Workflow diagram of the module and integration logic 54
Figure 5 - sample integration of the Web Based Dashboard 55
Figure 6 - Mapping from EO, operational concept. 56
Figure 7 - Interaction diagram for EO mapping 59
Figure 8 - Workflow diagram of the module and integration logic. 59
Figure 9 - sample integration of the Management Backend with AR backend. 64
Figure 10 - Workflow diagram of the module and integration logic. 65
Figure 11 - sample integration of the AR Backend with the AR Frontend. 67
Figure 12 - Drone terrain mapping operational concept. 69
Figure 13 - Workflow diagram of the module and integration logic. 73
Figure 14 - Network Connections when existing cellular connectivity is used to collect UAV data. 76
Figure 15 - Network Connections when Fallback Connectivity using LEO Satellite backhaul is used to collect UAV data. 77
Figure 16 - Network Connections considering both Cellular and Fallback Connectivity options to collect drone data. 77

Tables

Table 1 - Components and layers of the architecture involved 16
Table 2 - Management backend inputs and outputs for the current situation 17
Table 3 - Management backend inputs and outputs for image processing 18
Table 4 - Management backend inputs and outputs for requests form services 19
Table 5 - Management backend inputs and outputs for representing information on events 19
Table 6 - Functional requirements for management backend 20
Table 7 - Web based dashboard inputs and outputs for static layers. 24
Table 8 - Web based dashboard inputs and outputs for the visualization of the information. 25
Table 9 - Web based dashboard inputs and outputs for the representation of features on the map26
Table 10 - Web based dashboard inputs and outputs for enriching the situation awareness. 26
Table 11 - Functional requirements for Web based dashboard 27
Table 12 - Mapping from Earth Observation imagery request 31
Table 13 - CBK Floods (surface water extent) 32
Table 14 - Global Flood Monitoring (GFM) 33
Table 15 - Automated AI-based mapping from EO imagery 33
Table 16 - Copernicus EMS On Demand Mapping 33
Table 17 - On-demand mapping (by CIK CBK PAN) 34
Table 18 - Functional requirements for Mapping from Earth Observation module 34
Table 19 - AR Backend inputs and outputs for Remote rendering component 37

Table 20 - AR Backend inputs and outputs for Client Component (Frontend application) 37
Table 21 - AR Backend inputs and outputs for Signalling Server 38
Table 22 - AR Backend inputs and outputs for Map data interface 38
Table 23 - AR Backend inputs and outputs for Additional interfaces 39
Table 24 - Functional requirements for AR Backend 39
Table 25 - Expected inputs and outputs for AR Frontend. 42
Table 26 - Inputs and outputs from accurate drone secure navigation. 43
Table 27 - Functional requirements for accurate and secure drone navigation 44
Table 28 - Drone terrain mapping expected inputs and outputs for real time mapping. 45
Table 29 - Drone terrain mapping expected inputs and outputs for Area selection for navigation. . 45
Table 30 - Drone terrain mapping expected inputs and outputs for VTOL Telemetry. 46
Table 31 - Drone terrain mapping expected inputs and outputs for Orthorectified images. 46
Table 32 - Functional requirements for Drone terrain mapping 47
Table 33 - Fallback communication system expected inputs and outputs. 49
Table 34 - Functional requirements for the fallback communication system 50

Document revision history

Version	Date	Modification reason	Modified by
0.1	15/02/2023	Table of contents and work allocation after diagram design in power point	ENG, LINKS, CBK, CINAMIL
0.2	20/03/2023	Templates refinement and collection of the data flow diagrams and interaction protocols and technical module description	LINK, ROBOTTO, INESCTEC, ISQ, CINAMIL, ENG
0.3	22/05/2023	Collection of the information technical modules	HOLO, ITHACA, LINKS
0.4	23/05/2023	Final version ready to be peer reviewed	ENG
0.5	31/05/2023	Final version with revisions added	ENG, CBK, LINKS and CINAMIL

List of authors, contributors and reviewers

No.	Name	Role	Organisation
1	Tommaso Nicoletti	Author	ENG
2	Giuseppe Vella	Author	ENG
3	Edoardo Arnaudo	Author, Reviewer	LINKS
4	Brendan David Polidori	Author	LINKS
5	Hugo Silva	Author	INESCTEC
6	Helder Fontes	Author	INESCTEC
7	Ruben Queiros	Author	INESCTEC
8	Iuliu Novac	Author	ROBOTTO
9	Anna Nałęcz-Kobierzycka	Reviewer	CBK
10	Anna Foks - Ryznar	Author	CBK
11	Emil Wrzosek	Contributor	CBK

Abbreviations

AI Artificial intelligence

AIMS	Artificial Intelligence Backend Management System
API	Application Programming Interface
AR	Augmented reality
ARM	Advanced RISC Machine
BPL	Broadband over power lines
C2	Command and control
CH	Chapter
CBI	Composite burned index
COTS	Commercial of the shelve
D	Deliverable
DM	Decision-making
DSM	Digital Surface Model
DTM	Digital Terrain Model
ECS	Emergency communication systems
EGNSS	European Global Navigation Satellite System
EGNOS	European geostationary navigation overlay service
EMS	Emergency Management System
EUB	End-users board
ETAG	Entity TAG
EO	Earth observation
EMS	Emergency management and security
ER	Emergency resonders
FCS	Fallback Connectivity System
FR	First responder
FSX	Full-scale exercise [an exercise that involves multiple organisations or functions and includes actual activities (ISO22300:2021, n.d.)]
GCS	Ground control station
GeoJSON	Geographical JavaScript Object Notation
GeoTIFF	Geographical Tag Image File Format
GIS	Geographic information systems
GFM	Global Flood Monitoring
GNSS	Global navigation satellite system
GPS	Global Positioning System
GPU	Graphics Processing Unit
LIDAR	Laser imaging, detection, and ranging
HAS	High Accuracy Service

HDFS	Hadoop Distribuited File System
ISAR	Interactive streaming for augmented reality
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
KMZ	Keyhole Markup Language Zipped
LEO	Low Earth Orbit
MRTK	Mixed Reality Toolkit
NBR	Normalised burn ratio
NMEA	National Marine Electronics Association
noSQL	Non Structured Query Language
NGO	Non-governmental organisations
NVME	Non -Volatile Memory Express
O-RAN	Open Radio Access Technologies
OSM	Open Street Map
OSNMA	Open service navigation message authentication
POV	Point of View
PVT	Position Velocity and Time
R&D	Research and development
RBAC	Role Based Access Control
REST	RE presentational State Transfer
RGB	Red Green Blue
RTK	Real-Time Kinematic
SA	Situational awareness
SAR	Search and rescue
SSD	Solid State Disk
SART	Situation assessment and reconnaissance teams
SDK	Software development kit
SLAM	Simultaneous Localization And Mapping
SotA	State of the art
STUN	Session Traversal Utilities for NAT
TCP	Transmission Control Protocol
TO	Theatre of operations
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UI	User Interface

UUID	Universally unique identifier
VR	Virtual reality
VTOL	Vertical Take-Off and Landing
WebRTC	Web Real-Time Communication
WiFi	Wireles Fidelity
WMS	Web Map Service
XAML	eXtensible Application Markup Language
XML	eXtensible Markup Language
XR	Extended reality

Executive Summary

The OVERWATCH initiative aims to build a comprehensive crisis management system that improves communication, information gathering, and coordination among disaster response teams. The project's main goals involve developing a user-friendly, decentralized, and well-informed tool for effectively managing various disasters. The project aims to enhance the response infrastructure's safety, resilience, and capability while minimizing risks to human lives and material assets.

This deliverable focuses on the System Architecture and highlights the covering of functional requirements by the end-user requirements through the modules that compose the architecture.

In section 1 the architecture and its layers are presented.

In section 2 each module is described with its functional requirements and technical specifications. Then the expected users of the system will be described as the tools' capability and ability to exchange data specifying input and outputs that we foresee exchanging with each module through a specific interface. In the end, we map the functional requirements of each module with the end user requirements.

Once described a module from an internal perspective, win section 3, the interactions with the other modules are described, expressing a set of requirements: technical, testing, performance, reliability and availability of each module. In section 4 all the connectivity hypotheses are done to choose the best solution depending on the use cases. In section 5 conclusions are illustrated.

Introduction

Goal of the deliverable

The present deliverable is an outcome of task 1.2, Functional requirements, and task 1.3, System architecture and technical requirements focus on the identification and description of functional requirements for each technology module and the system, namely the description of functional behavior, relating input and output data/results with operations and workflow of the system according to certain operational conditions. These high-level requirements are the basis of communication between the developers and are important to ensure the satisfaction of end-user needs. Understanding these requirements before implementation and determining the initial scope of work ensures consistency with the agreed vision, further enabling the assessment of overall quality and identifying any required changes.

1. The Architecture in a nutshell

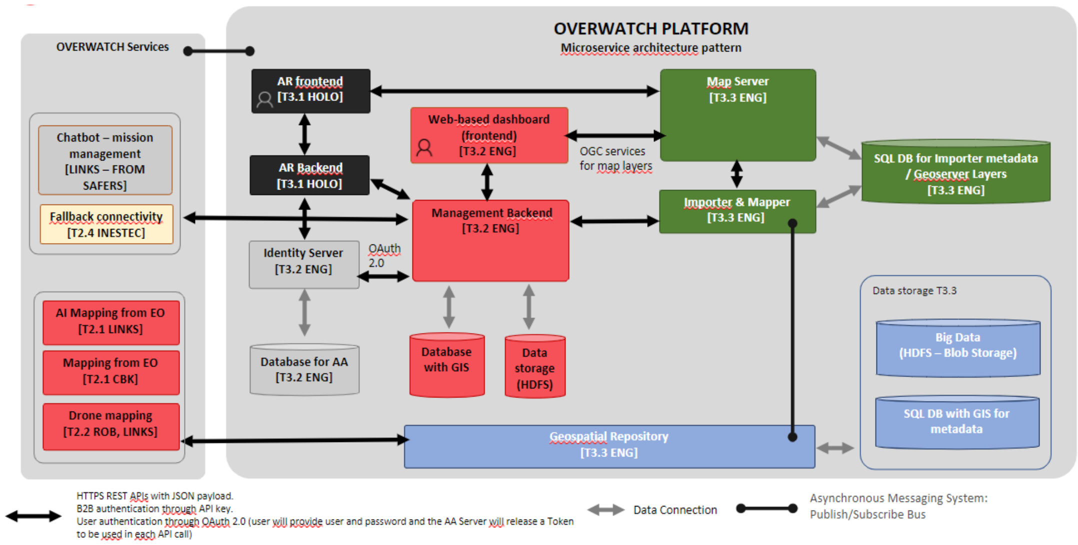


Figure 1 - The OVERWATCH architecture in a nutshell.

The architecture of the system is divided in two parts: the first provides useful data based on imagery that comes from satellite Earth observation or from drone acquisition or from legacy systems, the second gathers the information and serves the results to the final user.

The GIS layers and images are the primary data source of the architecture, therefore they require a specific standardized pipeline. First, they need to be ingested, using a Big Data storage solution to receiveand store this data through the Geospatial Repository that, in event-driven way, will share the signalling of the operation using a message broker, maintaining a loose coupling among the components that provide data from the others.

Once images and other GIS-enabled data have been stored, the underlying storage infrastructure triggers the processing phase that performs required operations to import and harmonize the data into map layers.This will provide a mechanism for cataloguing and querying the imported data together with metadata information, exploiting OGC-compliant services such as GeoServer[2].

The delivery of this information is possible thanks to the management system able to query and visualize the content that has been imported,through OGC services that provide the capability to serve and deliver maps to clients or end-users in a format that can be easily rendered and displayed on devices or applications such as AR and web dashboard.

The two parts that comprise the architecture are connected to each other with technologies and mechanisms that also foresee a fallback connectivity. In operational scenarios, the network connectivity is a crucial component to ensure the correct interactions between modules. However disruptions or failures can occur due to various reasons, such as network outages, hardware failures, or environmental factors above all, especially in crisis situations. By considering the possibility of such disruptions, system designers incorporate backup or alternative connectivity options to mitigate the impact on the overall system functionality. The symbol  represents the presence of a message bus that will allow to exchange all the metadata between the services and the backend and all the updates that will be forwarded from the services to the geospatial repository. The message bus it's a standard component in the architecture and is based on a publish/subscribe mechanism

[2] GeoServer is an open source server for sharing geospatial data, https://geoserver.org/

Table 1 - Components and layers of the architecture involved.

Applications (based on D1.1)	Module name	Owner	Layer of the architecture
Artificial intelligence-based backend management system (AIMS)	Web Based dashboard	ENG	Presentation layer
	Management Backend	ENG	Data, Integration and Business layer
Earth observation (EO)	Mapping from EO	CBK	Service Layer
	AI Mapping from EO	LINKS	Service Layer
Augmented Reality (AR)	AR Backend	HOLO	Backend
	AR Front end	HOLO	Presentation layer
Drones (DRS)	Drone terrain mapping	ROB	Service Layer
Fallback communication system (FCS)	Fallback Connectivity	INESC	Infrastructure Layer

The listed layers are a logical grouping of components or modules that perform a specific set of functions and provide a well-defined interface to other layers or components. They are used to structure the entire architecture in a way that promotes modularity, scalability, and maintainability. The layers are divided into several parts, such the Presentation Layer, composed by the Web Based Dashboard and AR Frontend, Service Layer such as EO and Drones Mapping modules, Infrastructure Layer such as Connectivity and Fallback Connectivity. Finally, the Management Backend oversees Data, Integration and Business Layer.

2. Technical modules description and functional requirements

2.1. Management Backend

2.1.1. Brief description of the tool/module

The module represents the core processing of the system. Indeed, it allows the connections among the presentation layer module like the Web based dashboard or other backend modules like the AR backend. For these two modules it allows to present all the information processed by the services to the end users that will use the dashboard. It will also enable production of all the needed information for the Augmented reality back end. Moreover, the backend will allow all the users to access the whole OVERWATCH system through RBAC[3] paradigm providing the right authorizations and hence guaranteeing the navigation sessions to the users through a single sign-on system. Essential subsystems of the management backend are the geospatial repository, the map server and the importer and mapper . Going to the information hub, namely the geospatial repository subsystem, this is connected through a message broker to the services of the service layer that will produce raster images, geoJSON information, metadata on the current situation. This information will be stored according to the necessary topology of the information: on HDFS in case of files and on noSQL database for what concerns the metadata related to the situation and on the orthorectified images. All this information will be then imported into the system through the Importer and Mapper subsystem that will allow to save the different layers representing the different conditions of the terrain in case of flood and fires on the map server subsystem.

2.1.2. Expected users of the system

Emergency responders: tactical management, operational experts using the *Command and Control* system. The software will be deployed as a docker image to be used in the control room of the end users and the modalities (on cloud, on premises, etc) will be defined in the piloting phase. All the end users will have internet connectivity and in case of network problems access to the fallback connectivity module, enabling them to use OVERWATCH services. The data will come to the backend management system through the other services and not from the users.

2.1.3. Description of capabilities of the tool/module

The backend capabilities are described below:

- Storage of the information in the appropriate database.
- Serving information to the AR backend.
- Serving information to the Web based dashboard.
- Management of the user and authentication and authorization
- Mapping of the data fused by the EO and the drone mapping module with the orthorectified images.

2.1.4. Expected technical inputs and expected outputs

Supporting Situation Awareness with following expected I/O:

- Static
    - Provide map system for current situation forwarding a secured channel for providing map tile services.
- Dynamic
    - Image processing for enhanced map historical situation for pre-analysis of the state of the infrastructure.
    - Data Model support to the event and information useful for the operators to better understand the critical situation, getting the AI-powered data analytics actionable information (from Web Dashboard and from AI).

Table 2 - Management backend inputs and outputs for the current situation

Type of information	Information

[3] We will use Keycloak as Identity Server, and Keycloak supports fine-grained authorization policies and is able to combine different access control mechanisms such as RBAC (https://www.keycloak.org/docs/latest/authorization_services/index.html)

Format	Imagery data format
Input payload draft/final	Query params for layer selection Eg. : https://<host>/<params>/{z}/{x}/{y}.png
Output payload draft/final	Content-Type: image/png
Protocol	HTTP GET Method

Image processing for enhanced map historical situation for pre-analysis of the state of the infrastructure consists of a trigger from final user. Before this the images that come from the sources need to be stored.

Table 3 - Management backend inputs and outputs for image processing

Type of information	Information
Format	JSON, async communication

Input payload draft/final	{
	"type": "drone-mapping"|"eo-mapping",
	"UUID": "3b32af91-c98e-4119-aff0-2042262b5c54",
	"start": "2022-06-21T00:00:00.000Z",
	"end": "2023-06-21T00:00:00.000Z",
	"feature": {
	"type": "Feature",
	"geometry": {
	"type": "Polygon",
	"coordinates": [
	[
	[100.0, 0.0],
	[101.0, 0.0],
	[101.0, 1.0],
	[100.0, 1.0],
	[100.0, 0.0]
	]
	]
	},
	"properties": {
	"prop0": "value0",
	"prop1": {
	"this": "that"
	}
	}
	}

Output payload draft/final	{ "requestId": 36001, "UUID": "3b32af91-c98e-4119-aff0-2042262b5c54", "type": "drone-mapping", "status": "ok|processing|done", "message": "Request has been accepted", "path": "<path REST/WMS/Tile Server,only on done>"

	}
Protocol	HTTP (method POST for creation, method GET for querying)

Table 4 - Management backend inputs and outputs for requests form services

Type of information	Information
Format	POST request multipart/form-data

Input payload draft/final	file: <the file> content type image/*
	data: content type application/json
	{
	"UUID": "3b32af91-c98e-4119-aff0-2042262b5c54",
	"type": "drone-mapping"
	}

Output payload draft/final	{
	"requestId": 36001,
	"UUID": "3b32af91-c98e-4119-aff0-2042262b5c54",
	"type": "drone-mapping",
	"ETAG": "33a64df551425fcc55e4d42a148795d9f25f89d4",
	"status": "ok",
	"message": "Request has been accepted"
	}

Protocol	HTTP (method POST, multipart form)

The following I/O is a generic Data Model support to the events and respective information gathered by the AI-powered data analytics and the Web Dashboard.

Table 5 - Management backend inputs and outputs for representing information on events

Type of information	Information
Format	JSON

Input payload draft/final	{
	<generic data model attributes>,
	"feature": {
	"type": "Feature",
	"geometry": {
	"type": "Polygon|Point",
	"coordinates": [
	[
	[100.0, 0.0],
	[101.0, 0.0],
	[101.0, 1.0],
	[100.0, 1.0],
	[100.0, 0.0]
	]
	]
	},
	"properties": {
	"prop0": "value0",
	"prop1": {

	"this": "that" } } } }}

Output payload draft/final	{
	"id": "<data-model id from DB>",
	}

Protocol	HTTP REST API CRUD (by the HTTP Method: Create/POST Read/GET Update/PUT-PATCH Delete/DELETE)

2.1.5. Available interfaces

The available interfaces are REST interfaces. They are based on a set of principles and constraints that make it easy to build scalable and interoperable systems, client and server are separate entities that communicate over a network, each request from a client to a server must contain all the necessary information for the server to understand and process the request and the interfaces between the client and the server are standardized and consistent.

2.1.6. Mapping of the functional requirements and the end user requirements within the technical module

In the Table 6 are mapped the functional requirements of the module with the user requirements defined in D1.1.

Table 6 - Functional requirements for management backend

IDENTIFIERS	FUNCTIONAL REQUIREMENTS	END USER REQUIREMENTS	DESCRIPTION END USER REQUIREMENTS
AIMS.FR.1	The AIMS manages and ingests AI algorithm results	AIMS-M-000	The AIMS must manage the OVERWATCH system using artificial intelligence algorithms, e.g., to analyse historical/current data, identify patterns and trends relevant to improve risk assessment and decision-making
AIMS.FR.2	The AIMS supports the C2 into providing information on wildfires and floods	AIMS-M-010	The AIMS must support the C2 in managing emergencies along the response and recovery phases by providing information related to wildfires and floods
AIMS.FR.3	The AIMS will support in the assessment and update of the SA by providing alerts and	AIMS-M-020	The AIMS must have the ability to continuously assess the TO and update the SA of the C2 decision-makers and other emergency responders

	information on current situation	AIMS-S-010	The AIMS should be able to provide real-time updates and alerts to emergency responders based on changing conditions
		AIMS-S-060	The AIMS should be able to provide early warning alerts to emergency responders based on detected risks
		AIMS-S-070	The AIMS should provide situational awareness to residents in the TO or possible emergency evolution areas, allowing them to make informed decisions about evacuation
AIMS.FR.4	The AIMS stores the information to manage the allocation of resources for managing tasks during the emergency	AIMS-M-030	The AIMS must facilitate and support C2 in allocating resources (personnel, equipment, supplies, etc.) based on, e.g., severity, the potential spread of the natural hazard and the impact on human life and property
AIMS.FR.5	The AIMS stores the necessary information and supports the management and coordination of the usage of UAV	AIMS-M-040	The AIMS must support managing/coordinating the operation and use of autonomous vehicles in the environment, e.g. the missions and airspace used by drones
AIMS.FR.6	The AIMS ingests and manages satellite imagery	AIMS-M-060	The AIMS must access and ingest satellite imagery and other remote sensing data to supplement on-the-ground data collection efforts
AIMS.FR.7	The AIMS ingests Copernicus EMS data elaborated by the AI mapping for EO	AIMS-M-070	The AIMS must collect and analyse data from the Copernicus EMS
AIMS.FR.8	The AIMS ingests metadata on resources to provide information to ERs	AIMS-M-120	The AIMS must be able to identify resources such as water, fuel, roads, or alternative routes to escape danger and emergency progression, providing valuable information to emergency responders
AIMS.FR.9	The AIMS integrates existing emergency	AIMS-M-130	The AIMS must support integration with existing

	management systems storing the information that will be visualized by the SA		emergency management systems
		AIMS-C-050	The AIMS could support visualising the results of modelling the development of the emergency provided by third-party services
AIMS.FR.10	AIMS integrates external sources and data sources	AIMS-S-000	The AIMS should be able to integrate with multiple external sensors and data sources to enhance data collection and analysis
AIMS.FR.11	AIMS prepares data for visualization on the Web based dashboard and AR backend	AIMS-M-150	The AIMS must be able to provide real-time situational awareness to emergency responders
AIMS.FR.12	AIMS dispatches information for reports and visualizations to the web-based dashboard	AIMS-M-160	The AIMS must be able to generate reports and visualisations that help emergency responders make informed decisions quickly
		AIMS-M-170	The AIMS must be able to generate reports and visualisations for post-action review and lessons learned
		AIMS-M-180	The AIMS must provide situational awareness to all emergency responders, enabling them to make informed decisions in real-time and coordinated responses
AIMS.FR.13	AIMS allows to share information with ERAs	AIMS-M-200	The AIMS must have the capability to share data and information with other emergency response agencies and organisations
AIMS.FR.14	AIMS supports the SA with the visualization on air quality monitoring	AIMS-C-020	The AIMS could have the capability to provide real-time air quality monitoring for responders and the public
AIMS.FR.15	AIMS models the development of the emergency	AIMS-C-060	The AIMS could model the development of the emergency under different conditions to provide SA support in identifying effective routes for emergency

			responders to access/leave the affected area
AIMS.FR.16	AIMS supports SA to provide recommendations	AIMS-C-080	The AIMS could be able to provide recommendations to emergency responders on the most effective response strategies

2.2. Web-based dashboard

2.2.1. Brief description of the tool/module

The Web-based dashboard's main aim is to visualize all the information needed to better understand and handle an emergency by collecting, normalizing, and presenting data from all heterogeneous sources (e.g., involved operators, drones, assets, sensors, and AR devices). A Dashboard must be provided to the incident commanders operating at the command center. Leveraging on the inputs coming from the OVERWATCH components, the dashboard provides a detailed picture of a current situation, even updated in real-time, in the area affected by the hazard, mitigating the consequences of a disaster more efficiently.

By using the Geographic Information System (GIS) and in collaboration with other OVERWATCH modules, the dashboard provides the current positions of all engaged subjects (involved operators, drones, AR users, assets), activity status, and collected data.

Each item on the map is presented using descriptive visual elements (e.g., colour, icon, animation), making it easier to understand their type and activity.

A new critical area with all specific details can be defined on the map by the commander and visualized by all users. Processing and visualizing data collected such as images, videos, and laser measures, gives a detailed overview of the consequences of the disaster, and a better starting point for quality mission planning.

Also, the end-user can create a new mission, assign it to selected team member or a drone, modify the details of existing one, update status of the mission, send and inspect messages, inspecting missions, reports, etc.

2.2.2. Expected users of the system

Political and strategic decision-makers will have access to the software. Emergency responders, who are experts in tactical management and operations, will use the Command and Control system. The software will be deployed as a docker image in the control room of the end users.

2.2.3. Description of capabilities of the tool/module

The primary purpose of Web based dashboard is enhancing situational awareness, improving, and extending information sharing and visualization, decision support, and communication into any operational environment within the emergency management.

The combination of detected critical events, alerts, available infrastructures, and resources, first responders' location, images, videos from the scene, and other relevant information is visually communicated in an easy-to-use way. The main features are:

- **Interactive Map**: the map is the place where all the data containing any geo-localized information is visualized All subjects engaged in a specific mission are presented on the map at their current location. Also, predefined, and additionally created critical zones in the form of polygons are shown on the map. Reference to a specific level of severity is presented with the use of different colours so that the user can have essential information about the current situational picture, providing a rapid overview of entire system with a looking style dashboard.

- **Filter View**: the dashboard offers the possibility for the user to filter the data presented on the map. With a single click, only the subjects of the desired type are presented on the map. Since the dashboard can be used for monitoring multiple critical situations in different locations, using the list of all ongoing missions, the commander can easily switch between them, and map focus will fly to the selected area.
- **Location and Information**: Besides the current location of all subjects on the map, the commander can quickly get detailed information about their activity, status, collected data, etc. When a specific subject is selected, the mix of data collected, and Geolocation information is presented on the information panel. The type of information differs related to the subject type. For example, in the case of drones, collected multimedia data is presented (e.g., videos, images). By selecting any item on the map, the dashboard calculates the distance between selected and closest ones (distance limit must be defined) and creates a list ordered by distance with graphically visualized relations between them.
- **External Geoserver Layer:** the dashboard has support for including additional raster layers on the top of already existing ones. The user can manage multiple extra layers. Including additional information can enhance and enrich the current situational picture using historical or statistical data.
- **Visualize and manage the critical Area:** The size and severity of the area affected by the disaster is something that is being changed over time. Before the beginning of the mission, the boundaries of the disaster scene must be defined. Using the drawing tool, user can set the area on the map in the form of a polygon, also with the possibility to add information related to zone describing the current situation (e.g., zone name, severity, locality name, hazard type). Starting from the polygon, a request to the underlying services will be sent to trigger a request mapping to retrieve information from EO and drone observation modules.
- **2D/3D orthomosaic Mapping:** In case of disasters, the involved operators need to have a detailed landscape overview. To achieve that, the user can add 2D or 3D orthomosaic scans of a specific area on top of the existing map layers.
- **User management** is inherited from the Identity Management module, in order to create, delete, update users

2.2.4. Expected technical inputs and expected outputs

Visualize and manage the Situation Awareness with following expected I/O:

1. Static
   - Visualize static layers which can enhance and enrich the current situational picture using historical or statistical data.
2. Dynamic
   - Visualize information in the map for pre-analysis of the state of the infrastructure.
   - Visualize the information useful for the operators to better understand the critical situation.
   - Adding and editing the map information for enrich the situation awareness.

Table 7 - Web based dashboard inputs and outputs for static layers.

Type of information	Information
Format	Imagery data format

Input payload draft/final	Query params for layer selection Eg. : https://<host>/<params>/{z}/{x}/{y}.png
Output payload draft/final	Content-Type: image/png
Protocol	HTTP GET Method / WMS

Visualize information in the map for pre-analysis of the state of the infrastructure. In the table are represented the technical specifications of the modules.

Table 8 - Web based dashboard inputs and outputs for the visualization of the information.

Type of information	Information
Format	JSON, async communication

Input payload draft/final	`{`
	`"type": "Asset Data”,`
	` Id": "<data-model id from DB "`
	`}`

Output payload draft/final	`{`
	`    "type": "Asset Data”,`
	`    "UUID": "3b32af91-c98e-4119-aff0-2042262b5c54",`
	`    "start": "2022-06-21T00:00:00.000Z",`
	`    "end": "2023-06-21T00:00:00.000Z",`
	`    "feature": {`
	`        "type": "Feature",`
	`        "geometry": {`
	`            "type": "Polygon",`
	`            "coordinates": [`
	`                [`
	`                    [100.0, 0.0],`
	`                    [101.0, 0.0],`
	`                    [101.0, 1.0],`
	`                    [100.0, 1.0],`
	`                    [100.0, 0.0]`
	`                ]`
	`            ]`
	`        },`
	`        "properties": {`
	`            "prop0": "value0",`
	`            "prop1": {`
	`                "this": "that"`
	`            }`
	`        }`
	`}`

Protocol	HTTP (method POST for creation, method GET for querying)

Visualize the information useful for the operators to better understand the critical situation. In the table are represented the technical specifications of the modules.

Table 9 - Web based dashboard inputs and outputs for the representation of features on the map

Type of information	Information
Format	JSON, async communication

Input payload draft/final	{
	"id": "<data-model id from DB>",
	}

Output payload draft/final	{
	<generic data model attributes>,
	"feature": {
	"type": "Feature",
	"geometry": {
	"type": "Polygon|Point",
	"coordinates": [
	[
	[100.0, 0.0],
	[101.0, 0.0],
	[101.0, 1.0],
	[100.0, 1.0],
	[100.0, 0.0]
	]
	]
	},
	"properties": {
	"prop0": "value0",
	"prop1": {
	"this": "that"
	}
	}
	}
	}}

Protocol	HTTP (GET for querying)

Adding and editing the map information for enriching the situation awareness. In the table are represented the technical specifications of the modules.

Table 10 - Web based dashboard inputs and outputs for enriching the situation awareness.

Type of information	Information
Format	JSON

Input payload draft/final	{
	<generic data model attributes>,
	"feature": {
	"type": "Feature",
	"geometry": {
	"type": "Polygon|Point",
	"coordinates": [
	[
	[100.0, 0.0],

	[101.0, 0.0],
	[101.0, 1.0],
	[100.0, 1.0],
	[100.0, 0.0]
	]
	]
	},
	"properties": {
	"prop0": "value0",
	"prop1": {
	"this": "that"
	}
	}
	}
	}}

Output payload draft/final	{
	"requestId": 36001,
	"UUID": "3b32af91-c98e-4119-aff0-2042262b5c54",
	"type": "",
	"status": "ok|processing|done",
	"message": "Request has been accepted",
	"path": "<path REST/WMS/Tile Server,only on done>"
	}

Protocol	method POST

2.2.5. Available interfaces

The Dashboard communicates directly with the Management Backend module. The communication between the two components occurs through the REST API Services.

The Dashboard uses Axios to create HTTP requests. It is a promise-based lightweight HTTP client, straightforward and flexible for configuring (e.g., base URL, headers, interceptors).

The Dashboard can also include external layers hosted on WMS server. Using the data provided by a GIS database, the map server produces georeferenced map images serving them using Web Map Service (WMS) protocol.

2.2.6. Mapping of the functional requirements and the end user requirements within the technical module

In the table 11 are mapped the functional requirements of the module with the user requirements defined in D1.1.

Table 11 - Functional requirements for Web based dashboard

IDENTIFIERS	FUNCTIONAL REQUIREMENTS	END USER REQUIREMENTS	DESCRIPTION END USER REQUIREMENTS
AIMS.FR.1	The user can set the area on the map in the form of a polygon, also with	AIMS-M-190	The AIMS must incorporate advanced drawing, marking and

	the possibility to add information related to zone describing the current situation. In the map, the user can select any item and the dashboard calculate the distance between selected and closest one.		measuring functions through the AR interface
AIMS.FR.2	The Dashboard provides multi-language support.	AIMS-S-220	The AIMS must be able to provide multi-language support, e.g., English, Portuguese or Polish, for ER from different member states
AIMS.FR.3	The dashboard has support for including additional raster layers on the top of already existing ones. This layer (heat maps and other visualisations) including additional information can enhance and enrich the current situational picture	AIMS-S-020	The AIMS should be able to generate heat maps and other visualisations that support ER in understanding the spread of emergencies and critical areas
AIMS.FR.4	The dashboard visualize in map a specific level of severity with the use of different colours so that the user can have essential information about the current situational picture.	AIMS-S-040	The AIMS should be able to support prioritising response efforts based on the potential impact on human life, property, and the environment
AIMS.FR.5	The dashboard is based on a combination of the latest technologies with optimized speed, scalability, and deployment on multiple platforms. Each of them is carefully selected based on mutual compatibility to	AIMS-S-050	The AIMS should provide remote access to emergency responders to allow them to access the system from their own devices, e.g., providing lightweight UI for inputting data, dashboards or chat boards

	ensure maximum performance.		
AIMS.FR.6	The dashboard can be used for monitoring multiple critical situations in different locations, using the list of all ongoing missions, the commander can easily switch between them, and map focus will fly to the selected area.	AIMS-C-090	The AIMS could be able to provide situational awareness for multiple incidents in different locations
AIMS.FR.7	The dashboard could provide the optimal route for waterline laying in unfamiliar TO as an additional layer.	AIMS-C-100	The AIMS could provide the optimal route for waterline laying in unfamiliar TO, namely the water supply line set along a road and composed of hoses, fire engines and transfer tanks, according to the DSM

2.3. Mapping from EO

2.3.1. Brief description of the tool/module

The module consists of various EO-based products which have been identified as the most useful by the end-users. Its scope includes monitoring services and on-demand products.

The service is one of the modules offered by T2.1 - "AI algorithms exploiting Copernicus EMS EO data and in-field acquisitions: training, testing, and operationalization". Leveraging on Sentinel data, this component will provide an automated pipeline to process and analyse EO data, in a given area of interest and time frame. The analysis will comprise the production of several maps, from wildfire delineation to damage assessment over different hazard types.

The monitoring services will provide timely information, as soon as the processed EO data are available, e.g., continuous (near real-time) surface water extent in the form of GIS layers. They are aimed to be used in the prevention and preparedness crisis management phases, as well as in disaster risk reduction activities.

The products, delivered upon request, will be defined, and further personalized according to the end-users' needs, exploiting AI-based tools for the generation and a team of geospatial analysts for ground truth production and validation. The maps may depict specific details such as the impact of the event on infrastructures, allowing for damage assessment, or provide other advanced analyses (e.g., burned area delineation) based on the most recent and historical satellite images, and other GIS data. Ground truth data will be imported from the Copernicus EMS On-Demand Mapping or provided by CIK CBK PAN and ITHACA. Operational layers will be generated by AI services (LINKS) in the form of layers ready to be displayed in OVERWATCH system. The aim of this type of mapping products is to increase situational awareness during and in the immediate aftermath of a disaster as well as outside the response phase.

2.3.2. Expected users of the system

The EO mapping module is expected to be an automated and detached service, which only communicates via the project-level message bus (WP3). From this bus, users will be able to send map request messages, provided in JSON format, through the main dashboard and other UI tools, which in turn rely on the central backend (WP3). The service will not require prohibitive computational requirements; however, a GPU-enabled physical or virtual machine is certainly expected to deal with the machine learning models. This machine will also require a high-speed internet connection, to download large quantities of EO data through third-party web services, as well as substantial on-board storage, to be able to cache this data for the AI processing part.

Considering the expected user base, this service is envisioned for expert users, i.e., first responders and decision makers, since the provided outputs will require a certain degree of expertise to be interpreted. Nevertheless, given its use will be ultimately decided by the access control rules defined in the backend and the communication only happens through the message bus, the availability of this service could be easily expanded to other user groups with little effort. The expected end-users of the module are First responders – Civil Protection and Emergency Response teams, On-Site Operations Coordination Centres, Crisis centres, monitoring and supervisory teams, analytics sections and other management units of various Civil Protection and Emergency Response actors and the Critical Infrastructure operators.

2.3.2.1. Input required from the end users

In terms of inputs, the service shall only require an input message from the message bus to be activated. This message must contain a set of minimum required information to be able to automatically map a specific area, namely:

- **Mapping type**: ideally, a unique identifier to unambigously determine which kind of algorithm needs to be applied to the requested mapping. This also determines which EO images should be retrieved (e.g., Sentinel-2 in case of wildfire delineation).
- **Time interval**: a specific time frame is required to map with any given algorithm. This is necessary to identify the period in which the service should look for available and valid input images. While the availability is given by the satellite revisit times (e.g., 1-3 days for Sentinel-1, 5-6 days for Sentinel-2), the validity of the data itself relies on the specific source. For instance, an excessively cloudy S2 image cannot be successfully used for burned area delineation. The time interval shall at least be identified with a *start* and *end* timestamp. Additionally, an *event time* could be provided, to carry out a pre/post comparison.
- **Input geometry**: this field describes the area to be mapped by the service. Despite allowing for complex shapes, given the outputs comprising raster images, the polygons will be converted into axis-aligned bounding boxes, using the bounds of the polygons as coordinates.
- **Mapping frequency (optional)**: optionally, the user can provide a fixed amount of time between periodical mappings, expressed in days. This field can enable periodical mappings (i.e., repeating the same EO mapping every N days where N is the frequency). It indicates how many days will elapse between subsequent activations of the mapping service for the given area. The provided time interval will be divided into many sub-requests for the actual one-shot execution. The start and end limits of the periodical mappings can potentially be past, present, or future: the service will have the task to schedule each sub-request for immediate or deferred execution.
- **metadata**: any other kind of information that should be mirrored in the response (e.g., some identifier to keep track from the exterior) shall be inserted in a JSON dictionary, stored

alongside the map request. In this case, no validation is provided to the inputs, and they can be exploited for compatibility with other data sources or other particular use cases.

2.3.3. Description of capabilities of the module

The module will include several monitoring services, including: CBK Floods and Global Flood Monitoring (GFM) service, automated AI-based mappings (LINKS), and additional Copernicus EMS On-Demand Mapping products, together with layers delivered on-demand by the CIK CBK PAN and ITHACA specialists, to serve as both output and support ground truth to AI-based tools.

The CBK Floods service has been developed by the Space Research Centre PAS. The service generates current surface water cover over Poland. The automatic detection algorithm uses Copernicus Sentinel-1 satellite radar images which are provided regularly. The map is updated after each passage of the satellite and shows different stages of inundation: new water extent, areas with long-lasting water and those from which water has receded in the last days.

The Global Flood Monitoring (GFM) was integrated into Global Flood Awareness System (GloFAS), which is a component of the Copernicus Emergency Management Service (CEMS). GFM computes on a regular basis 11 different flood-related products, two of which may be used in the OVERWATCH system:

1. Observed Flood Extent - Flooded areas mapped using Sentinel-1 SAR backscatter intensity,
2. Observed Water Extent - Open and calm water mapped as the union of the output layers Sentinel-1 Observed Flood Extent and S-1 Reference Water Mask.

The Copernicus EMS On-Demand Mapping products are produced manually and provided within hours or days of Copernicus service request. They will also be made available in OVERWATCH system upon request based on territory coverage. The products delivered on-demand by the CIK CBK PAN and ITHACA will be developed on a case-by-case basis in accordance with user requirements. They may be based not only on EO, but also drone mapping. The procedure for their preparation will be established and tested within OVERWATCH.

AI-based services, provided by LINKS, shall include several on-demand layers based on end users' needs and the processing capabilities from EO imagery. These layers will be automatically computed within minutes from the request, provided that a valid area of interest and time range are provided, and a satellite feed is available. These layers will include maps for several hazard types including wildfires, floods and landslides, producing outputs such as burned area delineations, burn severity estimates, flood and landslide delineations, land cover and fuel maps. These outputs will be generated from machine learning methods from Sentinel imagery (mainly Sentinel-1 and Sentinel-2), and trained on available EO datasets, Copernicus EMS maps and on-demand products (e.g., CBK, ITHACA). Outputs will be provided in GeoTIFF or GeoJSON format and uploaded onto the Geospatial repository to be imported and converted into OGC-compliant layers for use on the OVERWATCH platform.

2.3.4. Expected technical inputs and expected outputs

Table 12 - Mapping from Earth Observation imagery request

Type of information	Information
Format	JSON

Input payload draft	`{`
	`"datatype_id": "unique_ID"`
	`"start": "2022-06-01T00:00:00.000Z",`
	`"end": "2022-06-30T00:00:00.000Z",`

```
        "frequency": 5,
        "geometry": {
            "type": "Polygon",
            "coordinates": [
                [
                    [
                        0.41988372802734375,
                        41.077539547047294
                    ],
                    [
                        0.5359268188476562,
                        41.077539547047294
                    ],
                    [
                        0.5359268188476562,
                        41.15539335830357
                    ],
                    [
                        0.41988372802734375,
                        41.15539335830357
                    ],
                    [
                        0.41988372802734375,
                        41.077539547047294
                    ]
                ]
            ]
        }
    }
```


Output payload draft	{ "datatype_id": 36001, "status_code": 200, "type": "start"|"end"|"update" ... "name": "Example request" "message": "Request Completed Successfully", "urls": ["https://...", "..."], "metadata": {...} }
Protocol	HTTP, TCP, AMQP (depending on the message broker adopted in WP3). Uploaded outputs will be stored in GeoTIFF or GeoJSON format.

Table 13 - CBK Floods (surface water extent)

Type of information	Information
Format	.tif
Input	CBK-Floods_YYYY-MM-DDThh-mm-ss (Sentinel 1 observed flood/water extent)
Output	CBK-Floods_YYYY-MM-DDThh-mm-ss (Sentinel 1 observed flood/water extent)

Protocol	tbd
Elapsed time	Near real time
Prerequisites	Network connection availability; Copernicus data availability
Comments	Surface flood extent by CBK Floods service is provided for the territory of Poland. New layer is available 1 or 2 times a day. Currently, due to only one satellite operating instead of two, on same days there is no new data layer at all. The layer covers the entire area of Poland, but for a specific date only part of it provides new updated information on water extent and this can be read from the legend. The area is classified into few classes. Files can be searched for and imported through self-written APIs in OpenAPIv3 standard or visualised by API WMS. Instruction can be found here: https://dane.sat4envi.imgw.pl/howto (google translation into English needed).

Table 14 - Global Flood Monitoring (GFM)

Type of information	Information
Format	.tif / .shp
Input	GIS layers - Sentinel 1 observed flood extent (Copernicus EMS Service)
Output	GIS layers - Sentinel 1 observed flood extent (Copernicus EMS Service)
Protocol	tbd
Elapsed time	<8 hours after S-1 image acquisition, for the specific region new S-1 data are available at least each 12 days (current status for 1 satellite, 2nd satellite will improve revisit time to 6 or less days)
Prerequisites	Network connection availability; Copernicus GFM service availability
Comments	...

Table 15 - Automated AI-based mapping from EO imagery

Type of information	Information
Format	.tif, .shp, GeoJSON
Input	JSON map request
Output	GIS images and layers (vector / raster)
Protocol	tbd
Elapsed time	Few minutes – < 1 day
Prerequisites	Network connection availability; Satellite data availability; computing resources availability.
Comments	Automated mapping done through AI-basd tools

Table 16 - Copernicus EMS On Demand Mapping

Type of information	Information
Format	.tif, .shp, GeoJSON, .jpeg, .pdf, KML, KMZ
Input	map(s) and GIS layers (vector / raster)

Output	map(s) and GIS layers (vector / raster)
Protocol	tbd
Elapsed time	2 hours – 15 days, depends on the time of EO data acquisition and other data availability, depends on product type
Prerequisites	Network connection availability; Copernicus EMS products availability
Comments	

Table 17 - On-demand mapping (by CIK CBK PAN)

Type of information	Information
Format	.tif, .shp, .jpg
Input	map(s) and GIS layers (vector / raster)
Output	map(s) and GIS layers (vector / raster)
Protocol	tbd
Elapsed time	2 hours – 5 days, depends on the time of EO data acquisition and other data availability
Prerequisites	Network connection availability; Copernicus and other data availability
Comments	Input data needed for maps and GIS layers preparation (the process will be transparent for the OVERWATCH system): EO remote sensing data; optionally other data, e.g. DEM (.asc / .tif / .xyz), OpenStreetMap (.shp), Topographic Database BDOT10k (.shp / .gml)

2.3.5. Available interfaces

The EO mapping service will mainly communicate through the message bus only, using JSON messages. Alternatively, a simple RESTful API will be provided, at least for what concerns the map request inputs, since incoming requests cannot be blocking. The I/O interface is therefore bound to the implementation choice for the message bus, i.e. AMQP for RabbitMQ, or TCP streams for Kafka.

2.3.6. Mapping of the functional requirements and the end user requirements within the technical module

In the Table 18 are mapped the functional requirements of the module with the user requirements defined in D1.1.

Table 18 - Functional requirements for Mapping from Earth Observation module

Identifiers	Functional requirements	End user requirements	Description End user requirements
EO-1	The system shall collect multiple EO data sources (EMS, GIS data, OSM).	EO-M-000 EO-M-010 EO-M-020 EO-S-050	The tool shall be able to gather and cross-reference data from different sources
EO-2	The system shall provide a range of thematic layers for different hazard types (fire, flood, damage assessment)	EO-M-000 EO-M-020 EO-S-010 EO-C-050	The tool shall be able to provide different output mappings.
EO-3	The service should support change detection using EO for different purposes (land cover,	EO-S-020 EO-S-040	Similarly, the different mappings may exploit pre- and

	land use, environmental changes that could exacerbate hazards)		post-hazard imagery for a more precise output.
EO-4	The service should enable mappings on historical data and past satellite acquisitions.	EO-C-000	The module shall be able to retrieve historical data and provide the same mappings.
EO-5	The service may provide a precise delineation and localization of hazards and their current status	EO-C-030	While the delineation and subsequent localization are certainly feasible, the current status depends on the availability of the satellite images.
EO-6	The system may provide useful information for evacuation and other emergency measures based on EO	EO-S-000	Tthe detection of affected infrastructures could be provided using EO data after the event occurred, considering the limitations of the temporal and spatial resolution of the considered satellite network (eE.g. for sentinel-2 10m and 5-6 days). Such detection could be taken into account together with other local information (e.g. road closures) to compute the evacuation route. The implementation of a service aimed at the provision of the evacuation route given the satellite mapping is out of scope in OVERWATCH.
EO-7	The system may provide alerts in case of detected emergency or escalation.	EO-S-030	Typically, alerts are issued according to the forecasted weather conditions in the early warning phase, while EO can provide detection capabilities

			(response or post-disaster phase). Therefore, we cannot implement any early warning alerting using EO data. Other instruments such as MODIS and VIIRS may be evaluated for this purpose.
EO-8	The service could provide ad-hoc layers to identify the potential risks (e.g., landslide, wildfire, etc.)	EO-C-010 EO-C-020	The feasibility of this requirement needs to be assessed for each kind of hazard (wildfire, landslide).

2.4. AR Backend

2.4.1. Brief description of the tool/module

This module will provide the application and its associated logic which enables the remote rendering and generation of three-dimensional Augmented Reality holographic content of the information produced by the OVERWATCH system. This module will receive information from the other modules of the OVERWATCH system, namely the backend management (see section 2.1). The module here will consist of a Unity application and will be enabled with the Interactive Streaming for Augmented Reality (ISAR) Unity plugin which allows remote rendering and whole application streaming to dedicated AR devices for three-dimensional visualization (e.g., HoloLens2). The client application present in the latter is also referred to as the AR Frontend, which will be described below. This module is also referred to as the "HoloServer". The application will run on a Windows environment with the required GPU processing. The HoloServer will interface with the OVERWATCH modules to receive processed information and send back relevant information to the OVERWATCH modules if needed. Importantly, this module will produce and send the remote rendered content/application pixels streams to the AR client device, and in return receive input data from the AR device (e.g., head pose, etc.). The connection between the AR backend and its AR client (AR Frontend) device through ISAR streaming will be mediated by WebRTC peer-to-peer connection via WiFi through an open TCP port 9999.

2.4.2. Expected users of the system

The users of this system consist of the following:

- Strategic decision-makers and tactical management and operational decision-makers who are located in the command centre.
- OVERWATCH researchers/personnel who are envisioned to execute the application on a physical station (or initiate the instance if on virtual machine or cloud-based).

The tactical management and operational decision-makers are the users who wear the AR devices and provide the user input data to this application and therefore interact with the 3D content produced by it.

2.4.3. Description of capabilities of the tool/module

The purpose of this module is to generate the AR visualization of the information processed by the OVERWATCH modules. As such, the capabilities of this module consist of the following:

- Providing a Unity-based AR application which allows the generation of 3D content based on the information from the OVERWATCH system.
- Streaming of this AR application (mediated by the ISAR SDK technology) to AR client devices (AR Frontend) for holographic 3D viewing for end-users.
- Retrieval/sending of information from/to the Management Backend.
- Interfacing with other OVERWATCH modules.
- Receiving input information (e.g., head pose) from the AR Frontend client AR device to subsequently send to the OVERWATCH system.

The module here therefore consists of the application which facilitates the remote rendering and visualization of 3D content derived from relevant OVERWATCH information.

2.4.4. Expected technical inputs and expected outputs

Table 19 - AR Backend inputs and outputs for Remote rendering component

Type of information	Information
Format	Unity SDK
Input payload draft/final	N/A
Output payload draft/final	Enabled remote application rendering functionality
Elapsed time	N/A
Prerequisites	Unity project with defined application logic and available client application to interact with. Hardware to run the application on, i.e,. Laptop, Cloud with NVIDA GRID, VM with GPU passthrough, etc. Stable WiFi connection between backend application (HoloServer, Backend) and AR Frontend application (AR Client). • Windows 10 (10.0..17763 Build), Windows 11 or Wind0ows Server 2019 With Direct X 11 support • GPU NVIDIA RTX 3080 TI, NVIDIA GRID for VMs • Unity 2019.4 and 2020.x • Network connection needed (Wi-Fi 5Ghz recommended) • Bandwidth 20 - 40 Mbit • Open TCP port 9999 • Open outgoing UDP ports 16384-32768
Comments	Remote Application Rendering is a two-component solution. The Unity SDK is the first one.

Table 20 - AR Backend inputs and outputs for Client Component (Frontend application)

Type of information	Information
Format	XAML based application

Input payload draft/final	IP address from Remote Rendering Component
Output payload draft/final	Visualization of AR application, including app logic.
Elapsed time	Motion to Photon latency is 80ms
Prerequisites	Backend application containing all app logic available, table WiFi connection and the open port 9999.
Comments	Supported devices are not limited to, but also include HoloLens 2, Oculus Quest 2, Android (not full feature set), iOS (not full feature set); More clients in development.

Table 21 - AR Backend inputs and outputs for Signalling Server

Type of information	Information
Format	API calls
Input payload draft/final	N/A
Output payload draft/final	Initiates communication between AR Backend and AR Frontend application via the "handshake" during the peer-to-peer connection establishment.
Elapsed time	N/A
Prerequisites	Available app infrastructure, open ports as mentioned above, stable connection.
Comments	

Table 22 - AR Backend inputs and outputs for Map data interface

Type of information	Information
Format	JSON or XML
Input payload draft/final	Depending on required format: IDs, name, coordinates, directional data, pull requests.
Output payload draft/final	Map data tiles per layer as .png
Elapsed time	N/A
Prerequisites	Initiated connection between backend application and map data storage location. Stable internet connection.
Comments	

Table 23 - AR Backend inputs and outputs for Additional interfaces

Type of information	Information
Format	REST APIs, JSON, XML
Input payload draft/final	N/A
Output payload draft/final	Depending on the interface, e.g., weather data, meta data, warning messages, etc.
Elapsed time	
Prerequisites	Unity Plug in or equivalent available, stable network setup.
Comments	

2.4.5. Available interfaces

The interfaces can be any Unity plugin or REST APIs. The unity plugins can be of two types: managed plugins and native plugins.

Managed plugins are managed .NET assemblies created with tools like Visual Studio or MonoDevelop while Native plugins are platform-specific native code libraries. They can access features like OS calls and third-party code libraries that would otherwise not be available to Unity[4].

REST APIs are based on a set of principles and constraints that make it easy to build scalable and interoperable systems, client and server are separate entities that communicate over a network, each request from a client to a server must contain all the necessary information for the server to understand and process the request and the interfaces between the client and the server are standardized and consistent.

2.4.6. Mapping of the functional requirements and the end user requirements within the technical module

Table 24 maps the functional requirements of the module with the user requirements defined in D1.1.

Table 24 - Functional requirements for AR Backend

IDENTIFIERS	FUNCTIONAL REQUIREMENTS	END USER REQUIREMENTS	DESCRIPTION END USER REQUIREMENTS	COMMENTS
AR-01	Map data format	**AR-M-000** **AR-M-010**	Data provided must be able to integrate into Unity and must have selectable layers.	
AR-02	Layer Selection Functionality	**AR-M-010** **AR-M-130**	Application must have the ability to select different layers.	

[4] https://docs.unity3d.com/560/Documentation/Manual/Plugins.html

AR-03	Remote Rendering	**AR-M-020** **AR-M-030** **AR-M-090** **AR-M-100** **AR-S-020** **AR-C-010** **AR-SCA**	Remote application rendering must be supported. Client application (AR frontend) must send input methods to AR backend server.	
AR-04	Metadata availability	**AR-M-040**	The OVERWATCH must provide relevant metadata.	
AR-05	Measuring features	**AR-M-040**	The application should provide measuring features (e.g., distance, volume, etc.)	
AR-06	Multi-platform support	**AR-M-050** **AR-M-030**	The solution must function with multiple device types (e.g., HoloLens2, iOS device)	
AR-07	Image stability	**AR-M-060**	The visualized content has to remain stable in space through e.g., reprojection, etc.	
AR-08	Object Recognition	**AR-M-070**	Where relevant, the application should provide object recognition and tracking through third-party plugins.	
AR-09	Multi-user	**AR-M-080** **AR-S-050**	The application must support multiple users in one session and allow interaction with each other.	
AR-10	Training module	**AR-M-110**	The solution must provide some form of training (e.g., tutorial, user guide, on-site training session) dependent on pilot execution logistics.	
AR-11	External visualization	**AR-M-120**	AR application, as well as users' point of view (POV) must be accessible and sharable with additional screens.	

AR-12	Possibility to integrate with external interfaces	**AR-S-000** **AR-S-010** **AR-S-060**	The application should interface with additional sources (e.g., weather, GPS, scenario simulation etc.).	
AR-13	Protocol generation	**AR-S-040**	The application should generate a protocol which includes relevant metadata (e.g., time stamps, made measurements, pulled weather information, etc) in the form of an XML or CSV file.	
AR-14	Research into possibility to integrate with external interfaces	**AR-C-000** **AR-C-030** **AR-C-040**	The application could interface with additional sources (e.g., weather, GPS, scenario simulation etc.).	
AR-15	Hardware Access based on biometric data	**AR-C-060** **AR-SEC**	The used hardware could manage access based on biometric data (e.g., retina, fingerprint, etc.).	
AR-16	User friendly UI design	**AR-ACC** **AR-USBT**	The UI must be minimalistic and comply with user disabilities.	
AR-17	Network stability	**AR-MNT** **AR-PERF** **AR-REL**	The network must be stable and robust	

2.5. AR Front end

2.5.1. Brief description of the tool/module

The AR Frontend model represents the AR visualization human-machine-interface component of this OVERWATCH system. The AR Frontend is also referred to as an “AR Client”. The AR Frontend, which consists of the AR device that visualizes the 3D content for the end-user, also consists of a software component, the ISAR client application which is installed on the AR device of choice (e.g., HoloLens2, iOS and Android devices). The ISAR client application allows the receiving of the AR application stream from the AR Backend (see section 2.4). This connection is facilitated by the ISAR SDK functionality, mediated by WebRTC protocol. In return, the AR Frontend sends input data such as sensor data for room tracking, head pose, audio stream, gesture input, SLAM to the AR Backend HoloServer (i.e., output data of this module).

2.5.2. Expected users of the system

As in the AR Backend, the users of the AR Frontend will be the following:

- Strategic decision-makers and tactical management and operational decision-makers who are located in the command centre.
- OVERWATCH researchers/personnel who will initiate the connection between the AR device and the AR Backend.

The tactical management and operational decision-makers are the key users here, as they will be using the AR devices and therefore be the original source of the data which is sent to the AR Backend.

2.5.3. Description of capabilities of the tool/module

The capabilities of the AR Frontend are:

- Receive application pixel stream data from AR Backend.
- Visualize the data holographically in three-dimensions for the end-user(s) using the AR device.
- Send human-machine-interface data (i.e., sensor data) to AR Backend

2.5.4. Expected technical inputs and expected outputs

Table 25 - Expected inputs and outputs for AR Frontend.

Type of information	Information
Format	Input into ISAR client application (AR Frontend) is pixels streams which have been converted into byte array.
Input payload draft/final	N/A
Output payload draft/final	Output to the AR Backend is sensor data which has been converted into byte array.
Elapsed time	Motion to Photon latency is 80ms
Prerequisites	To ensure that data can flow between the AR Frontend and AR Backend, the following is needed: • Network connection needed (Wi-Fi 5Ghz recommended) • Bandwidth 20 - 40 Mbit
Comments	

2.5.5. Available interfaces

AR Frontend (AR Client) interfaces with only the AR Backend (HoloServer) for data exchange via WebRTC. Currently, no other interface is envisioned for the AR Frontend.

2.5.6. Mapping of the functional requirements and the end user requirements within the technical module

Due to the two-component nature of the AR Frontend and AR Backend (i.e., application streaming), all technical requirements have been represented in the AR Backend (section 2.5.6), as they are functionally interconnected.

2.6. Accurate and secure drone navigation

2.6.1. Brief description of the tool/module

The “Accurate and secure drone mapping using advanced EGNSS signals and services” is one of the components to be developed in T2.2. This component focuses on providing the precise positioning that will be ensured by GNSS systems, with Galileo OSNMA being part of these. Leveraging state of the art receivers capable of using the OSNMA signals this component will provide accurate and secure positioning to the system. The provided “component” is not a software service but a physical GNSS receiver to be integrated onto the drone. The integration of the receiver with the navigation system will be under the drone manufacturer’s objectives. The “Accurate and secure drone navigation” does not require some of the subsections that other systems in the work package present, due to its nature.

2.6.2. Expected users of the system

The positioning module shall be an add-on to be integrated with the drone platform. The module will incorporate a GNSS receiver, which basic inputs are an antenna and power source. Additional ports such as communication via serial will need to be connected to the receiving drone.

Once installed, the module shall provide positioning information where the basic conditions are met.

The module is dependent only on a power supply and an antenna. The outputs are to be fed to receiving drone. If additional services that require downloading information such as corrections from the web are needed, then an internet connection should be envisioned. This module will not interface directly with the end user but provide only information for accurate mapping.

As of April 2023, Galileo HAS (High Accuracy Service) has been made available by the European union, but there is a lack of receivers in the marketplace that support such signal. While this limitation should be gradually solved in the foreseeable future, with the adoption of the HAS system by major manufacturers, additional methods (RTK post-processing, Differential GNSS) will be considered in the context of this project as a fallback option.

Other constraints remain to be evaluated.

2.6.3. Description of capabilities of the tool/module

The module should provide accurate and reliable positioning information, making use of both Galileo OSNMA and HAS. Unfortunately, as stated above only the first service is provided by state-of-the-art receivers. Other methods for providing higher than standard accuracy are being evaluated, such as RTK and DGNSS (Differential GNSS).

2.6.4. Expected technical inputs and expected outputs

Table 26 - Inputs and outputs from accurate drone secure navigation.

Type of information	Information
Format	NMEA 0183
Input payload draft/final	N/A
Output payload draft/final	NMEA 0183
Protocol	HTTP, TCP, IP/FTP, USB

2.6.5. Available interfaces

The positioning module can communicate with the drone using the USB interface, providing positioning information using the NMEA standard.

2.6.6. Mapping of the functional requirements and the end user requirements within the technical module

In the Table 18 are mapped the functional requirements of the module with the user requirements defined in D1.1.

Table 27 - Functional requirements for accurate and secure drone navigation

IDENTIFIERS	FUNCTIONAL REQUIREMENTS	END USER REQUIREMENTS	DESCRIPTION END USER REQUIREMENTS	COMMENTS
ASDM-1 (Accurate Secure Drone Mapping)	The system shall provide accurate and secure positioning information to the system.	DT-M-020 DT-M-050 DT-M-110	The drones must be able to fly autonomously or remotely piloted, with the ability to take off, hover, and land safely. The drones must use signals from GNSS to navigate in the airspace. The drones must have on-board real-time processing of data from sensors.	

2.7. Drone terrain mapping

2.7.1. Brief description of the tool/module

The system comprises a VTOL drone that is equipped with RGB and Thermal cameras, lidar, and an onboard computer. This drone has the capability to perform long-range flights while simultaneously processing sensor data in real-time on its onboard computer.

Deployed on the onboard computer is a specialized module that enables real-time mapping of the drone's captured area. This module facilitates automatic adjustments to the navigation path, ensuring efficient and accurate surveying and analysis of areas impacted by natural disasters like wildfires and floods.

The real-time maps generated are streamed to a ground station, allowing first responders to quickly access and gain an overview of the affected area.

2.7.2. Expected users of the system

The main objective of this system is to offer first responders a quick overview of a mapped area. In this context, prioritizing the speed of map generation is more important than absolute accuracy. The system aims to rapidly provide emergency responders with initial information about the scope of wildfires or floods in a particular area, enabling them to make informed decisions. However, the collected data is stored in a geospatial repository for further analysis and the production of more precise maps later on.

2.7.3. Description of capabilities of the tool/module

The VTOL drone has the capability to dynamically optimize its navigation path during flight for efficient mapping of wildfire perimeters, as well as for mapping floods.

The system generates maps by utilizing both RGB and thermal bands, combining orthorectified images with topological maps. Furthermore, it is equipped with lidar technology that produces topographic maps of the surveyed area. The images undergo orthographic correction to mitigate perspective distortion and ensure precise representation.

2.7.4. Expected technical inputs and expected outputs

Table 28 - Drone terrain mapping expected inputs and outputs for real time mapping.

Type of information	Information
Format	.tiff, GeoJSON, KML, KMZ (TBA)
Input	N/A
Output	map(s) and GIS layers (vector / raster)
Protocol	TBA
Elapsed time	5-60 seconds
Prerequisites	Network and radio connection availability.
Comments	The real-time maps for the display to the user. The primary delay is expected due to data transmission from onboard computer on the VTOL to the end user.

Table 29 - Drone terrain mapping expected inputs and outputs for Area selection for navigation.

Type of information	Information
Format	JSON
Input draft	`{ "selection_area": [ { "latitude": double, "longitude": double }, { "latitude": double, "longitude": double }, { "latitude": double, "longitude": double }, { "latitude":  double, "longitude": double }, { "latitude": double, "longitude": double }`

	] }

Output draft	{
	"Received": [bool]
	}

Protocol	TBA
Prerequisites	Network and radio connection availability.
Comments	Optional setting to send the search coordinates from a user interface. The alternative is to use the application designed for the VTOL on the controller.

Table 30 - Drone terrain mapping expected inputs and outputs for VTOL Telemetry.

Type of information	Information
Format	JSON
Input draft	N/A

Output draft	{
	"Timestep": [double],
	"DroneStatus":[int],
	"DronePosition":{[
	"latitude": double,
	"longitude": double,
	"altitude": double,
	]},
	"Battery Level":[float],
	"GpsStrength":
	"RadioStrength":[int]
	
	...
	
	}

Protocol	TBA
Prerequisites	Network and radio connection availability.
Comments	This data represents the telemetry of the drone, this is the overall draft of the structure and will be modified according to the user needs during the Overwatch project.

Table 31 - Drone terrain mapping expected inputs and outputs for Orthorectified images.

Type of information	Information
Format	.JPG
Input payload draft	N/A
Output payload draft	TBA
Protocol	TBA

Prerequisites	Network and radio connection availability.
Comments	The initial VTOL images captured by the camera undergo orthorectification and are then transmitted to the geospatial repository. These images are embedded with metadata, including intrinsic and extrinsic parameters of the camera.

2.7.5. Available interfaces

An interface is currently not available and will be developed during the OVERWATCH project.

2.7.6. Mapping of the functional requirements and the end user requirements within the technical module

In the Table 32 are mapped the functional requirements of the module with the user requirements defined in D1.1.

Table 32 - Functional requirements for Drone terrain mapping

IDENTIFIERS	FUNCTIONAL REQUIREMENTS	END USER REQUIREMENTS	DESCRIPTION END USER REQUIREMENTS	COMMENTS
DT-01	**Autonomous Flight and Navigation**	DT-M-020, DT-M-030, DT-M-040, DT-M-050, DT-M-060, DT-M-100, DT-S-010	The drones must be capable of autonomous or remote flight, using flight control and stabilization systems, algorithms, GNSS signals, and obstacle detection to navigate safely in diverse conditions. They should also support switching between piloted and autonomous modes of operation whenever necessary.	The autonomous flight system should include functionality to take off, hover, land safely and respond to changes in the environment. Flexible mode switching capability is vital for varying operational needs.
DT-02	**Drone safety**	DT-M-070, DT-M-090	The drones must have the ability to return to base or land safely in case of a malfunction or loss of communication and should be able to communicate their presence to other drones or the	A reliable return-home feature or safe landing protocol is necessary to handle communication loss or other malfunctions. Additionally, a secure and

			respective GCS to help coordinate sharing the same airspace.	efficient drone-to-drone and drone-to-GCS communication protocol is necessary for basic airspace management.
DT-03	**Emergency Response and Environmental Adaptability**	DT-M-000, DT-M-010, DT-M-120, DT-S-020	The drones must support emergency responders with functional capabilities and operate effectively in a diverse range of visibility conditions and adverse weather conditions like moderate wind, informing the GCS accordingly.	The drone must be equipped with relevant modules to assist in emergency scenarios, and advanced environmental sensors for safe and efficient operation in various conditions.
DT-04	**Data Relay/Streaming and Real-Time Data Processing**	DT-M-080, DT-M-110 DT-S-020	The drones must be capable of relaying/streaming data to the GCS/Geospatial repository and have on-board real-time processing capabilities of data from sensors.	The drones should have a robust communication system for efficient data transfer and on-board computational power for real-time data processing.

2.8. Fallback communication system

2.8.1. Brief description of the tool/module

The system consists of a tethered drone that will have power and fibreoptic link between a GCS and the drone itself and will carry on-board an embedded communication device and antennas that will allow to establish and act as communications backhaul to the OVERWATCH system.

The OVERWATCH fallback communications module consists of a fully integrated tethered drone and communications system composed of an aerial and ground component, interconnected by the fibre optic link provided by the drone tether. The aerial component contemplates the antennas and the necessary hardware to operate as Wi-Fi, and possibly 5G transceivers, creating a local wireless Internet hotspot for the drone GCSs and other systems in wireless range. The ground component is responsible for establishing an Internet backhaul link by using Low Earth Orbit (LEO) satellite communications hardware and provide means of local wired Ethernet connectivity to nearby systems such as drone GCSs and on-site command centre. It also houses part of the 5G processing hardware, taking advantage of the functional splits made possible by Open Radio Access Technologies (O-RAN) in order to reduce the complexity, power consumption and weight of the aerial component (the lower the payload, the higher the tethered drone can fly to overcome obstacles and improve radio line-of-sight to its users).

2.8.2. Expected users of the system

Civil Protection and Emergency Response teams.

Possible systems connected to this tool/module:

- Drone Ground Control Stations which, then, relay connectivity to the drones.
- On-site command centre (and its systems)
- Mobile terminals

2.8.3. Description of capabilities of the tool/module

All system to be developed during OVERWATCH project. However individual hardware components will be bought e.g., sensors, GCS, cables, fibre optics, drone,

2.8.4. Expected technical inputs and expected outputs

Table 33 - Fallback communication system expected inputs and outputs.

Type of information	Information
Format	JSON
Input	---

Output	{
	 “wifi_status”: boolean, // enabled (1) disabled (0)
	 “5g_status”: boolean, // enabled (1) disabled (0)
	 “rtt_to_cloud_ip”: integer,
	
	 //others, if possible:
	
	 “mv_avg_uplink_speed”: float, // in kBit/s
	 “mv_avg_downlink_speed: float, // in kBit/s
	 “nr_wifi_users”: integer,
	 “nr_5g_users”:integer,
	 “nr_ethernet_users”: integer,}

Protocol	HTTP

2.8.5. Available interfaces

Robot Operating System (ROS or ros) that is an open-source robotics middleware suite will be used over REST interface.

2.8.6. Mapping of the functional requirements and the end user requirements within the technical module

In the Table 34 are mapped the functional requirements of the module with the user requirements defined in D1.1.

Table 34 - Functional requirements for the fallback communication system

IDENTIFIERS	FUNCTIONAL REQUIREMENTS	END USER REQUIREMENTS	DESCRIPTION END USER REQUIREMENTS
FCS-FR-1	Provide external connection with other OVERWATCH sub-modules and legacy systems (namely C2)	FCS-FM-000 FCS-FM-010 FCS-FS-010	Provide external connection between the fallback connectivity and other Overwatch sub-modules, C2 and other legacy systems
FCS-FR-2	Provide Connectivity (network and Internet access) to other systems	FCS-FM-030 FCS-C-000	The fallback connectivity system must be able to provide support for reliable communications e.g, Wi-Fi, Ethernet and (if available) 5G to provide redundant communications coverage to nearby local systems, allowing them to communicate locally and to the Internet (e.g. upload and download data from the cloud systems).
FCS-FR-3	Tethered drone GCS functionalities	FCS-FM-020 FCS-FM-040 FCS-FM-060 FCS-FM-070 FCS-FM-080 FCS-FS-020	These requirements relate to the functional requirements of the tethered drone GCS and capabilities to system a reliable and functional system. Namely capability to ensure communications, user-friend interface, log all activities and be able to receive data.
FCS-FR-4	Tethered drone functional capabilities	FCS-FM-050	The tethered drone must remain airborne for extended periods of time and land safety whenever it loses connection to GCS
		FCS-FM-080	

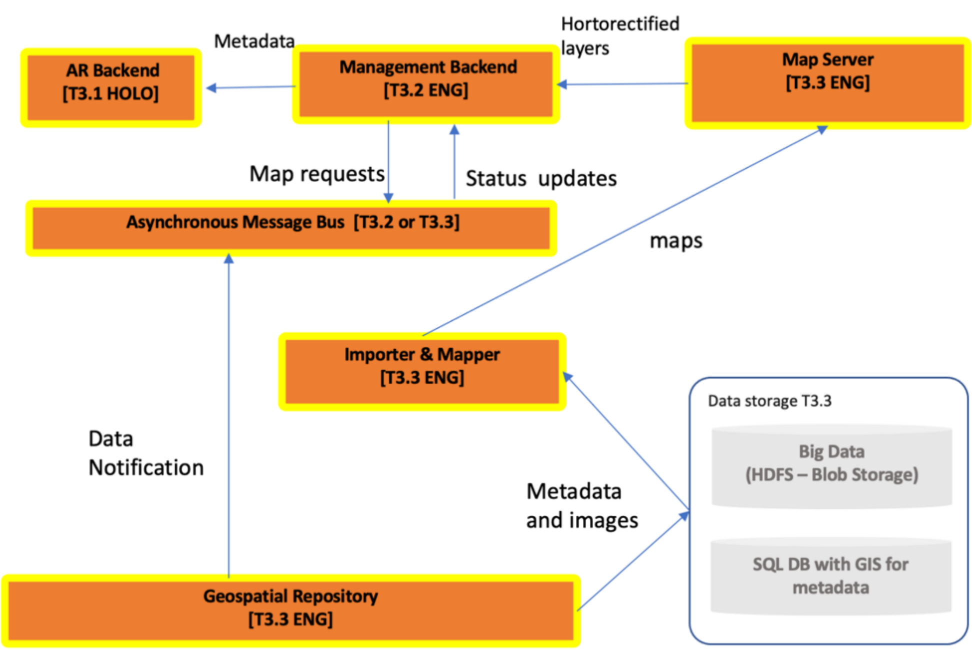
3. Data flow diagrams and interactions with the other modules

3.1. Management backend

3.1.1. Workflow diagram of the tool and operational concept

The diagram below illustrates the direct interaction of the Management backend with the other modules. The main interactions are with the Message Bus to retrieve the metadata related to the different conditions of the terrain, the raster images that will be used for the representation of the layers, the geospatial repository that will store the information on the data storage block, the importer and mapper module that will store everything in the map server and the map server that will serve the layers through the Management back end to the GUI.

Figure 2 - Workflow diagram of the module and integration logic of the Management backend



3.1.2. Technical requirements

3.1.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	The system must be available at all times, even during peak usage or unexpected events	Backend management	Service up and running
#2	The AIMS system must be highly reliable, with minimal downtime and a low probability of failure	Backend management	Test the throughput
#3	The system must ensure the availability of	Backend management	Test map layers

	maps for the 2D front end and the AR front end		
#4	The OVERWATCH must ensure that the privacy of individuals is protected and that the data is only used for its intended purposes.	Backend management	Test services providing sensitive data

3.1.2.2. Performance requirements

Id	Description	Family of services	Service to test
#1	The system must be able to process and analyse large amounts of data in real-time, providing timely insights to emergency responders	Backend management	Test requests on metadata and on maps
#2	The system must be able to stream scalable volumes of data to accommodate on-site commanders operating from satellite terminals	Backend management	Test the throughput

3.1.2.3. Reliability requirements

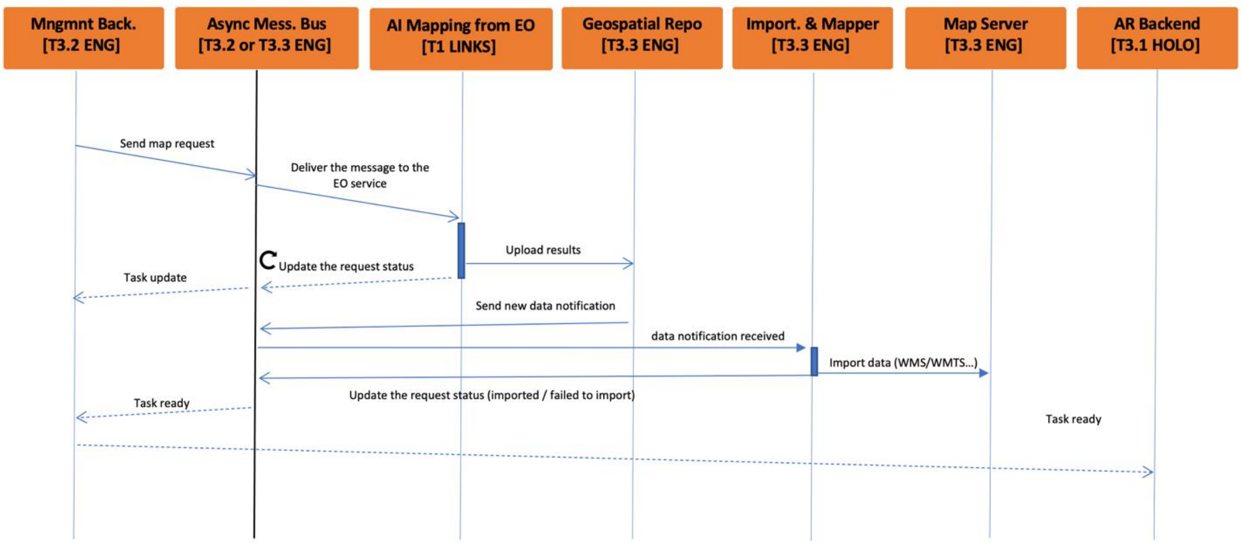
Id	Description	Family of services	Service to test
#1	The AIMS system must be highly reliable, with minimal downtime and a low probability of failure, to ensure it can provide the necessary information to emergency responders in critical situations	Backend management	Test status of the backend
	The system should be easy to maintain, with minimal downtime required for maintenance and updates	Backend management	

3.1.2.4. Availability requirements

Id	Description	Family of services	Service to test
#1	The AIMS must incorporate a backup and restore for all its contents	Backend management	n/a

3.1.3. Interactions with the other modules

Figure 3 - sample integration of the backend with the AI Mapping from EO and the AR backend



In the figure above the Management backend in order to be triggered an interaction needs to be started from the end user by using the GUI of the Web based dashboard. The end user asks for the mapping of the terrain in a particular area in order to know which is the evolution of the hazard, e.g. flood or wildfire. The request is transformed by the Backend in a topic that will be produced on the message bus. The AI mapping from EO will consume the information, will elaborate the request according to the selected area, and will return a series of updates, including the references to the images stored in the data lake, if successful.. The importer and mapper will be activated on upload to retrieve all the information from the different types of storage and will import raster images and other data onto the map server. The layers when ready will be served to the web-based dashboard and to the Augmented Reality backend that will serve all the information to the HOLOLens to the operators on site.

3.1.4. Roadmap of the release of the module

The backend management will be released in three different phases:

M13 release of the prototype of the Management backend with the integration of the message broker and single sign on and the setup of the Geospatial repository

M19 implementation of the importer and mapper with the storage on the map server subsystem

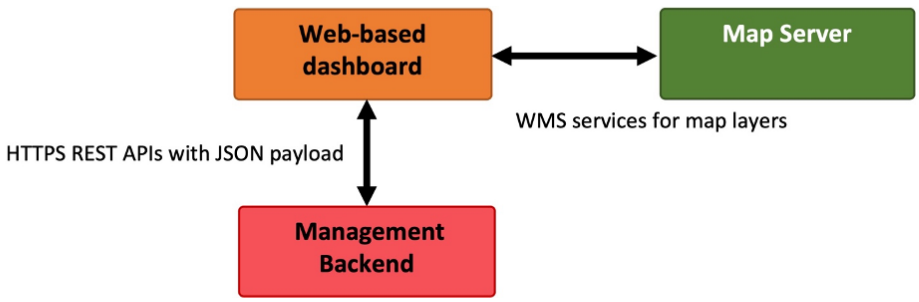
M25 integration with the AR backend system

3.2. Web-based dashboard

3.2.1. Workflow diagram of the tool and operational concept

The below diagram illustrates the direct interaction of the Web-based dashboard with the other modules. The main interactions are with the Management Backend that provides a detailed picture of a current situation, and the Map Server, that provides the georeferenced map images which include additional information can enhance and enrich the current situational awareness.

Figure 4 - Workflow diagram of the module and integration logic



3.2.2. Technical requirements

3.2.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	The system must be available at all times, during test usage	Dashboard	Management Backend

3.2.2.2. Performance requirements

Id	Description	Family of services	Service to test
#1	The system must be able to visualize large amounts of data in real-time, providing timely insights to emergency responders	Dashboard	Management Backend
#2	The system should be easy to use and intuitive, requiring minimal training for emergency responders to understand how to use it effectively	Dashboard	Management Backend

3.2.2.3. Reliability requirements

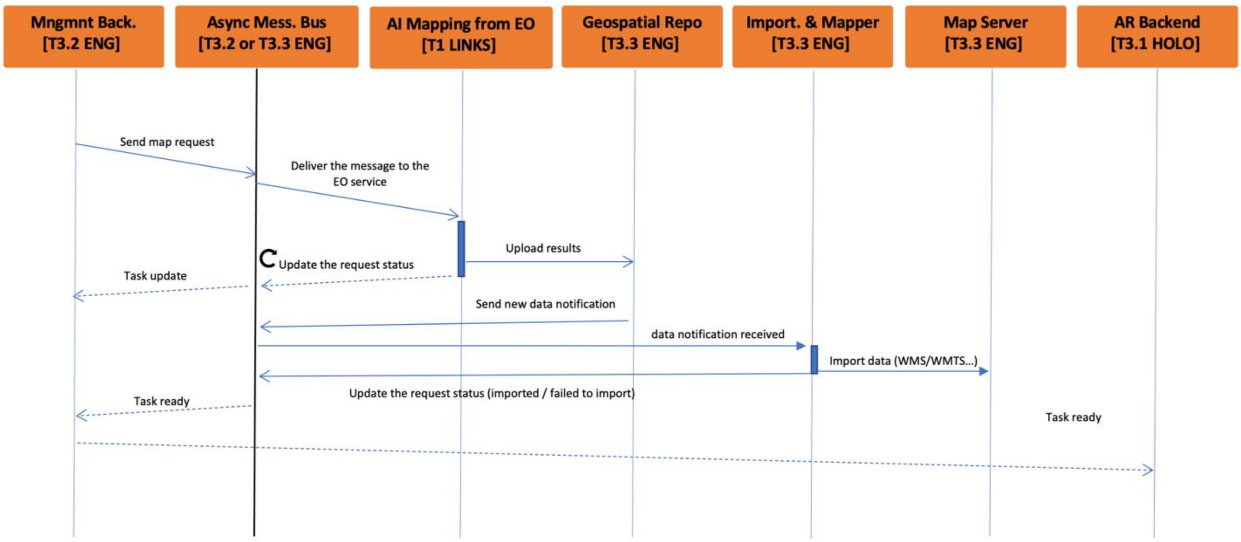
Id	Description	Family of services	Service to test
#1	The system should provide a user-friendly interface for emergency responders to access and use the system efficiently	Dashboard	Management Backend
#2	The system must be highly reliable, with minimal downtime and a low probability of failure, to ensure it can provide the necessary information to emergency responders in critical situations	Dashboard	Management Backend

3.2.2.4. Availability requirements

Id	Description	Family of services	Service to test
#1	The system should be compatible with intended devices and platforms, ensuring that emergency responders can access them from predefined locations and devices	Dashboard	Management Backend
#2	The system must be available at all times, even during peak usage or unexpected events, to ensure that emergency responders can rely on it when needed	Dashboard	Management Backend

3.2.3. Interactions with the other modules

Figure 5 - sample integration of the Web Based Dashboard



The end user by using the GUI of the Web-based dashboard will activate an interaction with the Management backend. The use case interaction is the same as for the Management Back end 3.1.2

3.2.4. Roadmap of the release of the module

The web-based dashboard will be released in the different phases:

M14 release of the prototype of the web-based dashboard with the first integration of the Management backend.

M20 integration with the map server subsystem

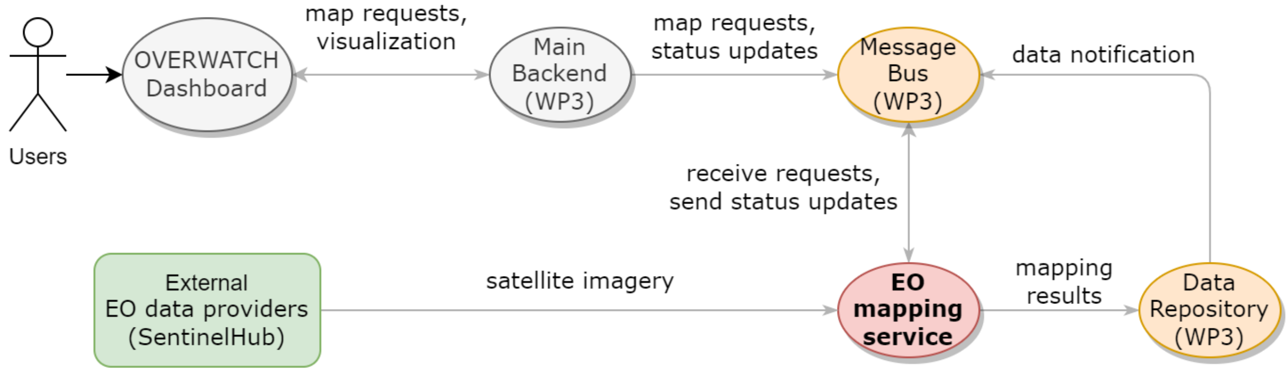
M23 release of the new version of the prototype for end user training

3.3. Mapping from EO

3.3.1. Workflow diagram of the tool and operational concept

The module does not interact with any legacy system, at least directly. Its only inputs and outputs comprise the message bus for communication, and the data repository for data storage.

Figure 6 - Mapping from EO, operational concept.



3.3.2. Technical requirements

3.3.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	Test the retrieval of EO imagery from external services	Mapping from EO	Retrieve EO data
#2	Test the creation of a map request, either manually or directly from the main frontend UI	Mapping from EO	Create map request
#3	Test the sending of update notifications for a given request	Mapping from EO	Update map request
#4	Test the upload of the retrieved EO data and layers produced by the service	Mapping from EO	Conclude map request

3.3.2.2. Performance requirements

Id	Description	Family of services	Service to test
#1	The service shall be able to handle multiple subsequent requests	Mapping from EO	Handle map requests
#2	The service shall be able to produce the required mapping in a relatively short amount of time (minutes or hours at most)	Mapping from EO	AI-based request processing
#3	The service shall be able to provide the required products on demand within 15 days.	Mapping from EO	Manual map request processing

#4	The service shall be able to handle large data files and outputs (several GB)	Mapping from EO	Data management and upload

3.3.2.3. Reliability requirements

Id	Description	Family of services	Service to test
#1	The service shall be able to handle failures from external tools (EO data providers, ML models, others) and provide meaningful updates	Mapping from EO	Map request processing
#2	The service shall be able to provide a retry mechanism for (possibly) recoverable errors	Mapping from EO	Map request processing

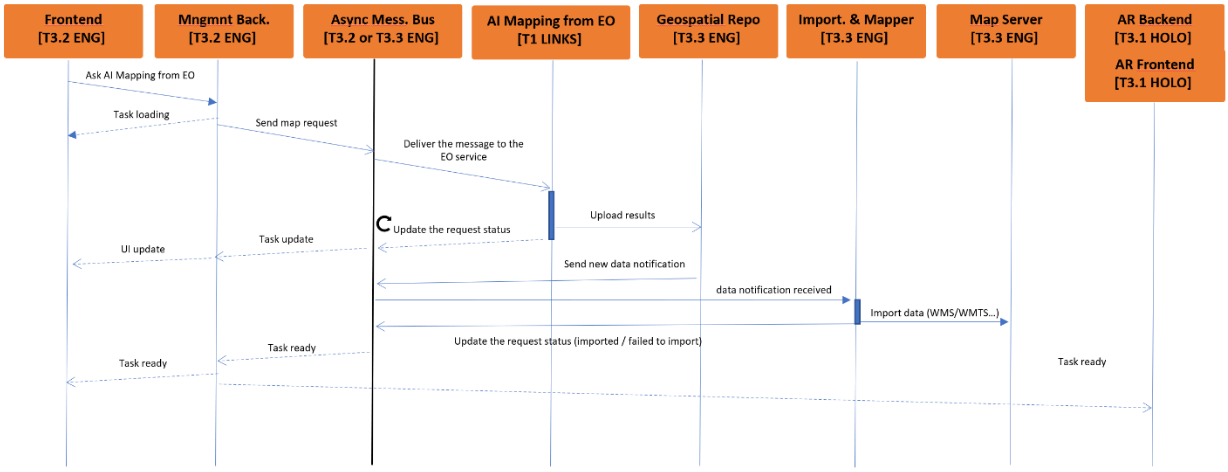
3.3.2.4. Availability requirements

Id	Description	Family of services	Service to test
#1	The service shall have an adequate bandwidth to download the required EO data and upload the resulting outputs.	Mapping from EO	Map request processing, data upload
#2	The service shall have a stable connection with the message bus to be able to receive map requests and send updates	Mapping from EO	Map request creating, status updates

3.3.3. Interactions with the other modules

The full map request lifecycle is detailed in the following sequence diagram. The EO mapping module itself has a simple interaction flow: when a request is received from the message bus, the processing begins. During this time, the service sends update messages containing the status of the ongoing map request. Once the request is completed successfully, data is uploaded onto the project-level data repository. In turn, the repository shall update other listeners that new data layers have been uploaded (e.g., the importer and map server). The full cycle can be considered completed when the raw GeoTIFF/GeoJSON outputs have been converted into actual map layers.

Figure 7 - Interaction diagram for EO mapping



3.3.4. Roadmap of the release of the module

- M9: First AI baselines ready for preliminary tests.
- M12: release of the prototype of the EO service module to test the functionalities.
- M18: initial integration with the message bus, backend and repository services.
- M18 onwards: iterative improvements of the mapping solutions and the ML models.

3.4. AR backend

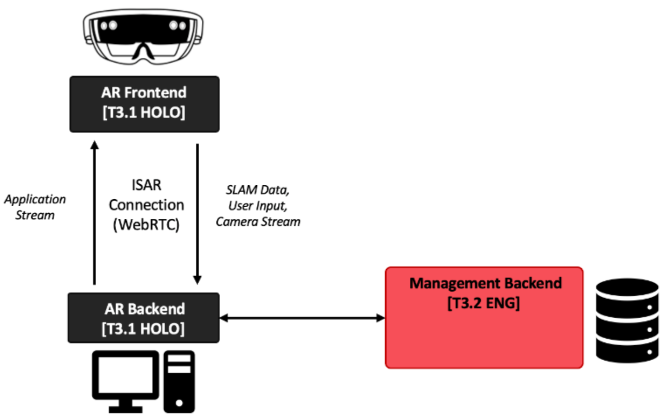
One major function of the created solution is the display of the generated map data in AR, which in turn is then enhanced with additional information. These requirements are usually render heavy and cannot be executed by contemporary smart glasses, such as the HoloLens 2. Hence, the AR Backend module utilizes HOLO's own ISAR technology, which enables remote application rendering.

The remote application rendering solution splits the app into two components: the Server (AR Backend); and the Client (AR Frontend).

This section will address the first component, which runs on a Laptop with a powerful GPU, the Cloud or on a VM and contains the entire app logic, UI and all relevant interfaces.

3.4.1. Workflow diagram of the tool and operational concept

Figure 8 - Workflow diagram of the module and integration logic.



As mentioned before, the AR backend is the first of the two-component solution. Using WebRTC the AR Backend establishes a connection to the AR Frontend application, which is installed on the smart glasses. The backend solution does not only contain the app logic, but also interfaces with all external tools, i.e., Management backend, possible message brokers, additional data sources, like weather data, etc. As shown in the figure above, the simplest depiction of the workflow is all relevant OVERWATCH tools which integrate into the Management Backend which is in turn connected to the AR Backend. The AR Backend receives from and sends data back to the Management Backend. The application stream in turn is being sent to the AR Frontend. This stream visualises the application content, as well as all UI elements. The AR frontend cannot only receive the data stream, but also sends information back, such as head pose, input methods, SLAM, Audio, etc.

3.4.2. Technical requirements

The AR Backend application allows more variety, as the application does not have to adhere to the device specific limitations. Nonetheless, certain requirements are needed to enable this:

Hardware Requirements

	Minimum	**Recommended**
Operating System	Windows 10 (10.0..17763 Build) Windows 11 Windows Server 2019	Windows 10 (10.0..17763 Build) Windows 11 Windows Server 2019
Memory	16 GB	64 GB
CPU	Intel i5 8th Gen. 6 Cores AMD Ryzen 7 3700X	Intel i7 12 Gen. 12 Cores AMD Ryzen 9 3900X
GPU	NVIDIA GTX 1070Ti NVIDIA GRID for VMs	NVIDIA RTX 3080 TI NVIDIA GRID for VMs
Storage	SSD or NVMe	SSD or NVMe

Development Environment Requirements

Visual Studio Version	16.7.x 16.8.x
Visual Studio Components	Universal Windows Platform Game development with Unity Game development with C++ Desktop Development with C++
Unity Version	2019.4.x 2020.3.x
Unity Components	Universal Platform Build Support Windows Build Support IL2CPP
Mixed Reality Toolkit Version	2.5.x 2.6.x 2.7.x (as of ISAR .2.3.0.0)

Network Requirements

	Minimum	Recommended
Network	Wi-Fi 5Ghz	Wi-Fi 5Ghz
Bandwidth	20 Mbit	40 Mbit
Round Trip Time (Latency)	-	Max. 50 ms

Enable incoming TCP Port 9999 and outgoing UDP Ports 16384-32768.

The local network of the AR Device must have the ability to use STUN (Session Traversal Utilities for NAT) in case the HoloServer is not located in the same network as the AR Client Device.

3.4.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	Stress test network stability during single use and multi-user use	AR frontend AR backend Management backend	Network
#2	Connectivity between AR backend and Management backend.	AR backend and Management backend	Push and pull request
#3	Latency of data transfer between AR backend and Management backend, to ensure a fluid display of all map tiles.	AR backend and Management backend	Connection between AR backend and Management backend
#4	After receiving a visual alert from the operator in a scarce connectivity context, a mission should be created.	Mission Management (sample)	Create Mission
#5	Bandwidth tests to confirm impact of resolution drop.	AR frontend	WebRTC
#6	Enable and disable display of map data layers.	AR backend AR frontend input methods Management backend	
#7	Enable and disable the display of additional content, such as weather information.	AR backend AR frontend input method External component	
#8	Ensure that during a multi-user session all relevant movements and inputs are	AR backend AR frontend	Photon

	displayed between all users.		
#9	Ensure data transfer between all interfaces.	AR backend	
#10	Ensure simple usability of the UI elements.	AR frontend	MRTK, etc.
#11	Ensure transmission of all relevant input data.	AR frontend AR backend	Custom send function, etc.
#12	Verify measurement results and ensure that deviation remains in acceptable range.	AR backend AR frontend input method	
#13	Verify that recorded details relevant for the protocol are accurately captured and stored.	AR backend	
#14	Verify input methods.	AR backend AR frontend	Identify if selected method works as intended.

3.4.2.2. Performance requirements

Id	Description	Family of services	Service to test
#1	Ensure the roundtrip time remain within the 50ms threshold.	AR backend Network	
#2	Impact of fast map interaction on GPU and CPU	AR backend	Verify if data is updating fluently or if there are long delays present.
#3	Image quality of the map tiles and layers	AR backend AR frontend Management backend	Assess if long buffering moments are present.
#4	Application can be used on several different platforms	AR frontend	Verify if the appropriate UI elements are displayed on the various selected platforms, i.e. iOS VS HoloLens 2
#5	The system must be able to visualize processed map data and display results of relevant analysis to improve the first responders' situational awareness.	AR backend Management backend	

#6	The system must be able to stream scalable volumes of data to accommodate on-site commanders operating from satellite terminals.	AR backend	MRTK, etc.
#7	Impact of 2+ users on network.	AR backend AR frontend	
#8	Verify that visual quality in unfavourable environments is appropriate.	AR frontend	

3.4.2.3. Reliability requirements

Reliability is an important aspect during an emergency. By using the remote rendering technology of the AR Backend, the main focus here will lie on the Network requirements.

Id	Description	Family of services	Service to test
#1	The network must be at least Wi-Fi 5Ghz.	AR backend AR frontend	
#2	Bandwidth should at least be 40 Mbit to ensure a clear image quality.	AR frontend	Identify lowest acceptable resolution
#3	Handling of poor network quality to reduce risk of shut down.	AR backend AR frontend	Stabilize connection between frontend and backend to avoid unwanted closures.

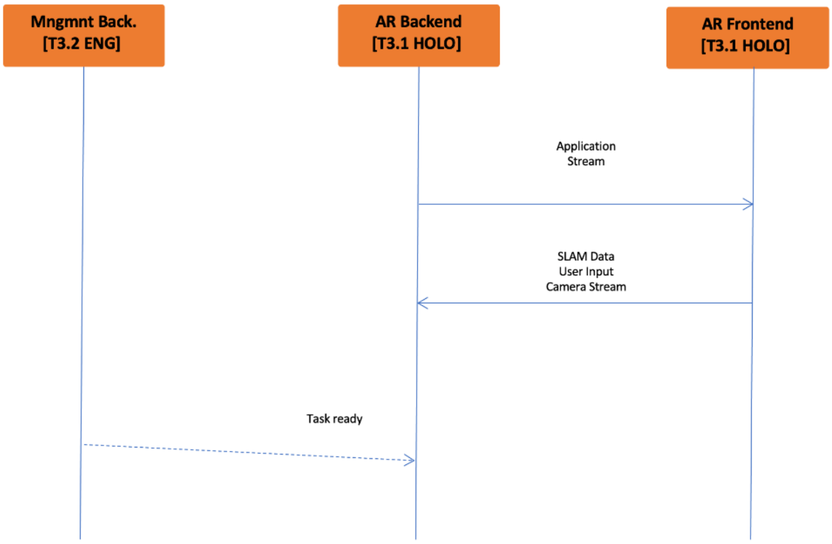
3.4.2.4. Availability requirements

Id	Description	Family of services	Service to test
#1	If Cloud is being used, then measures must be taken to ensure server availability.	AR backend instance	
#2	The system should be compatible with intended devices and platforms, ensuring that emergency responders can access them from predefined locations and devices.	AR frontend	
#3	The system must be always available, even during peak usage or unexpected events, to	AR backend AR frontend Management backend	

	ensure that emergency responders can rely on it when needed.		

3.4.3. Interactions with the other modules

Figure 9 - sample integration of the Management Backend with AR backend.



As described above, information from the Management Backend will be provided to the AR Backend. This information will consist of relevant map data and other information which is to be visualized by end-users utilizing AR devices. The AR Backend streams this data via WebRTC to the AR Frontend (to the ISAR client application) which is then displayed on the AR device for the user to interact and visualize 3D content. The AR frontend sends information, such as head pose and UI input back to the AR frontend, enabling interaction with the application logic. The user can select various map layers to display relevant information which in return pulls the updated data from the Management backend.

3.4.4. Roadmap of the release of the module

The AR Backend development depends on the development of the Management backend. In M25 the integration with the AR backend is scheduled to start.

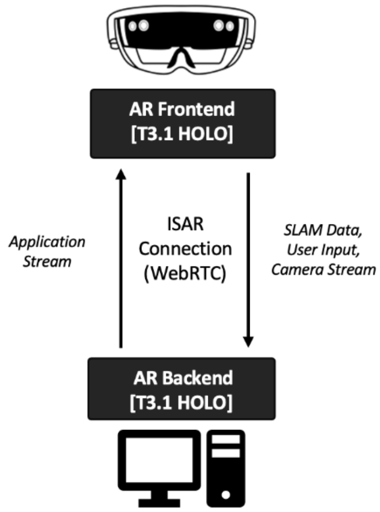
M18: Development of the AR Backend architecture will begin. All relevant and unrelated interfaces will be researched and integrated.

M27: First prototype of the final AR Backend application is envisioned to be available. This will follow a longer period of optimization and adjustments based on user feedback. These adjustments will continue until the final demo.

3.5. AR front end

3.5.1. Workflow diagram of the tool and operational concept

Figure 10 - Workflow diagram of the module and integration logic.



The AR Frontend diagram below represents the streaming data flow between the AR device (AR Frontend) and the HoloServer (AR Backend). The remote rendering and application stream from the AR Backend to the AR Frontend allows the visualization of holographic OVERWATCH information content and interaction with the content. User input data (e.g., head pose) is streamed to the AR Backend for subsequent use in other OVERWATCH modules.

3.5.2. Technical requirements

The AR Frontend application is designed to be a lightweight application installed on the end device of the user, e.g., HoloLens 2, iPad, etc. Therefore, only a few technical requirements and specifications are relevant to each platform.

HoloLens 2

	Minimum Specification	Recommended Specifications
Operating System	Windows 10 version 17763.0 or higher Xbox one	Windows 10 version 17763.0 or higher Xbox one
Architecture	ARM	ARM

Android Devices

	Minimum Specification	Recommended Specifications
Android Version	Android 8 or higher	Android 10 or higher
API	Level 26 or higher	Level 29 or higher

3.5.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	Ensure successful connection between frontend and backend application.	AR frontend AR backend	Signalling server

#2	Transmission of all relevant input data.	AR frontend	Data stream, Custom send, Audio, Hand gestures, Gaze
#3	Test impact of loss of connection.	AR frontend	
#4	Image quality in unfavourable environments is appropriate.	AR frontend	Visual quality outdoor or in room with bright light
#5	Setup of realistic test environment, regarding people count, used platforms, network quality, speed, and urgency	AR frontend AR backend Management backend	

3.5.2.2. Performance requirements

Most the of the requirements for the AR Backend are like those present in the AR Frontend. Therefore, the section below will focus on only those relevant for the AR frontend.

Id	Description	Family of services	Service to test
#1	Latency between UI input in AR frontend and recognition in AR backend	AR frontend AR backend	Custom send, Hand gestures, Gaze
#2	If direct interface to AR frontend is needed, test performance of feature.	AR frontend	No feature currently directly interfaces with AR frontend

3.5.2.3. Reliability requirements

The AR Frontend is mainly a visualization tool and is, therefore, very reliable. If the AR Frontend app shuts down, it does not impact the AR Backend application. For that reason, no data can be lost in such a case. The main issue might only be the successful interaction with the “Connect” button on the AR frontend interface.

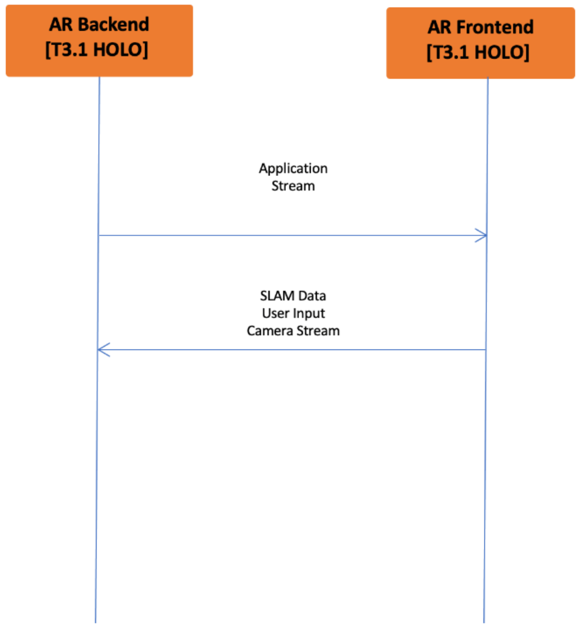
3.5.2.4. Availability requirements

As mentioned, the main impact lies with the AR Backend application. There are no availability concerns if a client is generally available for the defined platform. However, if a client is unavailable, the device in question cannot be used, as the development of an entirely new client is outside the project's scope.

Id	Description	Family of services	Service to test
#1	The defined device list must have clients available.	AR frontend	

3.5.3. Interactions with the other modules

Figure 11 - sample integration of the AR Backend with the AR Frontend.



The AR Backend streams data derived from the OVERWATCH Management Backend via WebRTC to the AR Frontend (to the ISAR client application) which is then displayed on the AR device for the user to interact and visualize 3D content. Sensor data is simultaneously streamed in return to the AR Backend. Additionally, once a user interacts with the UI of the AR frontend application to select a specific layer, this change will be transmitted to the AR backend application and triggers in turn the pulling of the right map data layers.

3.5.4. Roadmap of the release of the module

AR frontend development for the HoloLens 2, iOS, Oculus Quest 2 and Android are ongoing.

M14: Development of the MagicLeap 2 and Oculus Pro will begin, although these are not currently envisioned to be a part of this OVERWATCH project.

M19: Potential additional features will be analysed, and a roadmap defined.

Once the Management backend has been integrated into the AR backend application testing will start with the AR Frontend as well.

3.6. Accurate and secure drone navigation

The provided “component” is not a software service but a physical GNSS receiver to be integrated onto the drone. The integration of the receiver with the navigation system will be under the drone manufacturer’s objectives. Due to its nature, the usual sections do not apply to this hardware device, therefore paragraphs such as “interaction diagram” are not present in this chapter.

3.6.1. Testing requirements

Id	Description	Family of services	Service to test

#1	Test the validity of the outputs of the GNSS receiver selected under nominal conditions	Accurate Positioning	Create position request
#2	Test the validity of the outputs of the GNSS receiver selected using OSNMA under nominal conditions	Accurate Positioning	Create position request with OSNMA
#3	Test the validity of the outputs of the GNSS receiver selected using OSNMA under interfered conditions	Accurate Positioning	Create position request with OSNMA under interference

3.6.2. Performance requirements

Id	Description	Family of services	Service to test
#1	The module shall provide accurate positioning information	Accurate Positioning	Ability of vehicle to use the outputs from the GNSS receiver correctly

3.6.3. Reliability requirements

Id	Description	Family of services	Service to test
#1	Under high interference the module should alert the system	Accurate Positioning	Navigation capailities under interference

3.6.4. Availability requirements

Id	Description	Family of services	Service to test
#1	The GNSS receiver module should provide on its own the processing power needed for calculating the PVT solution	Accurate Positioning	Data processing

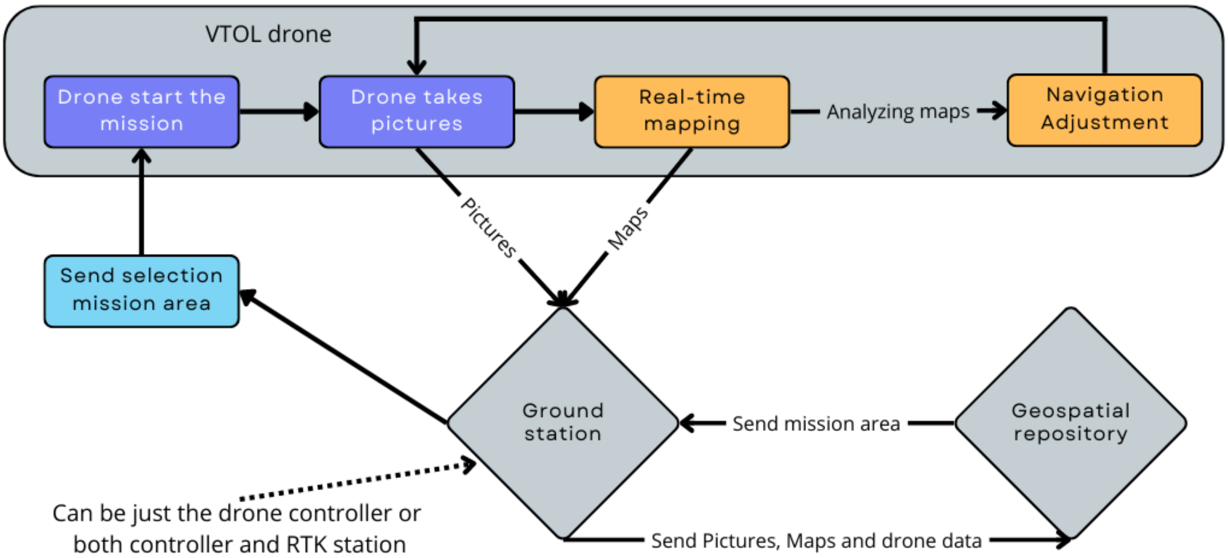
3.6.5. Roadmap of the release of the module

- M9: State of the art and literature review wrap-up
- M12: Prototyping the positioning module, selection of GNSS receiver.
- M18: Initial integration with the UAV, preliminary analysis on inter-compatibility of systems
- M18 onwards: Iterative improvements

3.7. Drone terrain mapping

3.7.1. Workflow diagram of the tool and operational concept

Figure 12 - Drone terrain mapping operational concept.



The workflow diagram presents the operational process for terrain mapping by a Vertical Take-Off and Landing (VTOL) drone in wildfire and flood scenarios, aiming for rapid map production. The user initiates the process by selecting a specific area to map, either obtained from an existing geospatial repository or chosen by the drone pilot through the ground station's user interface.

The ground station, which could comprise the drone controller, or both the drone controller and Real-Time Kinematic (RTK) station, communicates the selected area to the VTOL drone. The drone subsequently generates a lawnmower-like navigation pattern and begins its vertical take-off to start mapping the chosen area.

Throughout the operation, the drone continuously captures images, transmitting them to a real-time mapping module. This module processes the data immediately, rectifying the images and constructing an orthographic map. This real-time map is then analysed to identify areas of interest, such as wildfire or flood zones. Based on these areas, the drone adjusts its navigation to further explore the relevant areas.

In scenarios like wildfires, due to potential high-temperature threats, the drone maintains a safe distance, tracing the wildfire's perimeter. This allows for rapid mapping of the affected area's borders, constantly adjusting navigation to focus on areas of interest. However, the user can choose to discontinue these navigation adjustments and simply map the area following the original pattern.

The images captured and the maps created are transmitted to the geospatial repository. The real-time maps are used for swift decision-making by the incident commander, while the maps' precision is compromised for speed.

Considering the potential bandwidth constraints of transmitting a large volume of real-time images from the drone, the system supports data transmission over 4G when available. Additionally, there's support for transmitting images after the mission if required.

3.7.2. Technical requirements

3.7.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	Test the creation of a mission after being triggered the visual alert by the operator in a scarce connectivity context	VTOL drone system	Create Mission
#2	Test the drone's performance in various visibility conditions, including day, low-light/low-visibility, fog or smog, night.	VTOL drone system	Drone Sensors
#3	Test the drone's ability to switch between autonomous and piloted flight modes under diverse conditions.	VTOL drone system	Drone Navigation
#4	Test the drone's ability to return to base or land safely in case of a loss of communication.	VTOL drone system	Drone Safety
#5	Test the drone's capability to relay or stream data to the Ground Control Station (GCS).	VTOL drone system	Data transmission
#6	Verify the drone's ability to detect adverse weather conditions, such as strong winds, and provide relevant information to the GCS for informed decision-making.	VTOL drone system	Drone Sensors
#7	Test the real-time mapping module deployed on the onboard computer, including its ability to make automatic adjustments to the	VTOL drone system	

	drone's navigation path.		

3.7.2.2. Performance requirements

Id	Description	Family of services	Service to test
#1	Maximum fight time with the payload.	VTOL drone system	Drone Specifications
#2	Maximum flight range with payload	VTOL drone system	Drone Specifications
#3	Wind speed resistance.	VTOL drone system	Drone Specifications
#4	Radio communication distance	VTOL drone system	Drone Specifications
#5	The system should generate and stream real-time maps to a ground station within a specified timeframe	VTOL drone system	Data transmission

3.7.2.3. Reliability requirements

Id	Description	Family of services	Service to test
#1	The drone's autonomous flight and navigation system should maintain consistent performance in diverse conditions.	VTOL drone system	Drone Navigation
#2	The drone's safety features, such as return-home and safe landing protocols, should function reliably in case of loss of communication.	VTOL drone system	Drone Safety
#	The drone should withstand light rain, dust and smoke.	VTOL drone system	Drone Safety

3.7.2.4. Availability requirements

Id	Description	Family of services	Service to test
#1	Checking the drone's ability to reliably transmit collected data to the geospatial repository, whether over 4G or via a ground station.	VTOL drone system	Data transmission
#2	Ensuring the drone can effectively interface with other modules in the OVERWATCH project, including data sharing and coordination.	VTOL drone system	Data transmission

3.7.3. Interactions with the other modules

The VTOL drone system in the OVERWATCH project is responsible for gathering high-resolution mapping data of different areas. This data, including images, maps, and drone telemetry, is then sent to a geospatial repository. The data transfer can occur over a 4G network if available, or via a ground station if not.

Other modules within the project can access the necessary data from this repository. In other words, the geospatial repository serves as a hub for data exchange.

Furthermore, the drone system integrates with the positioning module provided to Robotto. This module helps accurately determine the drone's location.

3.7.4. Roadmap of the release of the module

M18 - Communication system from and to geospatial repository

M20 – Real time mapping

M22 – Navigation system with GNSS module.

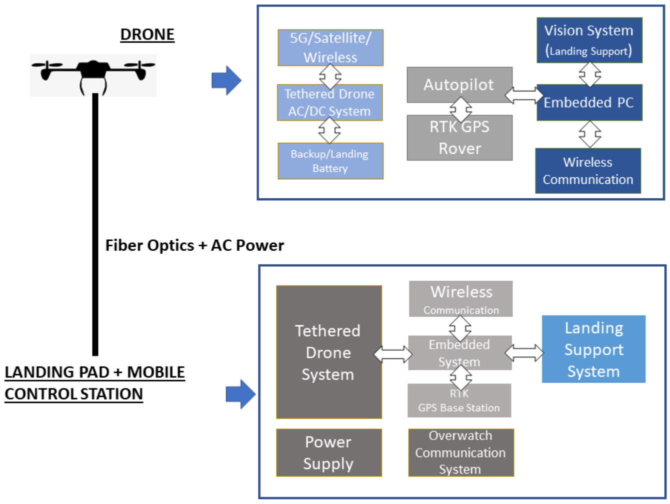
M24 - VTOL drone

M28 - Full integration of VTOL system

3.8. Fallback connectivity

3.8.1. Workflow diagram of the tool and operational concept

Figure 13 - Workflow diagram of the module and integration logic.



The operation of the tool is completely independent and transparent to the OVERWATCH local and cloud systems. The tool is deployed strategically, on site, avoiding nearby obstacles such as trees and buildings, and placed in close proximity to other drone GCSs and possible on-site command centre. These conditions allow for providing these client systems with broadband wireless communications, or even wired if necessary/feasible due to physical constraints.

The systems connected to this tool will be able to interact seamlessly with any Internet endpoint, such as the OVERWATCH cloud-based systems.

Note: While this tool is independent and transparent for the OVERWATCH system as a whole, it will provide some information so that its status is accessible through the cloud-based system interface, such as its location and the condition of its wireless communications links.

3.8.2. Technical requirements

3.8.2.1. Testing requirements

Id	Description	Family of services	Service to test
#1	Tethered drone flight capability	Fallback Connectivity System	All functions that make the tethered drone capable of flying e.g., batteries, motors, autopilot, missions
#2	Tethered system enrollment	Fallback Connectivity System	Deploy and land the drone with tethered cable
#3	Tethered system power and fiber-optic	Fallback Connectivity System	Provide power and connectivity to the drone
#4	Fallback connectivity CGS functions	Fallback Connectivity System	Control the drone and provide connection to external systems

#5	Tethered drone sensors	Fallback Connectivity System	Provide data and video to the GCS
#6	Wireless Network Broadcast	Fallback Connectivity System	Check if fallback network is properly broadcasting a local wireless signal
#7	LEO Satellite Connectivity	Fallback Connectivity System	Check if the LEO satellite connection service is operational and providing a functional backhaul Internet link
#8	Cloud backend Connectivity	Fallback Connectivity System	Check if connection with cloud system is established

3.8.2.2. Performance requirements

Id	Description	Family of services	Service to test
#1	Tethered drone autonomy	Fallback Connectivity System	Test tethered drone range and autonomy
#2	Tethered drone maximum altitude capability	Fallback Connectivity System	Test tethered drone maximum altitude
#3	Communications maximum throughput	Fallback Connectivity System	Test the maximum network throughput achievable for each communications link technology (local access and backhaul links)
#4	Communications maximum coverage range	Fallback Connectivity System	Test the maximum coverage range of each wireless communications access technology

3.8.2.3. Reliability requirements

Id	Description	Family of services	Service to test
#1	The Fallback connectivity system should be able to provide at least 2 alternative communications means (Wi-Fi and Ethernet, and	Fallback Connectivity system	Communications links (backhaul and local access) Reliability and Resilience

	if possible 5G) for establishing local connectivity with nearby systems in a reliable manner		

3.8.2.4. Availability requirements

Id	Description	Family of services	Service to test
#1	Tethered Drone Power availability	Fallback Connectivity System	Power-on of the communications payload and ground components
#2	Communications minimum network throughput	Fallback Connectivity System	Test if the minimum expected network throughput for each access and backhaul link technology is being provided
#3	Communications minimum coverage range	Fallback Connectivity System	Test if the minimum expected coverage range for each wireless communications access link is being maintained
#4	Communications delay below maximum threshold	Fallback Connectivity System	Test if the local and Internet connection delay are below the maximum delay threshold tolerable by the systems.

3.8.3. Interactions with the other modules

The Fallback Connectivity System is fully transparent to the other OVERWATCH sub-modules that will only have to connect to a Network (through Ethernet or wireless) depending on distance that will establish a connection between the sub-modules e.g., drones and OVERWATCH C2. The Fallback Connectivity System can also establish connection to other legacy systems if available. For example, if a VoIP-TETRA gateway is made available (either locally or on the Internet) by the emergency network provider, IP terminals connected to the Fallback Communications Network could interface with legacy voice-based TETRA terminals by using push to talk functionality). Possible connectivity scheme is explained in more detail in section 4.

3.8.4. Roadmap of the release of the module

The fallback connectivity system will be released in the following timeline:

M18 - Tethered System GCS and cable Deployment

M24 - Tethered Drone

M24 - Communications Systems (Payload and Ground Components)
M28- Full Fallback Communication System

4.Connectivity assumptions

This section details the connectivity options considered in OVERWATCH for enabling the upload of sensor data acquired from the UAVs to the Cloud/Control Room Premises using an available Internet connection.

There are two main possibilities for uploading the data collected by the UAVs through the Internet: (a) by default, using the Cellular Connectivity provided by the telecom operators with available infrastructure in the area of interest; and (b) as backup, using the Fallback Connectivity Service developed in the context of T2.4 of OVERWATCH. The latter is a key component of OVERWATCH since it allows for uploading data using an Internet LEO satellite backhaul link even if the existing communications infrastructure provide insufficient coverage, gets damaged (e.g., by fire), or becomes overloaded.

All connectivity options share the same base components, represented in the diagrams of Figure 14, Figure 15 and Figure 16. At the left of the diagrams are shown the UAV that collects sensor data, the UAV controller, and the UAV connectivity HUB. Typically, the UAV flies in the radio range of the UAV controller and uploads the collected data through that radio link. The UAV controller is, then, connected to the UAV connectivity HUB, which is connected to the Internet. At the centre of the diagrams is represented the Internet as a “Cloud” in which the OVERWATCH serviced for data collection and processing are running. And finally, at the right are represented the operator terminals, such as AR glasses and computers, which are typically data consumers. The Internet connectivity creates an abstraction allowing the UAVs (data producers), the cloud services (data storage/processing), and the operator terminals (data consumers) to operate either at the same premises or distributed anywhere in the world provided that there is an Internet connection available.

Figure 14 focuses on the network connections when using existing cellular connectivity, which means that, in this case, the UAV Connectivity Hub is dependent on the cellular coverage of the existing telecom operators. This Figure also shows that in some cases, the UAV itself can have an onboard cellular modem and upload its data directly to the Internet through the cellular link, instead of relaying the data using the link between the UAV, UAV Controller and UAV connectivity Hub.

Figure 14 - Network Connections when existing cellular connectivity is used to collect UAV data.

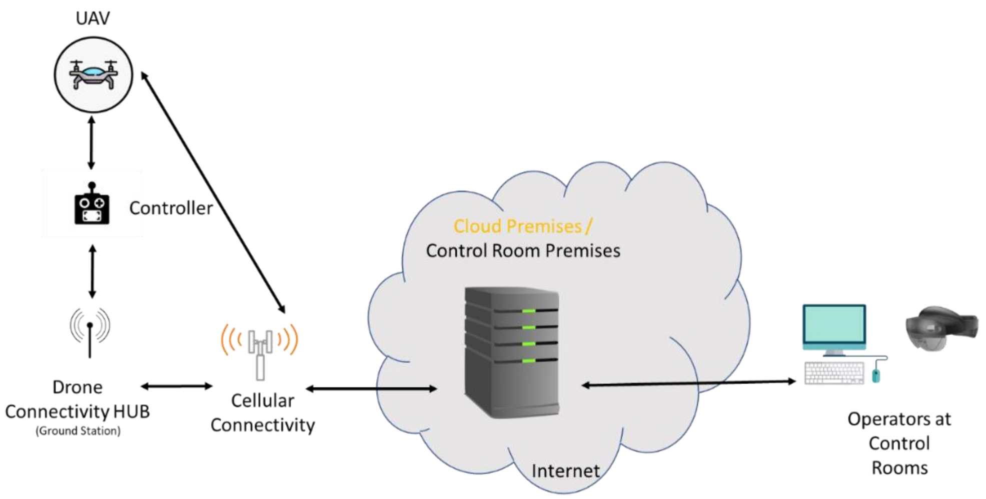


Figure 15 concentrates on the network connections when using the Fallback Communications System. In this case, the UAV Connectivity Hub is no longer dependent on the cellular coverage of the existing telecom operators and connects to the Internet using the LEO satellite backhaul through the provided local Ethernet, Wi-Fi, and potentially 5G connectivity. This Figure also shows that in some cases, the UAV Controller itself can have an onboard Wi-Fi network interface and upload its data directly to the Internet through the Wi-Fi link provided by the FCS, instead of relaying the data using through the UAV connectivity Hub. Note that if an on-site control room is deployed, the FCS can also provide connectivity to the operator's terminals.

Figure 15 - Network Connections when Fallback Connectivity using LEO Satellite backhaul is used to collect UAV data.

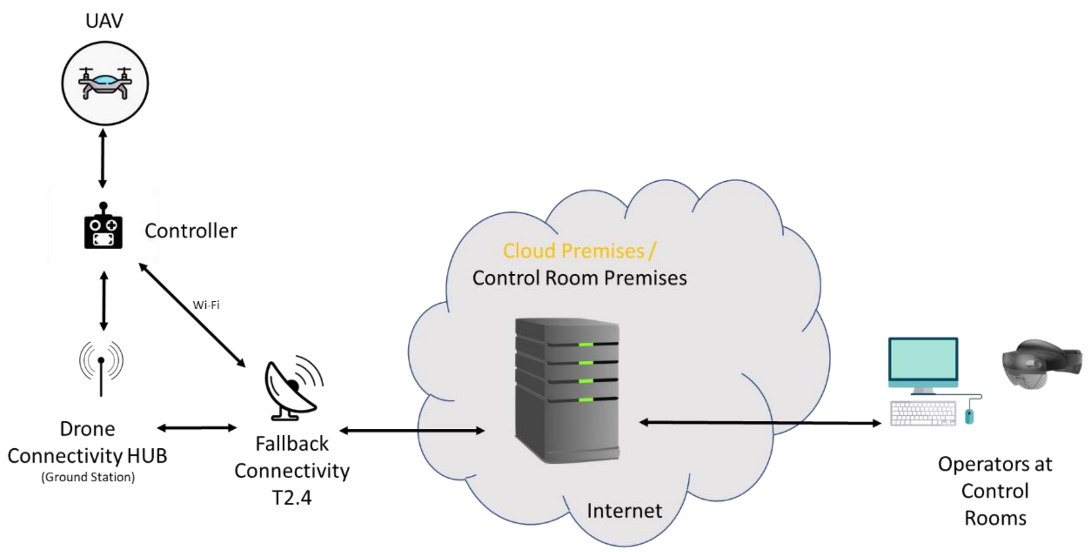
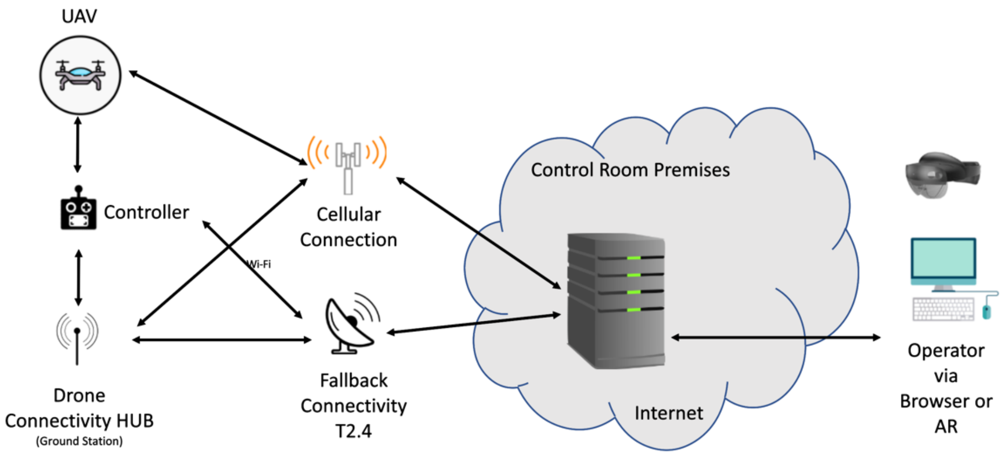


Figure 16 shows a combination of the multiple possible connectivity options, providing the necessary redundancy for resilient system operation. Note that the arrow between the UAV Connectivity Hub and the FCS is, in fact, representing up to 3 simultaneous redundant links: Ethernet, Wi-Fi and potentially 5G.

Figure 16 - Network Connections considering both Cellular and Fallback Connectivity options to collect drone data.



5. Conclusions

The present report is the deliverable "D1.2 - Functional / Technical requirements & System architecture", which focuses on providing the basis for the system architecture of the entire OVERWATCH project. The deliverable provides the explanation of each module that belongs to the architecture, specifying the fulfilment of the requirements defined in "D1.1 - End user requirements" by each tool of the entire system.

OVERWATCH modules are distinct units of code that encapsulate specific functionality or components of a system. The interaction among them refers to how these modules communicate and work together to achieve the overall objectives of the needs requested by the two realistic scenarios such as floods and wildfire.

The deliverable describes how modules interact through well-defined interfaces and communication channels. They exchange data, messages, or function calls to pass information and trigger actions. Furthermore, modules may depend on each other to fulfil their functionalities. This dependency can be in the form of data dependencies, where one module requires input or output from another module, or functional dependencies, where one module relies on the behaviour or results of another module, and this deliverable shown how this take place.

Modules are designed to have clear boundaries and encapsulate specific functionality. They provide a defined set of public interfaces that expose their functionality to other modules while hiding their internal implementation details. This promotes modularity, abstraction, and reusability. The interaction between modules can be influenced by coupling and cohesion. Coupling refers to the level of interdependence between modules, where low coupling indicates loose coupling and high cohesion implies that each module focuses on a specific and well-defined task. Modules with low coupling and high cohesion are easier to understand, maintain, and modify.

In synthesis, deliverable D1.2 provided the following information:

- The description of the technical modules of entire architecture and their covering of functional requirements in chapter 2.
- The interactions with the other modules through a technical explanation such as data flow diagram and a set of requirements covered in chapter 3.
- The connectivity options and assumptions for enabling the upload of data from drones towards the control room in chapter 4.