H2020 – Secure societies - Protecting freedom and security of Europe and its citizens
SU-DRS02-2018-2019-2020– Technologies for first responders – Research and Innovation Action (RIA)

Emerging technologies for the Early location of Entrapped victims under Collapsed Structures & Advanced Wearables for risk assessment and First Responders Safety in SAR operations

D7.6 Integrated S&R platform 2nd version

Workpackage:	WP7-S&R Platform Design, Development and Service Integration
Authors:	SIMAVI
Status:	Final
Due Date:	31/08/2022
Version:	1.00
Submission Date:	31/08/2022
Dissemination Level:	PU

Search and Rescue Project Profile

Grant Agreement No.:	882897
Acronym:	Search and Rescue
Title:	Emerging technologies for the Early location of Entrapped victims under Collapsed Structures & Advanced Wearables for risk assessment and First Responders Safety in SAR operations
URL:	www.search-and-rescue.eu
Start Date:	01/07/2020
Duration:	36 months

Partners

	NATIONAL TECHNICAL UNIVERSITY OF ATHENS (NTUA) Co-ordinator	Greece
	AIDEAS OÜ (AIDEAS)	Estonia
	SOFTWARE IMAGINATION & VISION S.R.L (SIMAVI)	Romania
	MAGGIOLI SPA (MAG)	Italy
	KONNEKT-ABLE TECHNOLOGIES LIMITED (KT)	Ireland
	THALES ITAIA Italia SPA (THALIT)	Italy
	ATOS IT SOLUTIONS AND SERVICES IBERIA SL (ATOS)	Spain
	ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS (CERTH)	Greece
	UNIVERSITA DEGLI STUDI DI CAGLAIRI (UNICA)	Italy
	UKEMED GLOBAL LTD (UGL)	Cyprus
	PUBLIC SAFETY COMMUNICATION EUROPE FORUM AISBL (PSCE)	Belgium

	UNIVERSITA DEGLI STUDI DI FIRENZE (UNIFI)	Italy
	DEUTSCHES FORSCHUNGSZENTRUM FUR KUNSTLICHE INTELLIGENZ (DFKI)	Germany
	UNIVERSITA CATTOLICA DEL SACRO CUORE (UCSC)	Italy
	VRIJE UNIVERSITEIT BRUSSEL	Belgium
	SYNYO GmbH (SYNYO)	Austria
	UNIVERSITEIT HASSELT (UHASSELT)	Belgium
	SPOLECZNA AKADEMIA NAUK (SAN)	Poland
	GIOUMPITEK MELETI SCHEDIASMOS YLOPOIISI KAI POLISI ERGON PLIROFORIKIS ETAIREIA PERIORISMENIS EFTHYNIS (UBITECH)	Greece
Search and Rescue End-Users		
	ELLINIKI OMADA DIASOSIS SOMATEIO (HRT)	Greece
	ENOSI PTYCHIOYCHON AXIOMATIKON YPAXIOOMATIKON PYROSVESTIR OY SOMATEIO (EPAYPS)	Greece
	JOHANNITER-UNFALL-HILFE EV (JOHAN)	Germany
	JOHANNITER OSTERREICH AUSBLIDUNG UND FORSCHUNG GEMEINNUTZIGE GMBH (JOAFG)	Austria
	CONSIGLIO NAZIONALE DELLE RICERCHE	Italy

POMPIERS DE L'URGENCE INTERNATIONALE PUI FRANCE / UNITED NATIONS INSARAG Classified MEDIUM USAR Team 2010-2015	POMPIERS DE L'URGENCE INTERNATIONALE (PUI)	France
Romanian Cluster PROECO - CBRNE	ASOCIATA CLUSTERUL ROAMN RENTRU PROTECTIE SI ECOLOGIE IN DOMENIUL MATERIALELOR CHIMICE, BIOLOGICE, RADIOLOGICE/NUCLEARE SI EXPLOZIVE (PROECO)	Romania
Servicio Madrileño de Salud SERMAS SaludMadrid	SERVICIO MADRILENO DE SALUD (SERMAS)	Spain
SaludMadrid FIIBAP FUNDACIÓN PARA LA INVESTIGACIÓN E INNOVACIÓN BIOSANITARIA DE ATENCIÓN PRIMARIA Servicio Madrileño de Salud	FUNDACIÓN PARA LA INVESTIGACIÓN E INNOVACIÓN BIOSANITARIA DE ATENCIÓN PRIMARIA (FIIBAP)	Spain
ESCUELA E.S.D.P. SALVAMENTO Y DETECCION CON PERROS	ESCUELA ESPANOLA DE SALVAMENTO Y DETECCION CON PERROS (ESDP)	Spain

Document History

Version	Date	Author (Partner)	Remarks/Changes
0.10	14/07/2022	Razvan Gliga (SIMAVI)	Table of Contents
0.20	21/07/2022	Razvan Onet (SIMAVI)	Add description for S&R Integration Dashboard
0.21	22/07/2022	Razvan Gliga, Razvan Onet, Horia Dascalu, Alin Badea (SIMAVI)	Updates for 2.4.5 and 2.5.1 chapters
0.30	26/07/2022	Razvan Gliga, Razvan Onet, Horia Dascalu, Alin Badea (SIMAVI)	Added new 3.4-chapter, Message Broker Kafka
0.40	05/08/2022	Elena Politi (UBITECH)	Internal Review
0.41	14/08/2022	Marta Maini (UNIFI)	Internal Review
0.50	29/08/2022	Razvan Gliga (SIMAVI)	Corrections based on comments
0.60	31/08/2022	Christodoulos Santorinaios (NTUA)	Quality Control
1.00	31/08/2022	Christos Ntanos (NTUA)	FINAL VERSION TO BE SUBMITTED

Executive Summary

The current document presents the description of the final S&R integrated platform. It integrates all the developed software components and communication interface of the hardware components and sensors to form the final S&R platform as a whole.
Current integrated S&R platform extended and refined the initially developed S&R platform is described in this document. A user's manual will also be produced in order to facilitate the deployment and maintenance of the S&R platform.
According to DoA the S&R platform of technologies and solutions is split into the following three categories:

- **Front-End equipment systems and collection tools**: which are mainly based on technologies employed in the crisis scenarios in order to acquire relevant and sensible information in real time (e.g., sensors, drones, rescue robots, smart glasses, smart uniforms, first aid devices, and masks).
- **Data Fusion and Mediation Systems**: which merge the information coming from heterogeneous sources and provide a more detailed and accurate situational\context awareness to the decision makers and to the C&C centres. Within S&R the integrated version of CONCORDE EMS offers the particular functionality, but other systems may be also used.
- **Back-End applications, services and portals**: this provides decision support capabilities to crisis stakeholders.

The platform is now equipped with heterogeneous autonomous assets and sensors that provide various sources of data, indoor positioning and extensions of communications efficiently by employing swarming behaviours. S&R platform contains now all the features that end-users identified and classified as highly important during the piloting of CONCORDE and IMPRESS and enhance them with the latest developments in First Responders' related technologies.
In the first part of this deliverable, the objectives and scope are briefly described. In the second part, the components of the Integration Framework – architecture and tools are presented; data sources and hosting infrastructure are also detailed.
In addition, the presentation of how to use the Apache Service Mix tool can be found, describing how it helps the conversion of all web services involved in the S&R project and offering them from a common base URL.
Also, in this updated version the presentation of the dashboard, information about Kafka, and description of the Data Lake transfer were added.
The last part presents the conclusions of this deliverable.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Explanation
API	Application Programming Interface
DSS	Decision Support System
ECG **EMG**	Sensors designed to measure voltage waveforms produced during muscle contraction
EMS	Emergency Medical Service
ERM	Emergency Response Health Condition Monitor
ESB	Enterprise Service Bus
GNSS	Global Navigation Satellite System
JSON	JavaScript Object Notation
KPI	Key performance indicators
LMS	Learning Management System
M	Month
MQTT	MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT)
ODS	Object Detection System
OFET	Organic Field Effect Transistor
ROS	Robot Operating System
SCORM	Sharable Content Object Reference Model
SOAP	Simple Object Access Protocol
SOT	Strategic Operational and Tactical
S&R	Search and Rescue
TCP/IP	Transmission Control Protocol/Internet Protocol
UAV	Unmanned Aerial Vehicle
UC	Use Case
VR	Virtual reality

1 Introduction

1.1 Document Objectives and Scope

This document describes the activities conducted in the Search and Rescue Project, under Task 7.5 S&R platform component and service integration and presents the approach for the integration of the developed software components and communication interfaces of the hardware components and sensors.
The document presents a description of the second version of the Integrated S&R platform.
The final integration plan between the various tools that comprise the S&R platform is also presented in this deliverable.

1.2 Relationship with other deliverables

This deliverable is linked to the following deliverables:

- D3.8 Situation Awareness Model – specification v2
- D3.3 BIM based services and applications – review and service design
- D3.5 Data-driven analytics applied on UAV imagery using deep learning
- D4.9 Design of SOT DSS component v2
- D4.10 Design of PHYSIO DSS component v2
- D4.11 Development of SOT DSS component v2
- D4.12 Development of PHYSIO DSS component v2
- D5.5 Design & development of the RESCUE MIMS v2
- D6.2 Voice, data and services Interoperability frameworks
- D6.3 Presentation and analysis of the designed S&R interoperability framework
- D6.7 S&R Training System
- D7.3 Component Interface specifications for interoperability within S&R
- D7.5 Integrated S&R platform 1st version
- D7.7 S&R Legal and Security infrastructure 1st version
- D7.11 S&R extensive service catalogue 2nd version
- D7.12 Architecture and Design Specifications of S&R platform v2

2 Integration Framework components – architecture & tools

2.1 App context

The Search and Rescue platform is aimed at reducing the negative impact of various disasters on human existence. Taking into consideration the wide range of scenarios in which natural and premeditated events happen, we are planning to develop a platform which will help the First Responders and rescuers in the intervention process.
A series of pilot scenarios will be put in practice in order to get relevant information which will help us in the development and optimisation process. These scenarios have the role of helping to understand the case and the outcome depending on the type of event. Taking into consideration all the information which we will have at our disposal, data gathered after the pilots, information coming from historical databases or technological infrastructure coming from the COncORDE project, we will develop an interoperable framework.
Its main purpose is to help in the intervention process, by receiving real time information and sending alerts which are meant to help the first responders for quick localisation of the incident, but also to provide data about the incident, like the time and place, the number and the health state of the victims, the environment and so on.

2.2 Integration plan

An integration plan was prepared and discussed in order to guide the integration of the S&R components.
The proposal was to have two main platforms: data bus (real time data for Data Lake and Concorde DB and S&R DB) and ESB for service integration.
This integration phase is aimed at covering the following aspects: the connections with one external data source, the integration with one field data source which can be simulated and the creation of one Knox mock-up temporary instance for Data Lake connection.
The below diagram was designed after having several work package meetings with the technical partners and it represents the connection of all components which will be part of the final integration framework, the data sources and the used tools and frameworks.

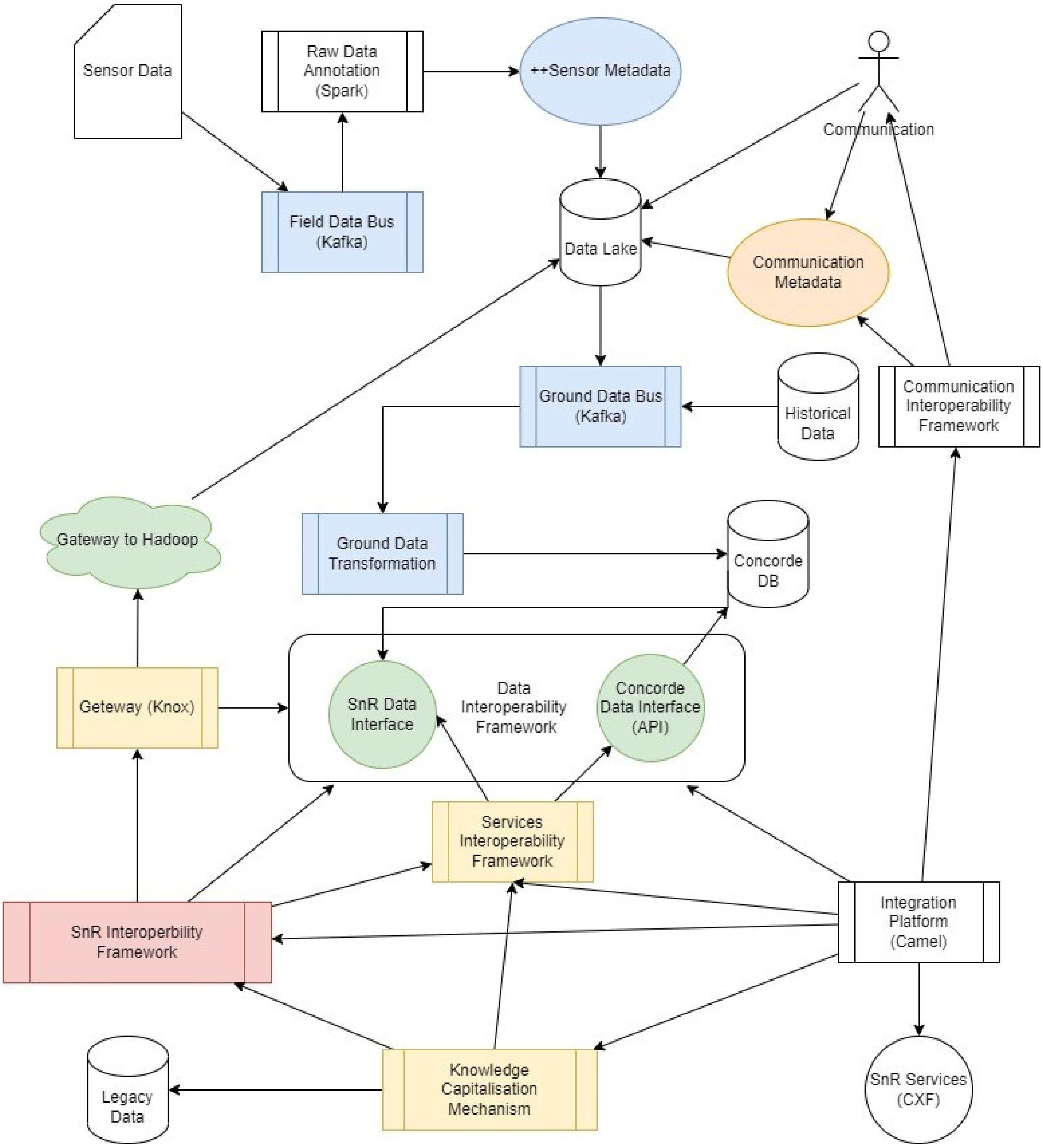

Figure 2.1 Integration plan

2.3 Component description (software, hardware, sensors)

This section presents a brief description of each component that will be integrated into the S&R platform.
Details on interface specifications and data format for each component can be found in D7.3 Component Interface specifications for interoperability within S&R.

2.3.1 RESCUE-MIMS

RESCUE-MIMS is a chemical device which can be used not only as an early warning system for the rescuers (safety purposes), but also for the near real-time detection of chemical compounds that can be found under the ruins and possibly be correlated with human presence (search and rescue purposes), based on literature recordings.

RESCUE-MIMS is portable and can be used either as a handheld device or as a payload, on-board robotic platforms, like DFKI's ground robot, serving as a screening tool of hazardous environment in the disaster scene. It is foreseen to be tested inside the S&R project under UC4 –Industrial fire scenario and UC5-Trapped people under rubble scenario; on-line monitoring of the abundance of chemical compounds, such as benzene, acetone etc.
RESCUE-MIMS produces csv files that include parameters, such as the data and time, masses with different increments, e.g. 0,1 atomic mass unit (amu) etc. The connection of the RESCUE-MIMS with the DSS system will be achieved as follows: (a) the RESCUE-MIMS device will communicate with a connector (e.g., router installed inside ambulances) using agreed protocols (e.g., Bluetooth, Restful APIs) in order to transfer data through an Enterprise Service Bus (ESB) to the data lake of S&R for data aggregation. Those data will be filtered using a canonical model created in T6.2 "S&R Data Communication Interoperability framework". In particular, the canonical model will facilitate the process of filtering out the non-relevant data as well as the transformation of the data to the required format (b) Then the data will be preprocessed, clustered, and empowered with machine learning algorithms that can facilitate decision making (e.g., detection algorithms). Upon their processing and transformation, the data will be retrieved as homogeneous data by both the Concorde Platform and the DSS, in order to help first responders with decision making.
This component is provided by NTUA-LPAD partner.

2.3.2 Smartwatch

YAMAY Smart-watch[1] was chosen by KT to be tested and provide valid information, such as heart rate monitor, timestamp, average speed among other valuable data, to the S&R use cases.

Figure 1.2 Yamay Smartwatch

The Smartwatch is integrated via Bluetooth on the first responder's smartphone, in order to get synchronized.
Considering the S&R's needs in every operational level, the aforementioned smartwatch will serve as a wearable device to the Use Cases. More specifically, first responders will wear them, in order to feed the final system with their health data aiming to avoid any danger. Moreover, there is the option to take health measurements from the patients and victims on the field by putting the smartwatch on their wrist too. With this approach the final platform will be fed with:

- first responders,
- patients,
- and patients' health data.

This will aid operational procedures, such as the situation awareness module and the decision support system, to process more accurate and valid data. As a result, the actors of S&R operation, such as the Command Center, the Field Commander, the EMS units and the first responders themselves will

be more aware and notified for potential grave situations on the field (e.g., Tachycardia that can cause stroke, sudden cardiac arrest or even death).
In order to achieve all the aforementioned KPIs and the integration to the platform, there are requirements that must be adapted, such as:

- Smartphones supporting android, google services and Bluetooth communication
- Installation of VeryFitPro app to the smartphone
- Installation of Strava app to the smartphone
- Synchronization of smartwatch to VeryFitPro and Strava
- Strava API produce the field data (received from smartwatch) to specified Apache Kafka Topic

Then, the rest system components will be able to:

- retrieve/consume,
- filter,
- process,
- aggregate,
- and detect the following data:


```
"activity_id": 5698997168,
"UTC_timestamp": "2021-07-27 18:12:19+00:00",
"local_timestamp": "2021-07-27 21:12:19",
"timezone": "Europe/Athens",
"location_city": "Athina",
"location_state": "Attica",
"location_country": "Greece",
"moving_time": "2:37:45",
"elapsed_time": "2:37:45",
"average_heartrate": 85.3,
"max_heartrate": 134.0,
"average_speed": 0.4,
"max_speed": 0.4
```


Figure 2.2 Smartwatch data in JSON format

The above JSON will host data coming from all the smartwatches, classified with different ids, in order to facilitate processes and functionalities of the rest of the platform and the operation.
More details can be found in "Chapter 2.2 Smartwatch" of "D7.3 Component Interface specifications for interoperability within S&R".

2.3.3 Emergency Response Health Condition Monitoring Device

2.3.3.1 Role in S&R Project

The Emergency Response Health Condition Monitor (ERM), provided by CERTH objective is to monitor the victim's health condition from the point they are discovered until they are transferred to an ambulance for treatment from medical professionals. The ERM is a compact patient monitor with internet connectivity that can be applied very fast in the field with minimal interference to the motion of the wearer.

Patient monitors purpose is to monitor over time the health condition of patients at hospitals. This is performed by continuously measuring the vital signs of patients lying on beds presenting the values in the display and communicating them to a central unit. The main components of the patient monitor are the sensors, the transducer, the display and the communication device. Mobile monitoring devices usually in combination with smartphones and wearables in Body Area Networks are used for medical and not medical purposes for monitoring for longer periods of time in the natural environment of the wearer. The purpose of the ERM is to develop a compact mobile monitor that incorporates all the functions of a hospital patient monitor.

2.3.3.2 API Specifications

Data communication with external sources is implemented with the MQTT protocol. MQTT protocol is a publish-subscribe network protocol for IoT devices. MQTT is lightweight and supports lossless communication. The MQTT protocol is functioning over TCP/IP protocol. The MQTT communication consists of a server message broker and clients which subscribe and publish in different topics. The implementation of the MQTT communication is performed inside the smartphone application.

2.3.3.3 Services

The ERM smartphone application does not provide any services. Its sole purpose is to digitize and display biosignals and create and publish MQTT messages of these signals.

2.3.3.4 Data format

Data communication with external sources is implemented with the MQTT protocol. MQTT protocol is a publish-subscribe network protocol for IoT devices. The MQTT protocol is functioning over TCP/IP protocol. The messages that are used in this application concern the position and state of the rescuers and victims
A message for a victim is a string that can be formulated as follows: 1623067157; v; 1; 40.56698565; 22.99799482; 40.644145; 22.936301; m; 12; l; 60; 12 with each sub-string separated by a semicolon having the following meaning:

Timestamp: 1623067157
type: (v)ictim
ID: 1
Latitude : 40.56698565
Longitude : 22.99799482
Destination latitude : 40.644145
Destination longitude : 22.936301
sex: male
age: 12
position: (l)ying down
heartrate (bpm): 60
respiration rate (rpm): 12

2.3.4 Rescue Robots & Autonomous Vehicles

A detailed description of the Robot Platform used in the S&R project is provided in section 3.3.1 of D5.4, while the directly available interfaces are provided in D7.3.
The planned robot platform to be used in this S&R project is the DFKI version of the SeekurJr. It is a very robust outdoor system which can especially be used on uneven, e.g., rocky or sandy ground, or urban environments. On heavy terrain it may struggle due to the low ground clearance. It also is not fully waterproof in the current state, hence rain, wet roads and muddy terrain should generally be avoided. Originally, the system is used for the primary research tasks of autonomous self-evaluation and the detection of unspecified failures and disturbances.
The rover is originally equipped with a SICK LMS-111 LiDaR scanner for long range mapping, and extended by a sensor tower at the front, which holds a periodically tilting Hokuyo UTM-30LX-EW LiDaR scanner for close proximity obstacle detection at an extended height, plus a pan-tilt unit with a mounted Velodyne VLP-16 LiDaR scanner and two IDS UI-5240CP Rev 2 GigE cameras for more dynamic perception capabilities including stereo vision. The peripheral components are connected via two switches using Gigabit-Ethernet or USB.
To be able to turn on the spot, the robot is skid-steered by two motors - one on each side. It has a swing radius of 52cm, a maximum linear velocity of 1.2m/s and a maximum traversable grade of 75%.
The rover can run for 2-3 hours on average until the batteries are depleted. This value depends on the rover's tasks and loads. For an extended operation duration, the batteries can be hot swapped manually during a mission, as long as the white cover on the robot base remains freely accessible.
The battery provides regulated power of 2A at 5V, 2A at 12V, and 1A at 24V. It also provides 20A of unregulated power at a nominal voltage of 24V which will be used for powering the ODS.

For autonomy and perception, the robot includes three PCs. Besides navigation and plan execution, the semi-autonomous control particularly includes the prediction of sensor values (generation of expectations) and the self-evaluation (detection of unexpected or unknown situations).

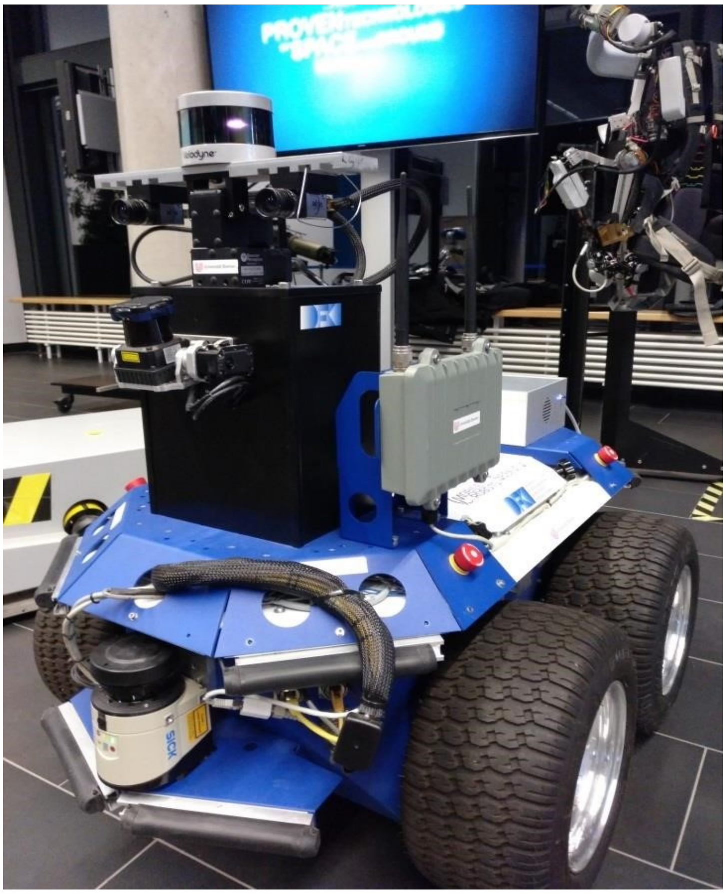

Figure 2.1 Robot

The robot has four easy to access full stop buttons in case of emergency, which halt the entire operation of the robot. In Figure 2.4, the modified platform is shown in its current state. Additional interfaces needed for the integration are elaborated in the following subsections.

2.3.4.1 Required Hardware interfaces

To mount the RESCUE MIMS (2.3.1) and ODS (2.3.20) on the robot platform, brackets have to be provided, which maintain a good accessibility to the white battery cover on top of the platform, such that the batteries can be hot-swapped during operation. All devices are then connected to an Ethernet hub, which provides access to a remote connection to a control system.

2.3.4.2 Required Software interfaces

Additionally, to the ROS messages sent by the robot as declared in 7.3, the robot needs a visual interface, which displays a spatial map of the environment and the corresponding measured gas values by the RESCUE MIMS. In this map, it should be possible to select areas, which the robot has to autonomously explore. For the control input, it needs a control interface which transform a directional input by the user to a ROS message which contains directional velocities.

2.3.5 Wearable GPS tracker

The Global Navigation Satellite System (GNSS) module, provided by UNICA partners, is integrated into the smartphone used by the first responder. The model of the module, therefore, depends on both the type and the brand of the smartphone adopted. Unfortunately, many smartphone manufacturers do not provide accurate information on the integrated GNSS module. A Samsung S10 smartphone was used in the preliminary tests, with the Broadcom[2] BCM47752 GNSS module. This model does not receive in double frequency as the Broadcom BCM47755 one, but it still has an excellent accuracy of about 5.3 meters.
A list of the BCM47752 module's features is provided by Broadcom's official website:

- advanced multi-path mitigation techniques provide faster time-to-first-fix performance in challenging environments, as well as a more accurate urban navigation experience.
- multi-constellation capability collects data from four satellite constellations (GPS, GLONASS, QZSS and SBAS) simultaneously and uses the best received signals, resulting in faster signal searches and more accurate real-time navigation.
- integration of key components such as LNA enables lowest bill of materials cost.
- uses 50% less power than previous generations, allowing location-aware applications to remain active for longer periods of time.
- new applications such as Geofencing™ that provide alerts or services based on location can be completely off-loaded from the smartphone host for ultra-low power operation.
- ground-breaking indoor navigation through integration of Wi-Fi (including 5G WiFi), Bluetooth low energy, NFC and handset inertial sensor data into navigation applications.
- industry-leading urban navigation by applying handset inertial sensor readings into the position computation.
- best in class assisted GNSS (AGNSS) with both GPS and GLONASS assistance data available worldwide from Broadcom's hosted reference network.

2.3.6 Wearable Strain Sensors

The strain sensors developed for S&R project by UNICA partner are fabricated on a transparent flexible plastic substrate. In particular, a 175 mm thick polyethylene terephthalate (PET) foil has been used. The core of the strain sensors is the interdigitated Organic Field Effect Transistor (OFET) which acts as a mechanical transducer since the output current varies depending on the applied mechanical stimuli.
In particular, the OFET-based strain sensor can detect uni-axial deformations and the sensor output current increases or decreases depending on the direction of the applied bending. Downwards and upwards bending, in fact, lead to a decrease or increase of the sensor output current and thus it allows the system to discriminate what sort of deformation has been exerted on the sensor.
This aspect is possible thanks to the peculiar characteristics of the active layer of the device. Depending on the applied deformation force, the organic semiconductor is lead to an increase or a decrease of its carriers mobility.
These flexible sensors allow the monitoring of different joints and, within the project and considering the detectable strain range of the sensor, such devices can be employed for monitoring knee and elbow motion, eventually inserted into a fabric.
Shielded conductive wires can be employed in order to connect the OFET-based strain sensor to the read-out electronic circuit. Since this sensor is based on a transistor structure, three signals are needed. In particular, one is the Source electrode common ground, whereas the other two are respectively the Gate electrode (VGS=-5V), and the Drain electrode (VDS=-5V). The current flowing between the Drain electrode and the Source one represents the output of the sensor.

Table 2.1 Strain Sensors Specifications

Strain Sensors	Specifications
Device Length	15 cm or less

Device Width	5 cm or less
Device Thickness	< 2 mm – dependind of the employed fabric
Device Weight	< 10 g
Operation Voltage	5 V
Typical Output Current	10 – 50 uA
Strain Range	0 – 2 %
Sensitivity	0.125
Strain Resolution	0.2 %
Detectable current variations	< 50 nA

The sensing node will communicate with the rescuer's smartphone through a Texas Instrument BLE module (using Bluetooth libraries), which is used for data acquisition from the analog front-end and sensor, edge processing and data communication to the Android app (Java programming) developed by UNICA. The back-end of the application will collect data and send a JSON file to the data aggregation unit.

2.3.7 Wearable ECG, EMG

ECG and EMG (provided by UNICA partner) sensing is guaranteed by the adoption of electrodes based on biocompatible conductive-polymer inks, with a formulation that enables the realization on a finished stretchable garment conceived to be in tight contact with the first responders' skin on the torso (ECG) and legs (EMG). The electrodes (3 for the ECG, including 1 reference ground electrode, and 5 for the EMG, including 1 reference ground electrode) will be connected to the readout electronic modules by textile metal wires sewed onto the garment. The two sensing subsystems are physically separated both in the sensing garment (t-shirt and pants) and in the electronic systems. In fact, every different sensing is performed by a different electronic module sharing with the others, which form a body area network, the same technology for the Bluetooth low-energy (BLE) radio communication. Conversely, the analog front-end is different, according to the different nature of the signals and their intended use. The skin-electrode contact impedance is generally below 150 kOhm, with lower values in presence of sweat.
ECG signals can be acquired up to 500 Hz, EMG signals up to 2 kHz. Signal conditioning and conversion, up to 16 bits, is performed by a custom electronic circuitry integrated in the garment and implemented in a small hard package. No hardware link between the sensing garment and the uniform is present, and the communication is wirelessly provided by the BLE modules towards a smartphone running a custom app. The sensing node – app communication exploits a proprietary protocol. The Texas Instruments microcontroller featuring the BLE stack, embedded in the sensing node, provides basic signal analysis and processing features, leaving to the smartphone app the refinement of the produced results. The microcontroller firmware is implemented in C language whereas the app is written in Java for Android. The first responders' smartphone acts as a collector of signals from the body area network and a display of alert and hazards. Geotagged data are sent to the data lake as JSON files according to the project's specifications and to the integration plan.

2.3.8 Six Gas HazMat Monitor

In order to promptly and simultaneously detect multiple dangerous gases, explosive and/or toxic, each rescuer will be equipped with a portable gas monitor. The selected gas hazmat detector is the commercial Digitron's HLX3000 wireless portable monitor, which is highly sensitive, relatively inexpensive for the performance it can offer, easy-to-use, ergonomic, easy-to-carry, and safe. The device can be placed in the jacket or trouser pocket or tied to the belt.
The HLX3000 monitor has been customized in order to detect the following gases:
- CO: 0-1000 ppm, resolution 0.1 ppm;
- CO_2: 0-2000 ppm, resolution 0.1 ppm;
- O_2: 0-25% Vol, resolution 0.1% Vol;
- EX (all flammable gases calibrated on CH_4) 0-100% LEL (lower explosive limit);
- H_2S: 0-100 ppm, resolution 0.1 ppm.

The concentrations of each gas are shown on the multiple numeric display in real time.
The gas monitor is provided with the main international anti-explosion certifications (ATEX), safety reminder function, sound and light vibration, general alarm function, man-down alarm function and storage through password. The main technical specifications are the following:
- 3.7 V rechargeable lithium battery, battery capacity 2200 mA;
- more than 15 hours in continuous working mode;
- fast response time: << 30 s;
- working temperature: -20 °C ~ +50 °C;
- environmental pressure: 86 ～ 106Kpa;
- size: 157 * 84.5 * 59.5 mm (length * width * height);
- weight: 365 g (including battery, belt clip and filter);
- wireless communication (Bluetooth Low Energy, BLE).

The device will communicate with the rescuer's smartphone through a Texas Instrument BLE module (using Bluetooth libraries), which will send data to a custom Android app (Java programming) developed by UNICA. The back-end of the application will collect data and send a JSON file to the data aggregation unit.

2.3.9 Radiation sensors

The radiation sensor employed in the project is an organic thin-film resistor. The combination of materials ensures a direct transduction of the impinging radiation into an electrical signal. The sensor exploits the peculiar properties of organic electronics, such as the low fabrication costs, flexibility and lightness-of-weight of employed materials. The sensor, a 5x5 mm pixel, can be easily integrated with standard electronic readout modules. The tested sensitivity (55.3±0.4 nC/Gy) made it comparable with inorganic detectors.
The sensing node will communicate with the rescuer's smartphone through a Texas Instrument BLE module (using Bluetooth libraries), which is used for data acquisition from the analog front-end and sensor, edge processing and data communication to the Android app (Java programming) developed by UNICA. The back-end of the application will collect data and send a JSON file to the data aggregation unit.

2.3.10 Drones & Collaborative drone platform

Drones play an important role in increasing safety and productivity across different urban search and rescue operations, such as mountain avalanches rescue operations, industrial accidents, desert rescue search, and rescue search under collapsed buildings after earthquakes. The development of Drones capabilities like onboard processing, flight duration, maximum load, camera resolutions as well as Drones' enhanced features introduced in industry like dual control, automation, and obstacle avoidance, make it a more promising future technology to enhance victim localization. Further, the thriving development of new computer vision technologies facilitate and accelerate the victim localization's task.
In the S&R project the DJI Phantom 4 v2 will be used to stream and/or record video of the operational field. Drone data will be then integrated with ICT systems that are used in S&R to provide

a clear vision of the drone and the environment situation for the ground operator which allow to protect physical resources. The DJI Phantom 4 v2 proved to have the required hardware specifications that allow the required field trial in the context of search and rescue operations. It includes a camera onboard with a Gimbal of 3 axes for stabilization (pitch, roll, yaw). The Gimbal pitch be controlled from -90° to +30° with a max speed of 90°/s. The camera integrated with the Drone provides a FOV of 84° 8.8 mm/24 mm (35 mm format equivalent). It provides pictures with three aspect ratios (3:2 Aspect Ratio: 5472×3648, 4:3 Aspect Ratio: 4864×3648, 16:9 Aspect Ratio: 5472×3078) and support different photography modes like Single Shot, Burst Shooting, Auto Exposure Bracketing (AEB) and Interval. The camera also supports different Video Recording Modes including Cinema 4K (C4K: 4096×2160 24/25/30p @100Mbps) and different video formats (MP4/MOV (AVC/H.264; HEVC/H.265)). The quality of the images and video considered sufficient for the requirements of victim localization module. The quality of the images and video may also be affected by communication technologies. The DJI Phantom 4 v2 has a 6000 mAh LiPo 2S battery allowing the drone to fly for max 25 min which is considered sufficient for the testing and the field trial. The flight time may also be affected by weather conditions. In term of resistance, The DJI Phantom 4 v2 could operate in max speed with of 10 m/s and in a range of temperature between 0° to 40°C. It is not recommended to perform a flight of DJI Phantom 4 v2 on snow and heavy rain.
The DJI Phantom 4 v2 will be controlled by a Control Unit which will be connected to a laptop (equipped with a 4G toggle) where an DJI SDK Windows application will be installed. These software components will send the camera position, the battery level, the GPS position of the drone and the video data to the Data Lake for the integration. The video data could also be transmitted to a streaming server. The camera position, the battery level and the GPS position of the drone will be published using MQTT messaging protocol. The integration of these data will give a better overall view of the S&R environment to the ground robot driver to avoid dangerous area like fire zones and increase awareness. The video data will be fed to an AI module developed by IDEAS to locate victims. In a large environment and with a limited fly time, to achieve a time-consuming task like SAR operation, Drones need to use collaboration to maximize the number of victims located, minimize the time needed for locating victims, provide more awareness, and protect the life of first responders. Controlling multiple Drones simultaneously is a distracting task for human operators where operators need to adjust different parameters of the Drones, react based on the change of the environment and track each Drone. The drone collaboration platform aims to allow multiple Drones to autonomously take over each other's task when it is time to charge or replace the battery, so there is no interruption in the execution of the mission. The drone collaboration platform will allow the operator to assign a mission to a leader drone and to choose to use the collaboration mode. If the collaboration mode is chosen, the operator needs to set additional settings like the drones that could participate in the mission. The drones participating in the mission will interact with leader drone to coordinate their task and achieve the mission. The drones will use internal communication interfaces to exchange messages. These messages will not be integrated with S&R integration platform. The drone collaboration platform will be realized using a simulation tool. Simulations of collaborative mission will be performed during the project and the uses cases.

These components are provided by UHASSELT partner.

2.3.11 3D Mixed Reality Command Centre

2.3.11.1 Role in S&R Project

A 3D Mixed Reality Command Centre (3D MR CC), provided by CERTH, in development in the S&R with purpose of visualizing contextually relevant and online spatial information from different data sources to the decision makers. The rescue operations in S&R produce a large amount of heterogenous data with both spatial and chronological information that needs to be timely interpreted from the decision maker in order to achieve the most informed decision. Virtual reality has already been used for visualization of big data with geotagged information in the initial geospatial domain with benefits the faster understanding of the underlying structure of the information.
The 3D MR CC allows the user to experience the virtual world without losing connection to the real world. This type of experience allows the user to keep the awareness of the real world (what is happening around them) and at the same time use in their benefit the virtual spatially correct representation of the S&R data.

2.3.11.2 API Specifications

Data communication with external sources is implemented with the MQTT protocol. MQTT protocol is a publish-subscribe network protocol for IoT devices. MQTT is lightweight and supports lossless communication. The MQTT protocol is functioning over TCP/IP protocol. The MQTT communication consists of a server message broker and clients which subscribe and publish in different topics. For the implementation of the MQTT communication inside the application the Unity plugin M2MQTT was used.

2.3.11.3 Services

The 3D MR CC is a visualization application of what is happening on the field. The application does not provide any services apart of visualizing information from the MQTT interface and the video streams from cameras on the field.

2.3.11.4 Data format

The messages that are used in this application concern the position and state of the rescuers, victims, robots and UAV as messages that can be sent from the application as the selection of an area on the map and new destination to robots and UAV.
A message for a victim is a string that can be formulated as follow: 1623067157; v; 1; 40.56698565; 22.99799482; 40.644145; 22.936301; m; 12; l; 60; 12 with each sub-string separated by a semicolon having the following meaning:

Timestamp: 1623067157
type: (v)ictim
ID: 1
Latitude : 40.56698565
Longitude : 22.99799482
Destination latitude : 40.644145
Destination longitude : 22.936301
sex: male
age: 12
position: (l)ying down
heartrate (bpm): 60
respiration rate (rpm): 12

2.3.12 Smart Glasses

One of the newest breakthroughs on the market of wearable devices are the VR headsets and they are already starting to fill different needs into daily lives of people. Before explaining the use of a VR headset, a general explanatory paragraph about Virtual Reality (VR) is provided in the following.
Any computer-generated environment that allows the user to enter and interact with it by the means of technology can be considered virtual reality. The environment doesn't necessarily need to simulate 100% the reality but it needs to be able to create the illusion of an alternate reality where the user is able to take part. Because of this, VR can involve not only the 3D content but also sound, vibration or various other inputs and effects.
A virtual reality headset is a special glass looking device that contains two heads-up displays, one for each eye that allows users to interact with digitally created environments and experience any type of activity in a first-person view. The user's view of the surrounding environment is replaced with virtual reality content containing videos, games or another 360-degree environment that immerses the user by allowing him/her to turn and look around, just as in the physical world. The main problem with older devices is the fact that those were not offering total freedom to the user as they were tethered hardware as well as being weighty and expensive. Today's VR headsets are lighter as they consist of a pair of goggles with smart phone hardware attached to it, which make them not only more portable, but also independent of the connection with other devices, being less expensive and more resistant. If the VR headset succeeds in providing total immersion with an experience that is so real to the user that he/she forgets about the headset, it succeeds its goal as a wearable device that can be used anywhere and under any circumstance.

The VR headset that SIMAVI has chosen to develop a demo on and further implement all the functional requirements needed in completed version of S&R use cases is **Oculus Quest 2**. The client application will run on the VR headsets.
Oculus Quest 2 is one of the most feasible VR headsets on the market, as it combines portability design with good hardware and reliable sensor. It has the best price/performance ratio on the market of VR headsets. One strong point of this headset is the fact that the technical support provided by the manufacturer is good. The manufacturer provides periodic security updates, which is an important support feature. The headset contains two high resolution screens and provides the user with 6 degrees of freedom (6DoF) head and hand tracking technology that allows any user to perform actions in the virtual environment. It also contains gyroscopes and accelerometers alongside rear cameras for precise movement tracking.

2.3.13 COncORDE Platform

COncORDE [3] is a cloud-based platform, provided by KT and dedicated in Crisis management operations. COncORDE will be used to organize the Use Case scenarios within S&R system in the following ways.
In operational level:

- When the incident occurs, in the beginning of every Use Case, the Command Center receives a call and the High Commander fills the incident details to the Incident Management System of COncORDE, in order to request dispatch by the rest members of operation. From this point on, the High Commander watches closely the operation through the other EMS features of COncORDE, such as the SitREP (situational report) filled by the EMS users on the field, the SOT DSS recommendations and the notifications coming from the field. The High Commander can use COncORDE in a Desktop in the operation room.
- The dispatch requests are first addressed to PSAPS (public answering point) and field commander, who have similar roles to the High Commander in orchestrating the operation, but from the field. PSAPS are communicating with the Hospital Commander (Hospital near the incident), in order to request beds for the patients and the Field commander orchestrate the rest EMS units on the field operations. These users can use COncoRDE in their smartphones.
- The dispatch requests are also addressed to EMS units and EMS actors, that are the first responders of the S&R operations. These actors have specific roles such as runners, retrievers and rescuers and they are assigned specific tasks by the High and Field Commander. The EMS users can use COncoRDE in their smartphones.

The brief description on usage of COncORDE in an operational level, highlights the services of the EMS features, which are:

- The Login Service
- The User Service
- The Organization Service
- The Incident Service
- The Triage Service
- The Chat Service
- The Notification Service

Through the usage of Django REST framework[4][5], the aforementioned services sustain inputs, parameters and responses to a plethora of S&R components, but also to the end-users on the operation. These services are exposed via **REST** based architecture, allowing their use through HTTP request operations. Their data format is in **JSON**. More information can be found on chapter "3.2 COncORDE Platform" in "D7.3 Component Interface Specification for interoperability within S&R".

2.3.14 Sensor web services

In use cases of S&R, five "multiple sensors" will be involved, each containing more than one individual sensor. These multiple sensors are:

1. Rescue kit for children
2. Smart textile professional uniform
3. Emergency response health condition monitoring device
4. Six gas monitor

5. Rescue MIMS chemical sensor

Until M16, the S&R pilot tailored web services for the three multi sensors that are participating in the use cases were designed and implemented. The implementation of their services is based on an existing data model as defined in T2. Also, the extraction of the sensor detections from the data lake will be done through the web services interoperability framework as it is described in T2.

Table 2.1 S&R pilot tailored web services

S&R pilot tailored web services		
Rescue kit Methods	**Smart Uniform Methods**	**Victim Simulator Web Service**
getAllDetections	getAllDetections	getAllDetections
getDetectionsByLocation	getDetectionsByLocation	getDetectionsByLocation
getDetectionsByBloodOxygenLevelValueRange	getDetectionsByBloodOxygenLevelValueRange	getDetectionsByBloodOxygenLevelValueRange
getDetectionsByBodyTemperatureValueRange	getDetectionsByBodyTemperatureValueRange	getDetectionsByBodyTemperatureValueRange
getDetectionsByEcgValueRange	getDetectionsByEcgValueRange	getDetectionsByEcgValueRange
getDetectionsByHeartRateValueRange	getDetectionsByHeartRateValueRange	getDetectionsByEnvironmetHumidityRange
getDetectionsByRespirationRateValueRange	getDetectionsByRespirationRateValueRange	getDetectionsByEnvironmentTemperatureRange
	getEnvironmentDetections	getDetectionsByHeartRateValueRange
	getEnvironmentTemperatureDetectionsByValueRange	getEnvironmentRespirationRateValueRange
	getEnvironmentUmidityDetectionsByValueRange	getEnvironmentByVictimPosture
	getGasDetections	
	getGasDetectionsCo2ByValueRange	
	getInclinemeterAngleDetections	
	getRadiationDetections	
	getRadiationDetectionsByValueRange	

Table 2.2 Web Services methods description

Web Services methods description	
Method	**Description**
getAllDetections	Show all the detection details based on the specific sensor
getDetectionsByLocation	Search for detections based on location for the specific sensor
getDetectionsByXYZValueRange	Search for detections based on XYZ sensor level
getEnvironmentByVictimPosture	Search for detections based on victim posture
getEnvironmentDetections	Search for detections based on ambient detections
getEnvironmentTemperatureDetectionsByValueRange	Search for detections based on ambient temperature value
getEnvironmentUmidityDetectionsByValueRange	Search for detections based on ambient humidity value
getGasDetections	Search for detections based on chemical detections
getGasDetectionsCo2ByValueRange	Search for detections based on ambient Co2 value
getInclinemeterAngleDetections	Search for detections based on body stance detections
getRadiationDetections	Search for detections based on x-ray radiation detections
getRadiationDetectionsByValueRange	Search for detections based on x-ray radiation value

The web services define their functions via the corresponding wsdl files for each web service, by using the SOAP messaging protocol.

2.3.15 SOT Decision Support System

The SOT DSS, provided by KT, enhances the decision making on the operation by providing the below recommendation services:

- **Service 1 - Allocation of EMS Units from EMS stations to Incidents:** This service concerns the recommendation of the most efficient allocation of resources to incidents by using optimization techniques. It depends on the demand of EMS units from the incident, the supply of EMS units (fleet size) from EMS stations and the location of the incident and EMS stations.
- **Service 2 - Allocation of Patients to EMS Units and to Hospitals:** This service concerns the recommendation of the most efficient allocation of patients to hospitals using optimization techniques. For this Service an amount of information is used like the evacuation order the Physiological score of the patient, the location, id of the patient and EMS unit etc. All this information is combined in order to produce the most efficient allocation.
- **Service 3 - Allocation of Actors to Tasks:** This service concerns the recommendation of allocation of tasks to available actors on the field, given demand pre-defined by the field commander. Here again optimization techniques are used. This Service takes the existing tasks and the available actors and assign the actors to tasks based on the skills, the state of the actors and the needs of the task.
- **Service 4 - Casualty Estimation:** This service requests from onePAGER product, an estimation of casualties for earthquake incidents. This information is provided through an API that is constantly updated.

All the above services use as parameters the COncORDE's platform EMS features, in order to provide their final responses. All the SOT DSS services are in JSON format and are provided to the rest of the platform through a RESTful API. The SOT DSS services and their endpoints can be found to "Chapter 3.6 SOT DSS" in "D7.3 Component Interface Specifications for interoperability within S&R".

2.3.16 PHYSIO Decision Support System

The PHYSIO DSS is one of components of the overall S&R DSS provided by CNR and is responsible for the prediction of the evolution of the physiological status of the victims of the incident. It is a model-based component constituted of classes, algorithms and functions and implements a stochastics approach: the victim's condition is not described deterministically but the results are given in terms of probability distributions and, at each time during the event, the results show where the victim is likely to be in the Physiological State Variables' (PSVs) space.

The PHYSIO component, with its classes, functions and algorithms allows the simulation of a set of different crisis scenarios and provides information to help in handling a real crisis situation from the victim management perspective. For each simulated crisis scenario, a certain number of victims can be generated, each one presenting with a set of anatomical lesions, typical of the hypothesized scenario, and with different levels of severity. First responders have to triage victims and treat them with the most appropriate care available in the field until they are transported to hospital or to some other health facilities to be administered the definitive treatment.

It follows a brief description of the functionalities of the PHYSIO DSS component. An event is generated, with victims which present lesions and defects on the considered physiological dimensions. Each scenario is a simulation of a possible real-life event and associated possible victims, therefore victims are either virtually created or really identified, and once a victim is known to the system all the other PHYSIO DSS functionalities can be used in the same manner.

The anatomical lesions determine the occurrence of physiological impairments along some or all of the physiological dimensions and the system computes the probability distributions of the values and of the rate of worsening of each PSV. The evolution over time of each PSV distribution is then determined on the delivered treatment (oxygen or blood administration, etc…) or on additional information from the field (health measurements such as blood pressure or oxygen saturation, etc..). On the basis of the effects that each treatment produces on the health status of the victim the DSS will provide support to the decision maker suggesting what treatments and in what quantities must be employed to produce an improvement, or a complete restoration, of the compromised physiological dimensions. Some functions also allow the user (personnel operating in the field) to exploit the PHYSIO component for training activities, by comparing their ability to triage patients with scores and codes generated automatically by the system from signs and symptoms that can be detected from the field (as for example the computation of the Glascow Coma Scale score).

All the functionalities offered by the PHYSIO DSS Component are based on a client-server architecture delivered as Web Services. All the services will be exposed at the url: https://biomatlab.iasi.cnr.it/SearchAndRescue/. The chosen architecture will support interoperable machine-to-machine (M2M) interaction over a network and no further interoperability constraints are required or imposed since any language can be used to program a client, provided that it interrogates the server following the public interface of the webservice, respecting the SOAP specifications, with the requests and the response being exchanged in XML language.

2.3.17 Volunteer application

The Volunteer Application has been developed with the purpose of knowing the human resources availability in case of emergency. This app is under development by CERTH for the Front-End and by KT for the back-end. This app is going to inform the High Commander for the total amount of available volunteers, their skills and equipment, in order to empower the crisis response in the disaster scene.
The back-end is implemented using Django REST framework, in order to make the following API endpoints:

- User's Authentication
- User's Information
- User's equipment
- The current emergency incident's basic information (type of emergency, location, etc.)

All the requests (to COncORDE) and responses by the Volunteer API are created with JSON objects. All these data are stored in a remote (only for this app) database (PostgreSQL) based on the models of the app.
The volunteer application endpoints can be found in "Chapter 3.8 Volunteer Application" of "D7.3 Component Interface specifications for interoperability within S&R".

2.3.17.1 API Specifications

Volunteer App's API was implemented using Python's Django REST Framework, which is a powerful and highly scalable framework used to build Web APIs. After that, the application was deployed in KT's server using Docker containers and Nginx. Docker is a platform that creates containers holding operating systems. Hence, it serves as a software that creates virtual containers/machines. Nginx is a web server that can be used as a reverse proxy, a load balancer, a mail proxy and HTTP cache. This technology was used to deploy the API in a server. A container holding the API itself and one more holding the Nginx web service were implemented to deploy the Volunteer App and make it available for production.
The available API endpoints of Volunteer App refer to:

- The user's authentication
- The user's information
- The user's CRUD (Crate, Read, Update, Delete) functions
- The user's equipment CRUD functions
- The current emergency incident's basic information (type of emergency, location, timestamp).

2.3.17.2 Services

The services that Volunteer Application provides lie in common ground with its role in the project. Its most important scope is to highlight the availability of volunteers in cases of emergency. A volunteer can use this app to mark themselves available for support in S&R's emergency incidents. By registering the country, they are situated in, the voluntary organizations they belong to (if any), their role in the organization, their skills and their available equipment, the users can offer their support and aid whenever they are needed.

2.3.17.3 Data format

All requests and responses sent by the Volunteer App's API are created with JSON objects. At the same time any data passing through the application is stored in a PostgreSQL [6] remote database based on the models of the app. These models are the following (portrayed as the database's tables):

- User model

Table 2.1 User Model Data Format

COLUMN	DATA TYPE
ID	UUID (Universally Unique Identifier)
Username	String
E-mail	String (Email format)
First Name	String
Last Name	String
Country	String (out of the country choices)*
Phone Number	String (phone number format)
Organization	String (out of the organization choices)*
Role	String (out of the role choices)*
Skills	String (out of the skills' choices)*

* Country choices: The choices for the countries are the country's ISO 3166 code that is used to indicate the country the volunteer is situated in. For instance, 'AU' for Austria, 'FR' for France, 'GE' for Germany, 'GR' for Greece, 'IT' for Italy, 'ES' for Spain and so on.

* Organization choices: The organizations registered in the Volunteer App are the following: ASSIST EMAS, O.AN.E.AN, EPAYPS, 'HELLENIC AMERICAN SEARCH AND RESCUE TEAM, IDAFK, Independent, Rescue Team Delta, SAR 312, SEP P P, THIVA AIR, Volunteer Team Forest Protection and Rescue Evoia, Volunteer Team PROTECTA of ILION.

* Skills choices: Driver, Emergency Communications, Emergency Medical Help, Firefighting, First Aid, Forest Firefighting, Forest Patrolling, Pilot, Psychological First Aid, Rope Rescue, SAR.

- Equipment model

Table 2.2 Equipment Model Data Format

COLUMN	DATA TYPE
ID	UUID (Universally Unique Identifier)
Owner	String (foreign key to User.id)
Quantity	Integer (Default: 1)
Content	String

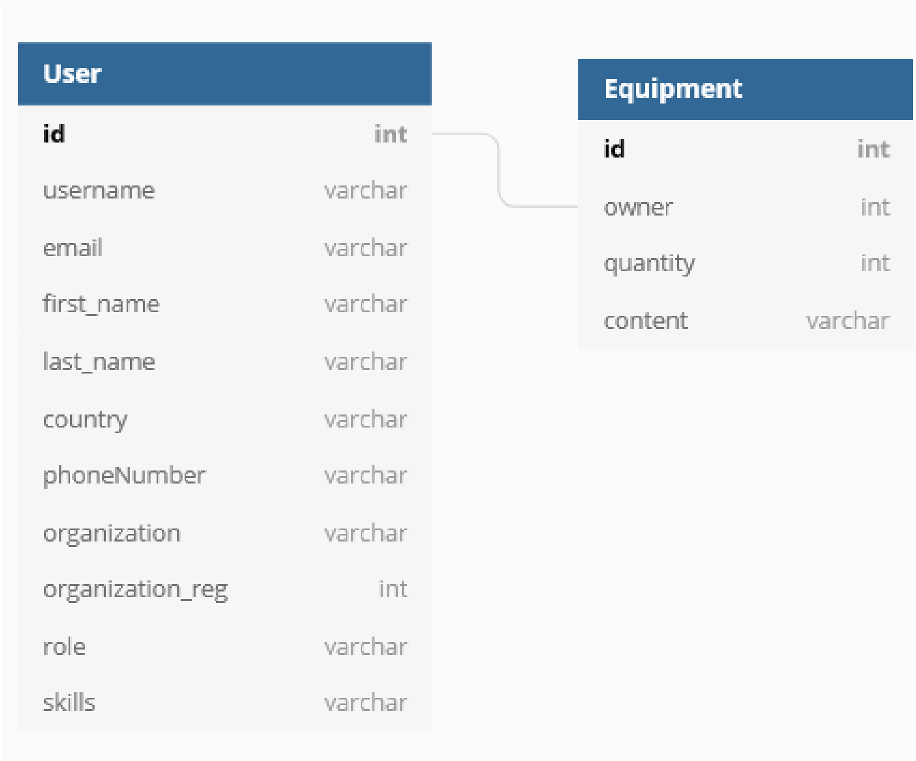

Figure 2.1 Volunteer App's Database Schema

2.3.18 E-learning based platform

2.3.18.1 Role in S&R Project

The role of the e-learning platform is to enhance participants' understanding, knowledge and skills in terms of safety and security management at operational and strategic levels. The users will be better prepared for field operations also act as security advisors within a response team providing them with the knowledge and skills to act appropriately according to the safety regulations.

2.3.18.2 API Specifications

The e-learning platform will be implemented using Moodle. Moodle is an open-source learning management system (LMS) written in PHP and distributed under the GNU General Public License. Developed on pedagogical principles, Moodle is used for blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors. With customizable management features, it is used to create private websites with online courses for educators and trainers to achieve learning goals. Moodle allows for extending and tailoring learning environments using community-sourced plugins.

2.3.18.3 Services

The e-learning platform should provide content on specific areas of S&R that need further assistance. A provided course may contain content, pictures, videos, sound, and tests-quizzes.

2.3.18.4 Data format

Sharable Content Object Reference Model (SCORM) is a collection of E-learning standards and specifications that define communications between client-side content and a server-side learning management system, as well as how externally authored content should be packaged in order to integrate with the LMS effectively.

2.3.19 Object detection algorithms for in disaster-scene SA

The object detection algorithms, applied in Search&Rescue context, are a set of algorithms developed for the collision detection. They are provided by THALIT partners. This is done by using and fusing information coming from the sensors equipped on the robot. These algorithms have been integrated in an Object Detection System (ODS) established from soft- and hardware with the following objectives:

- Detection of obstacles using data coming from different sensors;
- Tracking of obstacles by fusing information obtained in the previous step;
- Notification of potential collisions with detected objects to the robot pilot.

Thus, in the S&R context, the system will implement the following functionalities:

- Acquire sensors (LiDAR and Camera) raw data;
- Filter and process raw data at sensors outputs;
- Detect obstacles in front of the robot, associating information from different sensors and tracking them.

2.3.19.1 ***Software***

The developed software components for the ODS are shown in the block diagram represented in Figure 2.6. ODS architecture is composed of different modules with different responsibilities. These modules are combined in a pipeline in which each output of a module is the input of another one.

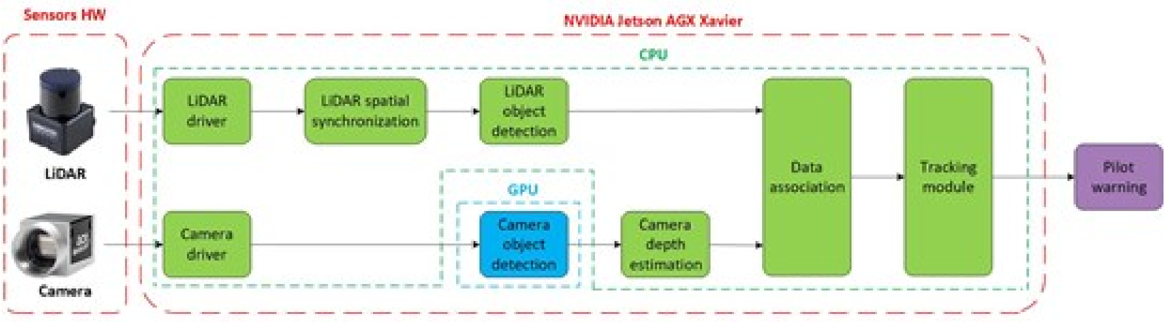

Figure 2.1 ODS block diagram

Data is processed and propagated from each block to the following one in order to accomplish the previously defined objectives. In order to do this the following operations are implemented:

- Output of smart sensors are the input of the system. This data is propagated to the following modules.
- Spatial Synchronization module can be used on sensor data to apply roto-translations to input data in order to represent the information in the same reference systems. For this application, this module is used on LiDAR point cloud.
- Objects are detected by using algorithms on LiDAR and camera data. Convolutional Neural Network, in particular YOLOv4 model, will run on video frames, enclosing objects detected by the model in Bounding Boxes (BB). Clustering algorithm will aggregate LiDAR 3D point clouds;
- Depth Estimation module can be used represents BB information (like central point coordinates, width, and length) on the robot Cartesian coordinate reference system. For this application, this module is used on camera data;
- Detected objects are fused in order to increase detection probability. A Sensor Fusion Algorithm is implemented, which is composed of two steps: data association and tracking. Data association associates the object detected from different sensors by using Global Nearest Neighbor algorithm. Then, associated objects are tracked using Linear Kalman Filters and Unscented Kalman Filters.

The Obstacle Detection System is based on the Robot Operating System (ROS) framework, which allows easy inter-process communication. Thanks to ROS, each functional block can be modelled as a separate application which can communicate with the other modules using ROS messages.
Docker containers are used to implement separated and containerized microservices: each module of the pipeline runs on a Docker container and can communicate with the others modules via ROS messages. Containers are orchestrated thanks to Docker Compose. Thanks to Docker containers, each module runs on a container generated by a predefined Docker image that can be easily managed to include the needed packages. This allows easy isolation and prototyping for ODS modules.

2.3.19.2 ***Hardware***

ODS runs on the NVIDIA Jetson AGX Xavier board with a Linux-based OS installed. This board features:

- CPU: 8-core ARM v8.2 (x64) @2.26 GHz, 8MB L2 + 4MB L3;
- GPU: 512-core Volta GPU @1.37 GHz with Tensor Cores;
- RAM: 32GB 256-Bit LPDDR4x (137 Gbps).

The ODS prototype is composed by the Peli 1400EU Protector Case including the Xavier board and a DC-DC converter from 24 V to 12 V. The dimensions of the ODS HW prototype are presented in Figure 2.7., together with the three interfaces. Reading Figure 2.7 from left to right, the following connectors are illustrated:

- RJ45 interface, for the Ethernet connection between the Xavier board and the robot.
- Power supply interface, which is a circular 3 pins connector.
- Green LED, which is turned on when the ODS HW prototype is fed

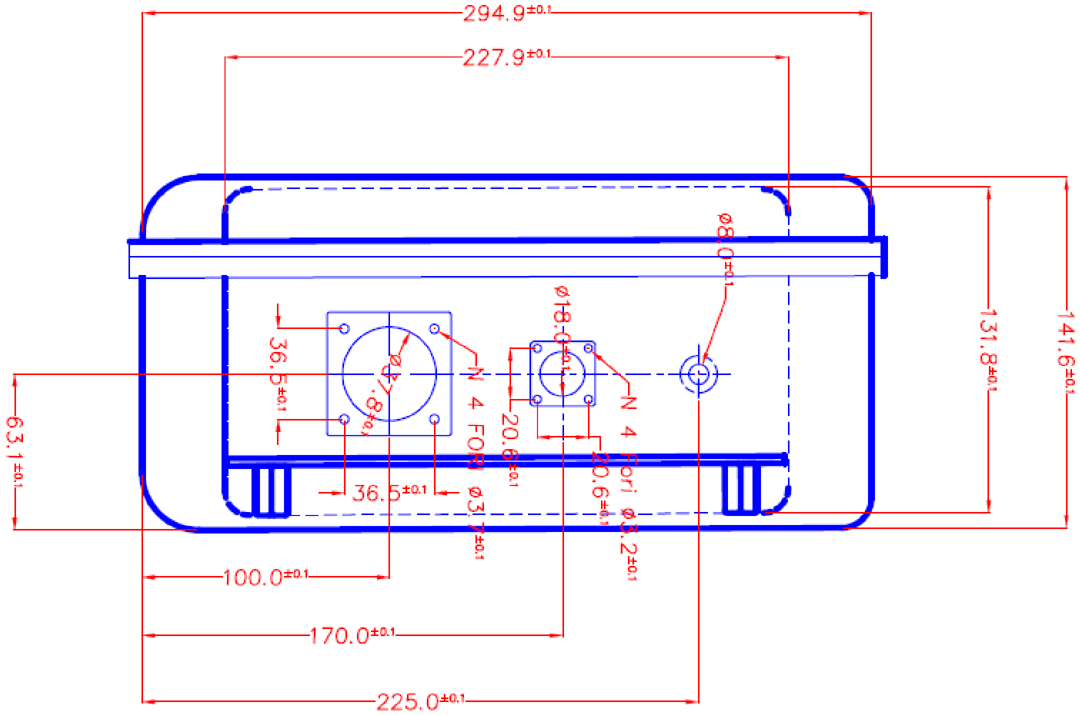

Figure 2.2 ODS Design of the interfaces of the HW prototype

2.3.19.3 ***Sensors***

The sensors used by the ODS for Search&Rescue are the ones equipped on the robot. ODS will use the following smart sensors:

- Cameras: the cameras used for this integration form a stereo setup. Each camera will provide to the system the captured frames as ROS messages (Image), which contains the raw data encoded as RGB 8 bits. Moreover, size of the frames (width, height) is also provided in this message;
- LiDAR: the LiDAR sensor used for this integration is a Velodyne VLP16. LiDAR data are 3D point clouds expressed in the sensor cartesian reference system and divided on 16 detection layers. The data is received by the ODS as ROS message (PointCloud2), which contains, for each point, the following informations:
    - Cartesian coordinates (x, y, z);
    - Intensity of the measurement;
    - Layer of detection.

2.3.20 Object detection algorithms applied on UAV imagery

2.3.20.1 Role in S&R Project

AiDEAS is in charge of creating AI-powered analytics to increase situational awareness in disaster scenes, as well as improve preparedness and response capabilities in the event of cascading failures. Object detection is a computer vision approach for locating and identifying objects in images and videos. It is used to count, assess, map, and label the exact locations of items in a scene. Object detection algorithms come in a variety of forms and can be used in a range of situations. We chose to train the human recognition class using aerial pictures from drones. Two datasets were chosen to train the object detection networks including pictures from low altitude (a collection of low-altitude datasets from various sources) and pictures from high altitude (this dataset is called Lacmus). These datasets are considered appropriate for this research since they include humans in various landscapes and during various seasons of the year as key objects.

2.3.20.2 Methodology of the object detection component

Dataset generation, training, and inference are the three steps of our proposed detection approach. The suggested methodology's major phases are summarized below.

- **Data generation / management:** A high altitude dataset, a low altitude dataset, and a concatenated dataset (combining both the low and high-altitude datasets) were generated in this phase. These datasets were utilized for training a variety of SoA deep learning algorithms.

- **Training:** This phase involves two major tasks: (1) the annotation of the collected images and (2) the training of the models. Our suggested technique was based on the most recent developments in object detection technology. Six state of the art object detection models and forty-four tests were used to compare and extract the optimum approach for reducing the inability to identify objects at various altitudes. The models, we tested, were the following:

  - Scaled YOLOv4
  - YOLOv4_Darknet Framework
  - YOLOv5_Pytorch Framework
  - Efficient Det
  - Detectron2
  - Faster R-CNN

- **Inference:** The best object detection models from above step were evaluated with images taken from various altitudes. The altitude ranges (low and high) were defined using a user-defined threshold (30m). The models were trained on three different altitude-specific datasets (low, high, and the concatenated one) and the accomplished detection accuracy results were estimated using a variety of performance metrics. Finally, the weights of the produced DL networks from the best low and high-altitude models were used to execute a fusion strategy in this phase that produces the final decisions.

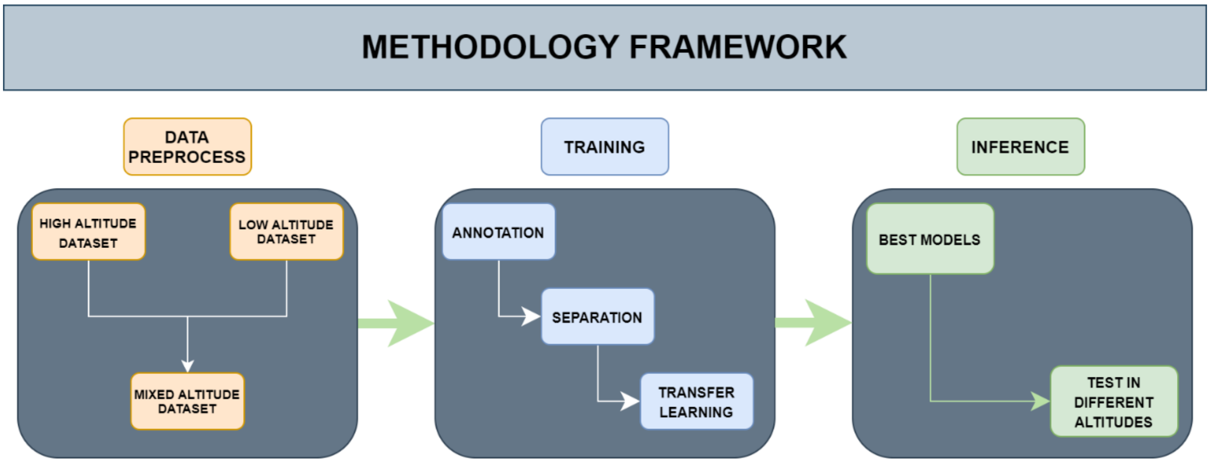

Figure 2.1 Methodology Framework diagram

2.3.20.3 Outputs of the component

YOLOv5 and Scaled-YOLOv4 gave the best results for the low and high-altitude datasets respectively whereas the Scaled YOLOv4 model yielded the best results in the concatenated dataset. The complete results will be provided on D3.5 Data-driven analytics applied on UAV imagery using deep learning.

The architecture that combines the outputs of the best local (altitude-specific) deep learning models was the one with the best overall performance (86.2%) that generalises well in both low and high altitudes.

The outputs of the proposed DL architecture can be expressed as:

- Boundary boxes created on a new photo once the model is applied, indicating where the object is in the image. The proposed object detection methodology can be also applied on videos resulting on a new video with boundary boxes (one for each object recognized). Some examples of the boundary boxes are given in the Figure below.
- Text file (.txt or. json) indicating where in the image the recognised object exists. Specifically in this case, the object detection model produces the output in three components:
  - The bounding boxes — x1, y1, width, height if using the COCO file format
  - The class of the bounding box
  - The probability score for that prediction— how certain the model is that the class is actually the predicted class

Figure 2.2 Examples of identified objects (humans)

2.3.21 Situation Awareness and Building Semantic Model modules

The scope of the semantic model web service is to ensure the timely delivery of proper information flow for the building-related and SA-related data that will be retrieved and properly interconnected in the context of the project.

The semantic model web service consists of three subcomponents, namely the Semantic Databases, the Data Transformation Handler and the Semantic Data Model Interceptor (Figure 2.10). The focus in terms of the architecture and the data which will be handled by the semantic model web service, mainly lies on the Building Semantic Model (BIM) and how the respective data can be retrieved. In brief, the main functional parts which compose the overall architectural design of the semantic model web service are the following:

- The **Backend service** that is composed of Data Transformation Handler and the Semantic Data Model Interceptor.
- The **Semantic Models** where the correlations and the knowledge-based information of each domain are stored. In the context of the project, two separate semantic models are stored, the Situation Awareness model and the BIM.
- The **Communication with the EMS** which will establish the utilisation of both semantic models through the CONCORDE platform.

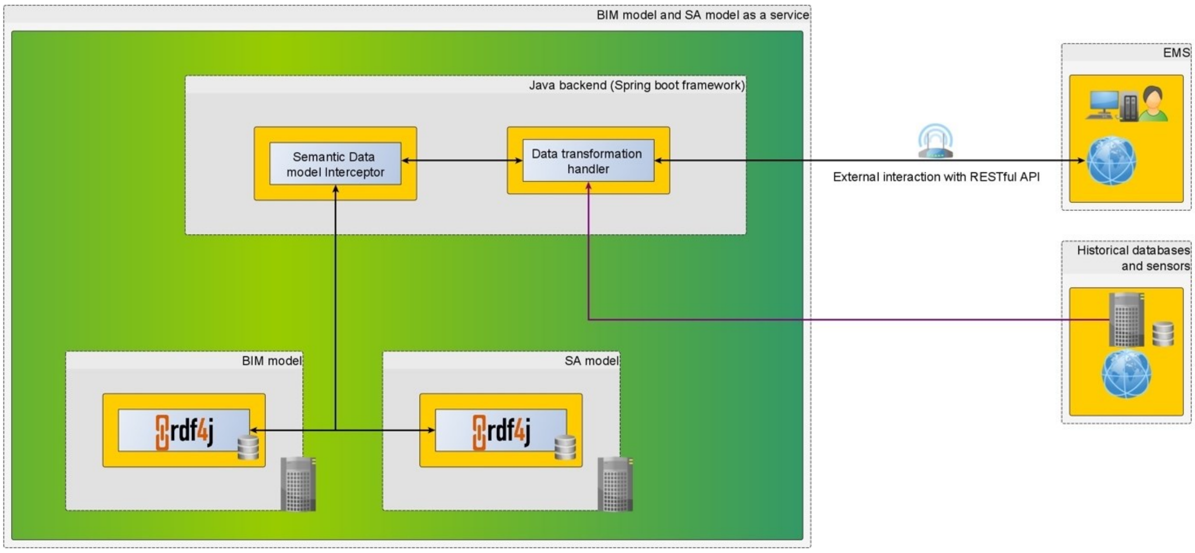

Figure 2.1 Semantic Model Web Service Architecture

These components are provided by UBITECH partner. More information can be found in D3.8 "Situation Awareness Model - specification_V2".

2.4 Application frameworks

2.4.1 Apache Kafka

Apache Kafka is a distributed data storage solution for streaming processing data. **Kafka** is mostly used to create real-time streaming data pipelines and applications that react to changing data streams. All this technology results in a faster and more efficient process of any type of real-time information and messages. The platform was developed using Scala and Java as programming languages. The way in which Kafka works is by storing the input as it comes into the system and, at the same time, provides the functionality of subscribing and publishing to streams of records to the user.

The main perk of using Apache Kafka would be that streams can be processed in real-time, avoiding lags or freezes of the application.

One other notable benefit of Kafka is the fact that it was built around the commit log. This results in a user being able to subscribe and publish to any running system or real-time applications. The two messaging models that Kafka uses are queuing and publishing-subscribe. Queuing is used for its scalability obtained through data distribution among many consumer instances; the second model, the publishing-subscribe model is a multi-subscriber solution. In order to make these two components blend together, Kafka uses a partitioned log model.

Apache Kafka is a fast, scalable, reliable and durable solution. Kafka stores the data in topics, which are created by the producers. The consumer uses the data stored on these topics. Because it is a distributed system, all the topics are split in partitions which functions as a commit log that maintains track of all records in chronological order and continuously adds new ones. Every message from a partition is associated to a unique offset in order to keep track of the incoming messaging.

Two key concepts when discussing about Kafka are *producer* and *consumer*. A producer is in charge of creating the message which contains information about where the specified message should be published on(topic) and the input of the message itself. It can also be used to define additional partitions if needed. On the other hand, a consumer is the entity which processes the input or the messages. They can have particular configurations, to work either alone or beside other consumers on specific workloads.

At the moment of writing this deliverable, regarding the S&R platform, Kafka brokers were installed on the S&R VM as Docker containers. By running the "docker ps" command on the S&R VM, all the available containers running on the instance will be displayed.

Moreover, SIMAVI will provide on-demand virtual machines for every involved partner, that will have Docker already installed. Every partner will then be able to install their own module in their perspective VM in order to test its use case.
Below you can find the list of topics available in the S&R ecosystem:

2.4.2 Apache Flink

Apache Flink is a stateful computing framework and distributed processing engine for unbounded and bounded data streams. It was built to work in all popular cluster systems and execute calculations at in-memory speed and scalability.
The core element of the framework, the distributed dataflow streaming engine is developed using Java and Scala. Data parallelism and pipelining are the main features of the engine, thus resulting in real-time event processing and parallel dataflow intake. Flink programs can process any type of data pipelines, regardless of them being bounded or unbounded data sets. The only difference is in the way in which the processing operation is done between these two. Bounded data sets have clearly defined beginnings and endings and can be processed as a batch, while the unbounded ones have no defined endings and are being processed continuously. Flink offers real-time processing, and once it is processed, data it's kept in storage file systems.

Other advantages of using Flink can be found in the following list:

- It speeds up the stream and batch processing.
- It is able to process up to millions of records per minute.
- It can power applications at scale.
- It uses in-memory performance.

2.4.3 Java Spring Framework

Spring is the most known and used open-source framework for Java, which helps in finding solution for a variety of problems when it comes to coding.
One of the main benefits of using spring is that it provides good integration with another framework. As we are developing an integration framework, composed of several components, a lot more frameworks will be used through the process. Spring allows easy integration with frameworks like Apache Kafka, Apache Flink, Apache Camel or Active MQ for example.
Besides that, Spring supports Enterprise application development through POJOs, application testing and allows modular development of the applications.
We have chosen to develop the main application using this framework because it provides a wide range of integration components.

2.4.4 Apache Camel

Apache Camel is an open-source integration framework and mediation engine meant to make the integration process easier. In other words, it transports messages from one endpoint to another. One specific message can be processed, or it can be only redirected. You can achieve this because it is connected to a variety of APIs and transports, but also EIPs and DSl, which are used to wire EIPs and transport.
Moreover, Apache Camel provides a good integration with Java, and can be easily used with the Spring framework. XML configuration through Spring Framework, XML Blueprint or even Scala DSL are a part of the configuration methods which can be used in order to set the rules. This process is followed to decide where to extract the data from, how Camel should manipulate it and where is should be sent further. In top of that, the output can be sent to a message queue, a file or can be even displayed on a screen for you to see and check it.
Camel has components which will assure the workflow of the above explained steps. These components also ensure the connection of web services, FTP services or other specific applications. Camel also provides predefined patterns which can be useful in the integration process. These are formed on basic design patterns for software.

2.4.5 Apache Knox

Apache Knox is designed as an application gateway used to interact with REST APIs and user interfaces of a Hadoop data cluster. It provides only one access point for every HTTP and REST interaction with a Hadoop cluster. It provides three categories of services:

- Proxying services;
- Authentication services;
- Client DSL/SDK services.

The main advantages of the gateway are the following:

- Improved security mechanisms since the REST and HPPT services are exposed without providing the details of the Hadoop cluster; also it uses a filter for vulnerability scanning and uses SLL protocol;
- It uses only one gateway which makes it easier to control, facilitating authorizations and auditing;
- The services encapsulation provides simplified access;
- Easy integration with custom solutions or leading market solutions.

Knox is designed so that it can be used with secured and unsecured clusters. If it is integrated with a secured cluster, it provides a security solution which is easy to integrate, protects the cluster's details deployment and it can simplify the services for client interaction.

The Apache Knox Gateway is a system that offers Apache Hadoop services an unique point of secure access. For users (who access the cluster information and performing jobs) and operators(who control access and manage the cluster), the approach makes Hadoop security simpler.The Gateway acts as a server (or cluster of servers) that gives services for one or more Hadoop clusters.

These technologies are used for S&R data lake. As an example, the sending of video drone files by Uhasselt partner to HDFS was carried out as follows:

```
curl -i -X PUT -T test_video.mkv
"http://159.89.25.78:9864/webhdfs/v1/data/processed/test_drone/test_video.mkv?op=CREATE&namenoderpcaddress=namenode:9000&createflag=&createparent=true&overwrite=false"
```

To access HDFS through the Apache KNOX gateway, the equivalent of the above command is the following:

```
curl -i -X PUT -T -u admin:admin-password -i -v -k
"https://datanode1:8442/gateway/default/webhdfs/v1/data/processed/test_drone/test_video.mkv?user.name=hdfs&doas=hdfs&op=CREATE"
```

admin:admin-password : default username and password for default topology in knox.

default : topology name , It is also default topology.

8442 : Knox gateway port number defined in gateway.port property.

datanode1 : hostname where Knox gateway is installed.

The next command is to check the content of the folder and provides us the details about each individual file:

```
curl -i -X PUT -T -u admin:admin-password -i -v -k
"https://datanode1:8442/gateway/default/webhdfs/v1/data/processed/test_drone/?user.name=hdfs&doas=hdfs&op=LISTSTATUS"
```

2.4.6 Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS) is a distributed file system. It is used because of its high fault-tolerance and because of its structure, which can be deployed on low-cost hardware. It is recommended to be used in applications with large datasets and can provide streaming access.

The main advantages of using HDFS:

- Quick detection of failures and automatic system recovery;
- Streaming access to data sets of running applications;
- It is designed to support large files, from gigabytes to terabytes of data per file;
- Portability from one platform to another.

2.5 Enterprise Service Bus (ESB)

Enterprise Service Bus, or shortly, **ESB,** is a software component which is able to realize the integration between multiple applications. The perks of integrating a component like this, would be that it can make the data model transformations and message routing, it can handle connectivity, and if needed it can also convert specific communication protocols and operate multiple requests. An ESB can expose these integrations as a service interface, which can be further used by other applications or users.

2.5.1 Apache ServiceMix

Apache ServiceMix is an open source ESB and was initially built in Java. It was created based on the Open Source Gateway Initiative framework. It can be used alone because it might be considered an ESB itself, but it can also be used together with another ESB. However, when using ServiceMix, another Apache tools should be taken into consideration, like Camel, Karaf, CXF and ActiveMQ.
The main characteristic of ServiceMix is that it was built upon a service oriented architecture (SOA). The main advantage of using this architecture is that it uses a standardised communication between the web services, it provides the flexibility in using multiple programming languages in the integration process but also ensures the independence between the client and the web service.
Moreover, the Java Business Integration API is used to root it, meaning that ServiceMix is still reconcilable with JBI services and components. More than that, the container is available under the Apache License, and it also supports the Java framework, Spring. Nevertheless, ActiveMQ can be used, which is a message broker that helps the custom configurations in order to meet the user's needs.
Apache ServiceMix is a runtime container for web services, legacy system connectivity services, and components of service-oriented architecture (SOA). It is one of the most established Apache top-level projects that implements an enterprise service bus.
The purpose of using this stack of technologies, Apache ServiceMix, Apache CXF, Apache Camel, in our context, is to facilitate the routing of the endpoints of all web services involved in the S&R project, in the same base url. The result will be an application that will act as a gateway and a REST and SOAP service handler, all coming from a common point.
The necessary configurations for a limited number of services have been performed. The new base url will be of the form https://search-and-rescue.service.eu/endpoint for all the web services. Among them will be the following: LoginService, UserService, IncidentService, OrganizationService, TriageService, NotificationService, SOT DSS, Physio DSS, Volunteer application, Push notification service, Event Log Service etc.
We will exemplify this routing through the user's management system CONcORDE platform. These services are varied in get, post, patch, delete methods. Requests, such as getting a list of Hospitals Commander by their organization Id, get all the CONCORDE users among others, can be accessed through the same entry point as the other services.
The above list illustrates some of the HTTP requests of this service. Other requests can be:

- get /users/user/update-online-status/, in order to update a user's last online time to now,
- post /users/user/user-ids/, in order to get the list of users by their ids,
- get /users/user/{id}, in order to only get a user by his/her id,

- put /users/user/{id}, in order to update a user,
- patch /users/user/{id},in order to partially update a user,
- delete /users/user/{id}, in order to delete a user instance,
- get /users/user/{id}/is-online/, in order to check if a user is currently online,
- get /users/{organizationType}/not-dispatched, in order to get a list of users that are not dispatched yet on an incident,
- get /users/{role}/, in order to get a user by his/her role

As an example of use for these REST API services, the /users/{organizationType}/not-dispatched endpoint will be accessed as follows:
https://search-and-rescue.service.eu/users/{organizationType}/not-dispatched
We will also exemplify IncidentService. Here we have a service offered by incidents management in the COncORDE system, with endpoints related to the incident management, such as the situational report (sitrep), the information form (incident), the
hazard type (hazardtype) and the notification service (notification), among others:

- post /incidents/incident/{id}/add-markers/, in order to add map markers in the CIS,
- post /incidents/incident/{id}/add-polygons/, in order to add map polygons in the CIS,
- post /incidents/incident/{id}/assign-role/, in order to assign a role to a user,
- patch /incidents/incident/{id}/decide-assignment/, in order to decide the assignment of a user in an incident,
- delete /incidents/incident/{id}/delete-markers/, in order to delete map markers instances,
- delete /incidents/incident/{id}/delete-markers/, in order to delete map polygons instances,
- post /incidents/incident/{id}/notify-hospitals/, in order to update the notification hospital list,
- post /incidents/incident/{id}/notify/notify-users/, in order to notify the users,
- get /incidents/incident/{id}/patients/, in order to get a list of patients for the incident,
- get /incidents/incident/{id}/status/, in order to change the incident status.

An example of accessing one of these services is the following:
https://search-and-rescue.service.eu/incidents/incident/{id}/assign-role/

2.5.2 Microservices

In order to develop the S&R platform, we will approach the situation by using a microservices architecture. The reason for that is that it represents a collection of small, separated and autonomous services, which can be connected, resulting in the final integration framework.
Considering that we are collecting data from a various series of sources and sensors, for every component will be a microservice to handle the information. In the project we will have four types of services: Situational Awareness Model, Disaster Medicine System, Mobile S&R Systems, and Integrated planning, which will collect and send data to the service registry. The service registry can be compared to a database for services. All the information collected from sensors will be then manipulated, for obtaining data which will eventually help in optimizing the intervention process.

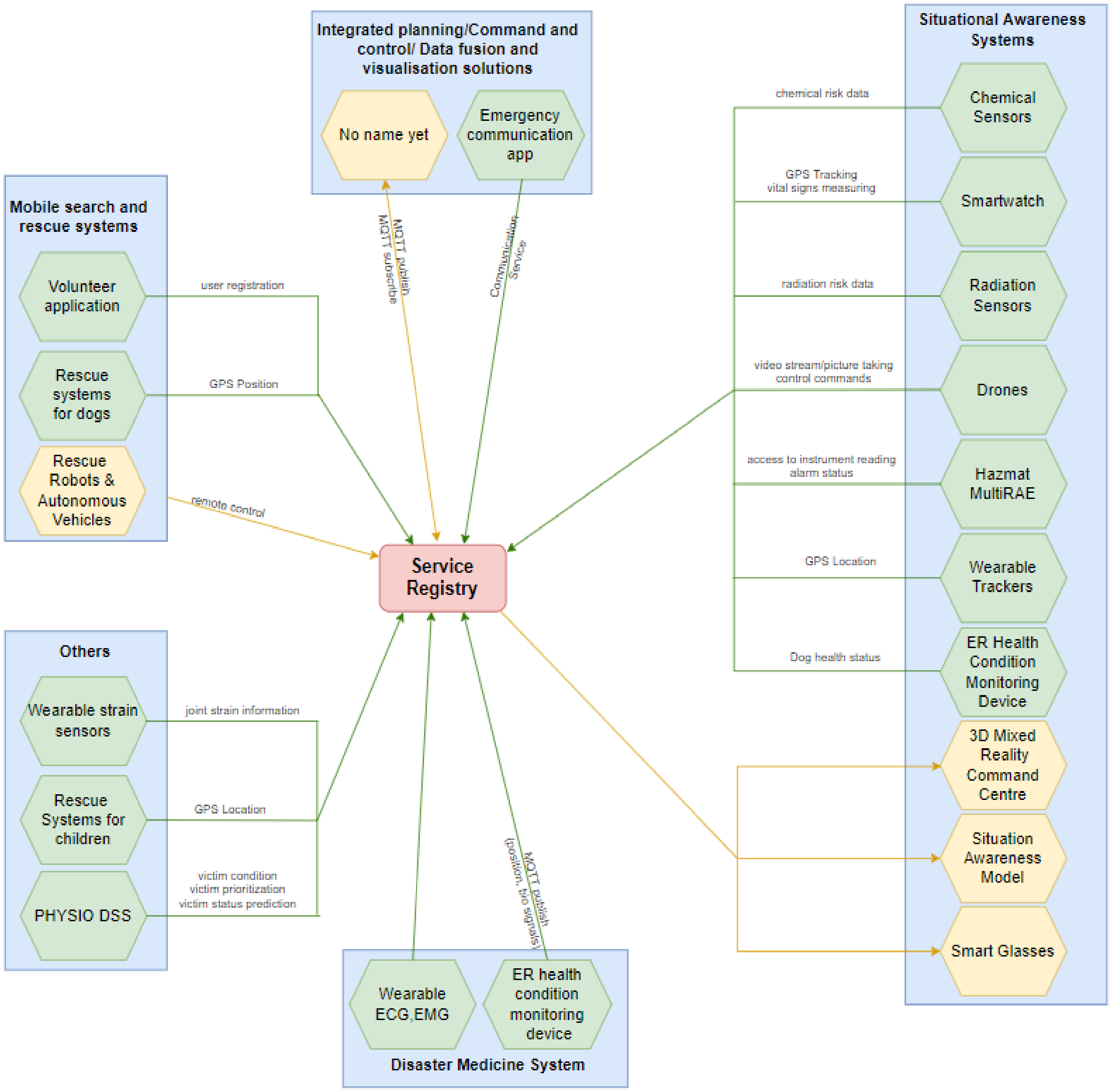

Figure 2.1 Service categories in S&R

2.6 Service integration

The service integration will be a process, through which services will have to be evaluated in several steps before arriving to their final form. During the integration process each service will go through a series of stages, before being exposed as shown in the below diagram.

The service discovery represents the way the services and microservices locate each other on a network. Once we have located a service, it will first go through the Service Interoperability Framework, which is composed of: Service handler, Service Control, Routing Engine and Access Control. The main goal will be to expose a SOAP service as a REST API, which will be used by other applications or users.
In the Service Handler a service is taken and prepared to be handled over to the protocol bridge. At this point, the handler handles the request for a service, knowing how it is defined and preparing its instantiation based on the protocol.
Once the service reaches the routing engine, a process to expose it is started. It will be transformed in a different endpoint once it is exposed.

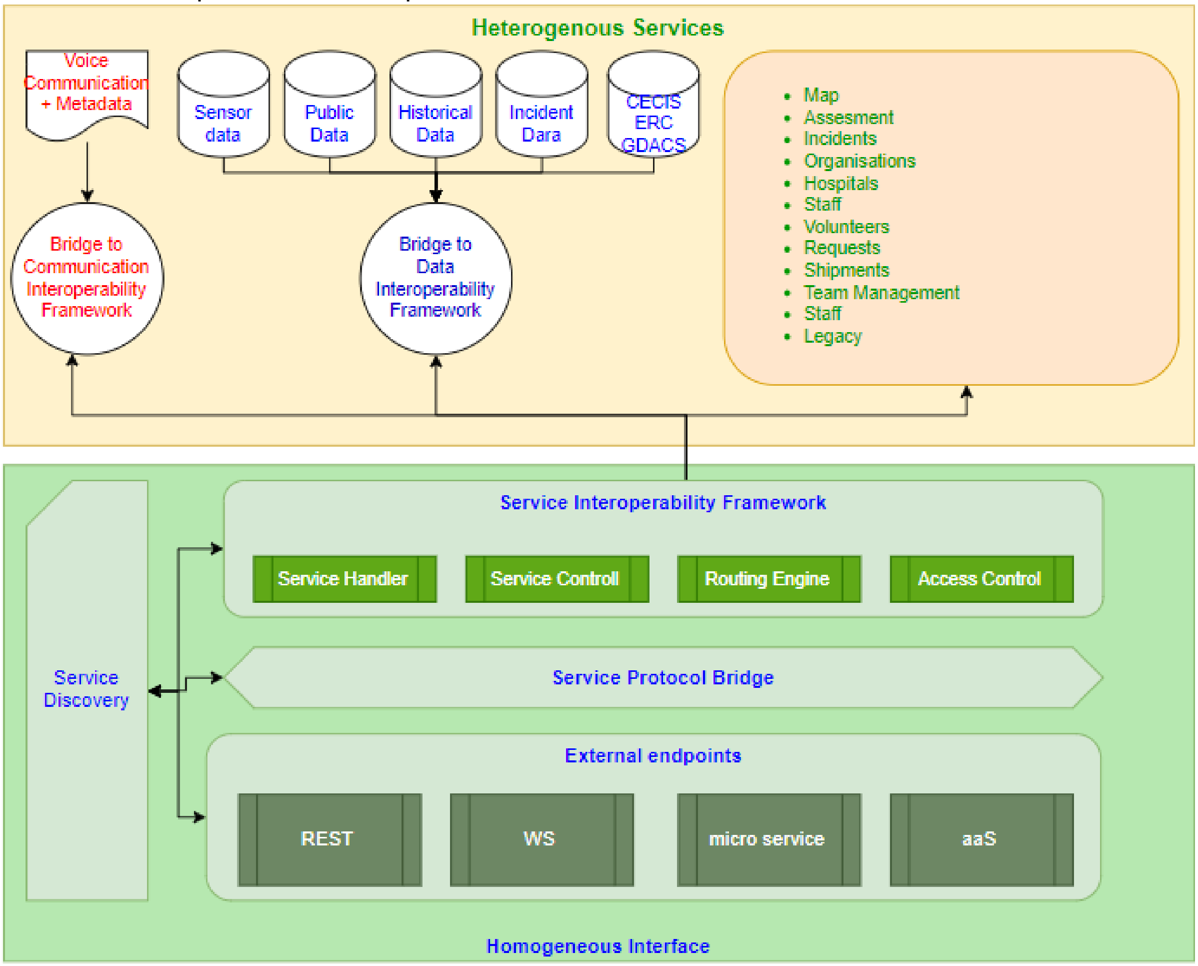

Figure 2.1 Service Framework

2.7 S&R Integration Dashboard

Search&Rescue Integration Dashboard will be used to manage Kafka cluster and to monitor data traffic from Kafka brokers and topics involved in the project.
This dashboard is very useful because Apache Kafka does not come with an inbuilt User Interface where the users can see the information related Kafka, and that's the purpose of that application, it gives us a simple and easy-to-use User Interface where one can not only see the required information but can also create and delete Kafka topics.
Through this application, the technical people have the guildelines for integration to happen and IT administration staff responsible for the infrastructure to check if the data traffic is happening properly.

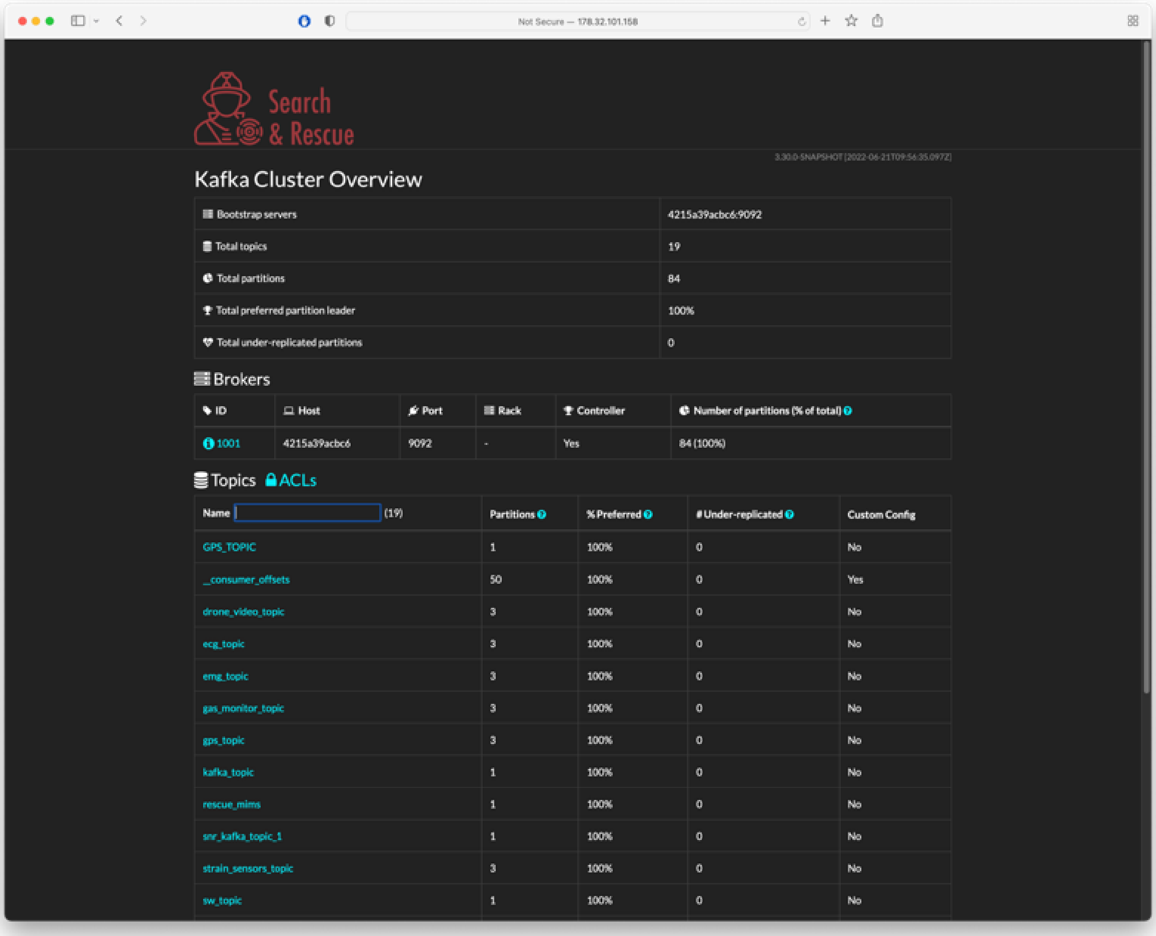

Figure 2.13 S&R Integration Dashboard

At the moment of writing this deliverable, the dashboard can be accessed at this address http://178.32.101.158:65197

3 Data Sources

3.1 Data Lake Ecosystem

The Data Lake ecosystem was built for "T4.2 Data aggregation" needs by KT, with the scope to support S&R operations with homogeneous knowledge coming from a variety of field sources. Due to the fact that all these data are structured, semi-structured, unstructured and they refer to different sources of information (e.g., heart rate from smartwatch, environmental temperature from temperature sensors), an aggregation mechanism was implemented and presented in the second section of "D4.8
Data aggregation, v2" at M12, highlighting the possible ways to:

- process,
- filter via S&R data model proposed fields (created by THALIT),
- aggregate
- and deliver valid information to the final platform

S&R's Data Lake ecosystem mainly consists of:

- Apache Hadoop
- Apache Spark

Furthermore, the Data Lake Ecosystem consists of WebHDFS, which defines a public HTTP REST API, permiting clients to access Hadoop from multiple languages without installing it in their own environment. In other words, WebHDFS concept is based on HTTP operations such as GET, PUT, POST and DELETE.
As a result, the request can be similar to the following:
Get "http://host:port/webhdfs/v1/path_with_requested_data", where:

- "host" is the server's ip where Apache Hadoop [7] is installed,
- "port" is the port which listens to (9870 for the S&R project),
- and the "path_with_requested_data" is the actual path where the required data are stored in HDFS

Considering the following figure (same as 3-11: HDFS directory structure from D4.8), the demanded request is easy to get captured. The figure below illustrates the GUI of HDFS with dummy data that were processed (transformed, filtered, aggregated, etc) for testing purposes with the usage of the Data Lake Ecosystem (HDFS for storing and Apache Spark as an aggregation mechanism).

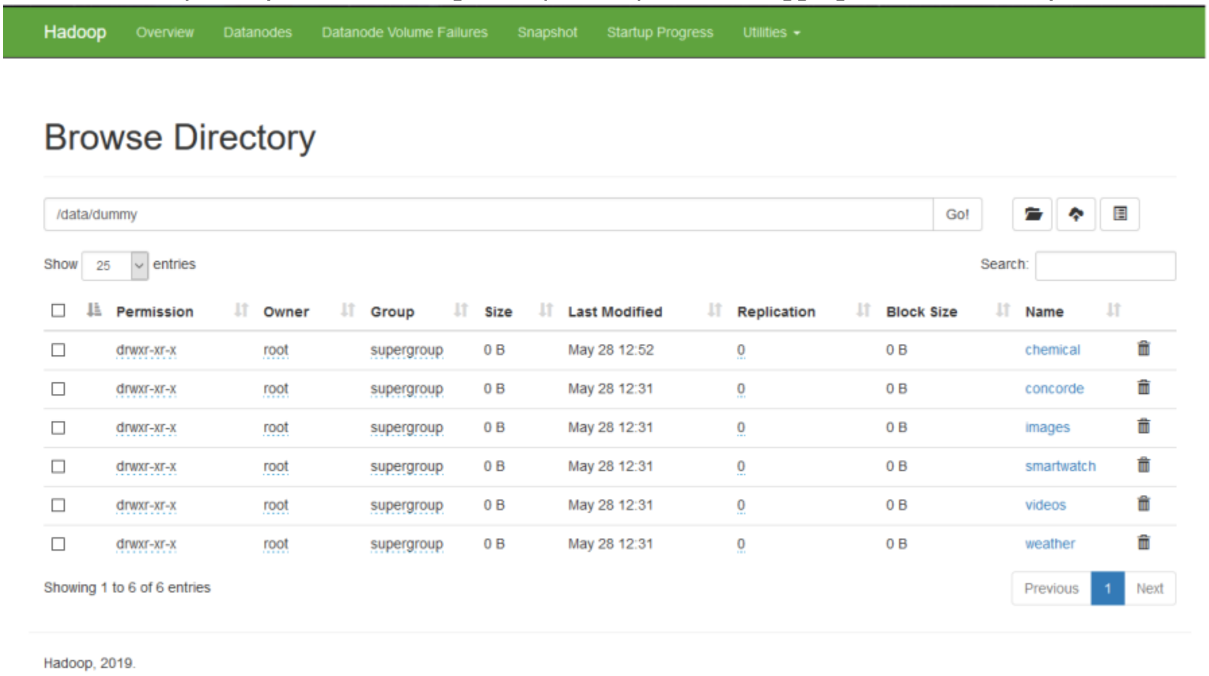

Figure 1.1 HDFS directory structure

In the above example, the possible requests are the following:
- get "http://host:9870/webhdfs/v1/chemical"
- get "http://host:9870/webhdfs/v1/concorde"
- get "http://host:9870/webhdfs/v1/images"
- get "http://host:9870/webhdfs/v1/smartwatch"
- get "http://host:9870/webhdfs/v1/videos"
- get "http://host:9870/webhdfs/v1/weather"

With this request, the user can retrieve different data that were stored and processed in HDFS. Furthermore, it is crucial to mention these are not the final directories but only examples with dummy data from D4.8 demonstration that were created for testing purposes at M12.
Data Lake's directories will be split and mapped, depending on the type of the stored data. More specifically, HDFS will be split in the following directories:
- **Field data**, where this directory will host all the incoming & processed data from the field technologies
- **Historical data**, where this directory will host all the aggregated data coming from existing databases from past incidents (provided by NTUA for "T4.1 Identify, analyse and connect to existing databases" needs)
- **Legacy data**, where this directory will host all the data retrieved from Pager or/and other legacy systems found during the project.

A directory also can have a subdirectory. For instance, smartwatch data is going to be processed via *pyspark* applications (in the Data Lake Ecosystem), temporarily stored in data frames and finally stored to specific paths in HDFS.
As a result, the division and the mapping to the HDFS could be the following:

Table 3.1 HDFS data division & mapping

Data Type	Data Usage	Data Source	Identifier
Field data/	Producer	/smartwatch/	id
	Producer	/rescue_mims/	id
	Producer	/gps_tracker/	id
	Producer	/strain_sensors/	id
	Producer	/ecg/	id
	Producer	/emg/	id
	Producer	/gas_hazmat_monitor/	id
	Producer	/radiation_detections/	id
	Producer	/health_condition_monitoring/	id
	Producer	/drones_video/	id
	Producer	/drones_other/	id
	Consumer	/3D_mixed_reality_center/	id
	Consumer	/smart_glasses/	id
Historical Data/	Producer	/historical_fire/	uc_id
		/historical_earthquake/	uc_id
		/historical_chemical_split/	uc_id
Legacy Data/	Producer	/notifications_on_earthquakes/	uc_id
Other/	Producer/Consumer	/concorde_incident_mgt/	incident_id
	Producer/Consumer	/concorde_rest_services/	incident_id
	TBD	/after_action_reports/	uc_id
	TBD	/other_component/	TBD

By creating the endpoint "/other/", HDFS will have the opportunity to communicate and retrieve inputs other than field, historical and legacy data. These inputs could be data coming from the COncORDE Platform API, with the aim to trigger other retrievals too. For example, the incident management service can function as parameter, in order to trigger the retrieval of specific historical and legacy data. So, the system could be acting as following:
- The end-user gives "earthquake" as the type of emergency to the COncORDE's incident service
- Spark jobs are triggered and make the following requests:

- get /historical_earthquake/, filtered by S&R data model
- get /notifications_on_earthquakes/, filtered by S&R data model
- post /field_data/gps_tracker/, via WebHDFS to sensor web services
- etc

The division/mapping of the data stored in the HDFS will be constantly transformed depending on the project's development (mainly the integration with field technologies and their actual data) and the end-users' needs.
A TUC (Technical Use Case) is currently under development, in order to:

- integrate via Apache Kafka (SIMAVI),
- the smartwatch data coming from STRAVA API to a specified topic (KT),
- consume them via Apache Spark Streaming (KT),
- transform/filer them with the help of S&R Data Model (THALIT),
- aggregate them to a single JSON file, divided by the smartwatches' ids (KT),
- retrieve them via WebHDFS, in order to get Detections (MAG)

After this TUC, the rest platform will retrieve this data via ESB operations and feed the services interoperability with valid knowledge for the operation. The same procedure will be followed for the integration of the rest of the hardware components.

3.2 S&R Data Model

Search and Rescue is a complex project where several heterogeneous independent technologies/sensors/assets are used to generate real time data during the intervention in various types of emergencies. All the data produced in the field shall be sent to the command centre and to the first responders in order to be analysed and evaluated to provide the best and effective intervention directives. The data are also useful to continuously monitor the health status of the first responders and the environmental conditions to identify the proper actions to be executed to solve the emergency or at least minimize the impacts it may have on the population and the environment.

Moreover, the S&R command centre shall also receive data produced by third party applications such as the European Civil Protection initiatives like CECIS (Common Emergency Communication and Information System), ERC (Emergency Response Centre) and GDCAS Service; and in the same way it shall receive also the historical data produced by other systems (for example the systems developed for projects similar to S&R): these data are useful to produce hints and directions about several things that shall be considered during an emergency intervention, such as the rescuers that shall intervene, the types of vehicles to send, the needs of the civilian affected by the crisis and much more aspects.

All these heterogeneous data have been modelled in specific data structures that altogether form the S&R data model. The modelling has been done by analysing all the possible sources of data (sensors, technologies, assets, European Civil Protection initiatives data model, historical data available on different past projects), then identifying all the relevant field data produced and grouping them according to relations and hierarchies. To design the data model, the UML approach has been adopted in order to identify the logical data structures or entities and their relations.

JAVA has been chosen as a programming language to implement the UML design and generate the java classes related to each identified UML entity. Annotations have been used within the JAVA project to automatically generate at compile time the XML-Schema files and the AVRO Schema files corresponding to the annotated java classes. To do this, specific libraries have been used such as JAXB, JAX-WS, JACKSON-AVRO.

Thanks to this approach, data model fixes and modifications (such as the addition of new entities or fields) are very quick to realize because it is enough to modify the JAVA classes (realized as POJOS) and recompile the project to automatically re-generate the XML-Schema files and the AVRO schema files.
The data model has been provided as a set of XML-Schema and AVRO schema files that can be adopted by third parties to both implement web services and/or message bus used to exchange the data through the data model entities.

The data model is based on relations between the various classes such as inheritance, associations and generalizations. For specific details of Data Model Topic refer to the deliverable D6.2 "Voice, data and services Interoperability frameworks".

3.3 Decision Support System

The data form SOT DSS services are in JSON format. In the following, there are examples of SOT DSS services in JSON Format.

3.3.1 Allocation of EMS Units to Incidents

Table 3.3.11 Examples of Service 1 in JSON format

```
{"EMSStationList":[
    {
        "id": 1,
        "location": [37.93341, 23.64579],
        "fleet_size": 6
    },
    {
        "id": 2,
        "location": [37.92944, 23.64379],
        "fleet_size": 3
    },
    {
        "id": 3,
        "location": [37.94267, 23.69757],
        "fleet_size": 4
    }],
```

Incident List

```
"IncidentList": [
    {
        "id": 1,
        "location": [37.94168, 23.65283],
        "demand": 3
    },
    {
        "id": 2,
        "location": [37.94044, 23.69132],
        "demand": 9
    }]
```

EMS Allocation

```
{
    "Allocation": {
        "1": {
            "1": 3.0,
            "2": 3.0
        },
        "2": {
            "2": 2.0
        },
        "3": {
            "2": 4.0
        }
    },
    "ReducedDemand": false
}
```

As it is shown in the above table, the SOT DSS Service 1 requests inputs from COncORDE services. More specifically, service 1 requests:

- The EMS Stations ids,
- their locations,
- their fleet size,

given by the Organization Service from COncORDE Platform,

- The Incidents id,
- The Location id,
- and the demand for resources,

given by the Incident Management of COncORDE Platform,
and finally SOT DSS Service 1 gives the most efficient allocation of EMS units to incidents.

3.3.2 Allocation of patients to EMS Units

Table 3.1 Examples of Service 2 in JSON format

<table>
<tr><td rowspan="4">Service 2</td><td>Patient List</td></tr>
<tr><td><pre>
{"PatientList":[
    {
        "id": 1,
        "physiological_score": 3,
        "evacuation_order": 5,
        "is_child": 0,
        "location": [38.010675, 23.602184]
    },
    {
        "id": 2,
        "physiological_score": 3,
        "evacuation_order": 4,
        "is_child": 1,
        "location": [38.015374, 23.598037]
    },
    {
        "id": 3,
        "physiological_score": 5,
        "evacuation_order": 6,
        "is_child": 0,
        "location": [38.005851, 23.595706]
    },
    {
        "id": 4,
        "physiological_score": 2,
        "evacuation_order": 3,
        "is_child": 0,
        "location": [38.011457, 23.601172]
        },
    {
        "id": 5,
        "physiological_score": 1,
        "evacuation_order": 1,
        "is_child": 0,
        "location": [38.013589, 23.594200]
    },
    {
        "id": 6,
        "physiological_score": 5,
        "evacuation_order": 2,
        "is_child": 1,
        "location": [38.003004, 23.602810]
    }],
</pre></td></tr>
<tr><td>EMS Unit List</td></tr>
<tr><td><pre>
"EMSUnitList": [
    {
        "id": 1,
        "location": [38.000470, 23.594144]
    },
    {
        "id": 2,
        "location": [38.010733, 23.604162]
    },
    {
        "id": 3,
        "location": [38.012321, 23.602151]
    }],
"UseUTA": false
}
</pre></td></tr>
</table>

	Patient Allocation
	{ "Allocation": { "-1": [1, 4, 5], "1": [3], "2": [6], "3": [2] } }

As it is shown in the above table, the SOT DSS Service 2 requests inputs from COncORDE services. More specifically, service 2 requests:

- The Patients ids,
- their physiological score,
- the evacuation order,
- if the patient is a child,
- their location,

given by the Triage Service from COncORDE Platform,

- The EMS unit id,
- The Location id,

given by the Organization Management of COncORDE Platform, and finally SOT DSS Service 2 gives the most efficient allocation of patients to EMS units.

3.3.3 Allocation of actors to tasks

Table 3.4 Examples of Service 3 in JSON format

Service 3	Actor List

```
{"ActorList":[
    {
        "id": 1,
        "role": "R11",
        "physiological_score": 5,
        "location": [38.010675, 23.602184]
    },
    {
        "id": 2,
        "role": "R23",
        "physiological_score": 4,
        "location": [38.015374, 23.598037]
    },
    {
        "id": 3,
        "role": "R31",
        "physiological_score": 6,
        "location": [38.005851, 23.595706]
    },
    {
        "id": 4,
        "role": "R12",
        "physiological_score": 3,
        "location": [38.011457, 23.601172]
        },
    {
        "id": 5,
        "role": "R22",
        "physiological_score": 1,
        "location": [38.013589, 23.594200]
    },
    {
        "id": 6,
        "role": "R11",
        "physiological_score": 2,
        "location": [38.003004, 23.602810]
    }],
```


Task List


```
"TaskList": [
    {
        "id": 1,
        "role": "R11",
        "demand": 3,
        "location": [38.000470, 23.594144]
    },
    {
        "id": 2,
        "role": "R23",
        "demand": 1,
        "location": [38.010733, 23.604162]
    },
    {
        "id": 3,
        "role": "R22",
        "demand": 3,
        "location": [38.012321, 23.602151]
    }],
"UseUTA": true
}
```


Allocation of Actors to Tasks

```
{
    "Allocation": {
        "-1": [
            3
        ],
        "1": [
            3
        ],
        "2": [
            2
        ],
        "3": [
            1
        ],
        "4": [
            1
        ],
        "5": [
            3
        ],
        "6": [
            1
        ]
    }
}
```


As it is shown in the above table, the SOT DSS Service 3 requests inputs from COncORDE services. More specifically, service 3 requests:

- The Actor ids,
- their roles,
- their location,

given by the Users Service from COncORDE Platform,

- The Task id,
- The role,
- The demand,
- The location,

given by the Incident Management of COncORDE Platform, and finally SOT DSS Service 3 gives the most efficient allocation of actors to tasks.

3.3.4 Estimation of casualties

Table 3.5 Examples of Service 4 in JSON format

	Estimation of casualties
Service 4	'Estimation of casualties': [['There is a 7% chance for 1-10 human casualties.'], ['There is a 26% chance for 10-100 human casualties.']]

The above table depicts SOT DSS service 4 response to an earthquake incident. In order to retrieve this response, SOT DSS Service 4 requests:

- The Incident,
- The Emergency Type,
- The Location,

given by the Incident Management Service from the COncORDE Platform. After this, Service 4 sends a request to the PAGER API [8], retrieves data for the submitted incident and finally gives a response in a PDF format. Last but not least, in order to obtain the response in the required form, KT has extracted the data from the PDF to JSON format. The final response gives to the user the probability of human casualties, classified between a scale of 1-10 and 10-100 humans.

3.4 Message Broker (KAFKA)

Kafka was chosen as the message broker. It lets you replay messages to allow for reactive programming, but more crucially, Kafka lets multiple consumers process different logic based on a single message. This makes Kafka a message broker or a streaming platform. It is a durable message broker that enables applications to process, persist, and re-process streamed data. Kafka has a straightforward routing approach that uses a routing key to send messages to a topic.
It will be used for the production and consumption of data by partners. At this moment, several topics are available for data traffic.
Below we find a table with the list of existing topics.

Table 3.6 S&R Kafka Topics

TOPIC	PRODUCER
GPS_TOPIC	GPS device
drone_video_topic	Drone Video data
ecg_topic	
emg_topic	EMG device
gas_monitor_topic	
gps_topic	
rescue_mims	
strain_sensors_topic	
sw_topic	
wearable_ecg_topic	ECG device
wearable_emg_topic	
xray_sensor_topic	
xray_sensors_topic	

4 Hosting Infrastructure

4.1 Cloud Provider

Regarding the chosen cloud services, the final decision was towards **OVHcloud** for hosting the S&R Platform, choosing from a list of other cloud service providers.
The main point in favour for choosing OVH for hosting platform is the fact that OVH is conforming to the EU legal perspective.
The other features that made OVH the prime candidate for the S&R platform are the following ones:

- OVH offers a secure cloud for businesses with a mission-oriented plan - any change must be understood by both the providers and the clients of the business model. As presented earlier, OVH provides enterprise-level security strategy in place and a continuous commitment to comply with all applicable legal and data privacy requirements, thus any business should be trusted to have their data protected. A complete enterprise-grade service should provide some level of customization and unique solutions to their customer to ensure alignment with specific technical demands and security concerns, as well as to accommodate extra requirements if needed.
- The provider has put a lot of emphasis on the secure standard. Thus, the most rigorous security standards are used by the cloud solution – the main cloud security features include the protection of data, apps, and infrastructure in the cloud computing environment. OVH cloud provides full cloud protection, from its data center to access control systems and API interfaces to automate software patching.
- It allows a variety of contributions - for each different requirement, the user has access to a huge library of solutions and services, from the simple software packages to the most specialized, personalized solution and are able to provide a reliable and long-term solution based on simplicity, accessibility, reversibility, transparency, data confidentiality, and the absence of vendor lock-in.
- Emphasis on Big-data technology - Data is all around. The way in which people use data always changes, thus more and more sources appear every day. Data are not only growing in volume, but also in complexity. It's difficult to store and process all this unstructured, heterogeneous data. For this purpose, OVH provides assist businesses in managing the problems posed by data's exponential expansion and diversification. It also provides the possibility to build infrastructures that can manage very large amounts of data, support data expansion, and address the demands of emerging technologies. It's also reversible, open, interoperable, accountable, and transparent.
- Open source —Most cloud solution uses in-house technologies that is limit the possibility of extension to other sources of technology. The open-source OpenStack platform is at the core of the OVH cloud system since 2012, and it is continually working to provide a completely reversible and interoperable Public Cloud, as well as a variety of additional services
- Top-spec components are used for the dedicated servers of OVH's infrastructure, and also have a high-bandwidth connection, along with advanced network. These configurations are designed to offer average to slightly above-average scalability requirements to companies and organizations.

4.2 Hardware requirements analysis

As shown in the subchapter 4.1, the OVH platform was selected as the preferred cloud service platform because of its scalability and security features. The cloud provider will contain the three separate server entities which will be the starting point of the S&R Platform. Each one of the systems will contain different VMs for the developing partners in order to deploy the necessary tools and this approach will facilitate the integration process between them.

The final *hardware* specifications chosen for the OVH platform are the following:

- 2x Intel Xeon Silver 4214 - 12c/ 24 t - 2.2GHz / 3.2GHz
- Memory: 192 GB DDR4 ECC 2400MHz
- Storage: 3x HDD SATA 4TB Datacenter Class Soft RAID
- Public bandwidth: 1Gbps unmetered bandwidth - 2Gbps burst
- Private bandwidth: 2Gbps unmetered and guaranteed

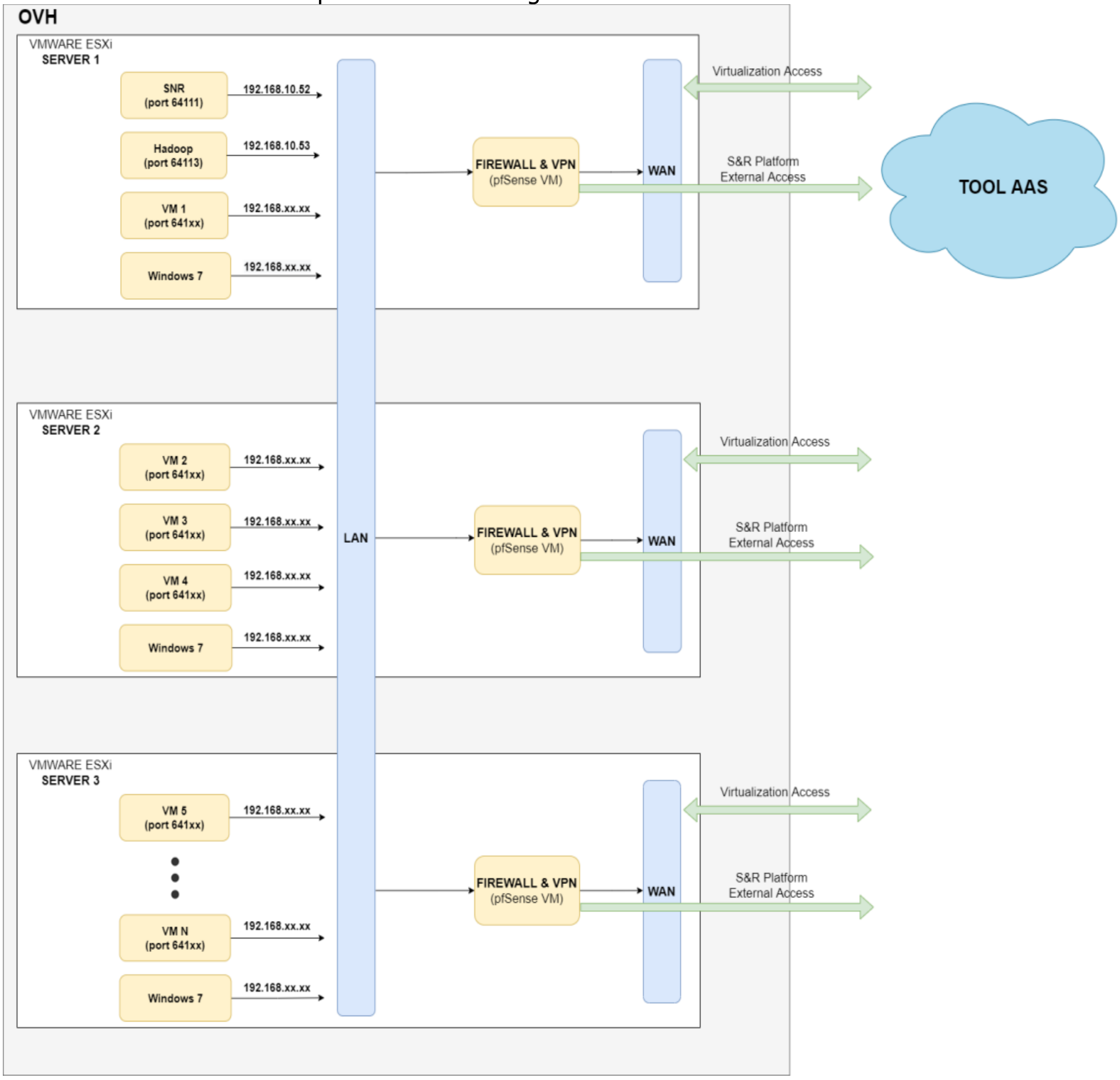

Figure 4.1 S&R graphical representation of the deployment platform

As the reader can see, the integration platform of the S&R application can be seen in Fig. Every different VM has an IP under the following form 192.168.xx.xx. These IPs are internal and are corresponding to every virtual machine that the partners will use. A strong set of firewalls will sit, for security purposes, between the outside world and the VMs. Each deployed VM can be accessed by the development team, using a combination of the IP of the individual machine and of the Port that is located on (which can be seen in the figure above).
Currently, there are three servers in total, each one having their own VMs according to their usage. The first machine contains three already installed and configured VMs (S&R, Hadoop, VM1). The S&R Platform will be located inside the first VM(S&R) accessible on port 64111. All the Windows 7 machines will not be user accessible as they are used for managing the firewalls for each of the three servers.

4.3 Docker

Docker is a containerization platform which is open source and widely used nowadays. It can be considered a software development platform and a virtualization technology used for deploying application in virtual containerized environments.
The advantages of using Docker would be that containers can be deployed on almost any machine without any compatibility issues. In the context of S&R project Docker will be used to deploy various components for testing and developing purposes.
Docker will be installed on the virtual machines SIMAVI will provide to the partners, in order to optimise the installation and utilisation of the needed components. Furthermore, the partners have to install the components they need in order to test their use cases.

5 Conclusions

This deliverable is the second version of the integrated S&R platform and presents the schema of framework integration as it was agreed in consortium and covers the integration process of all the modules and components, except the ones that are not yet finalized. The tools provided by partners that help in the information gathering process, and the services that are part of the application are presented.

The document also describes how to use the Apache Service Mix, how it helps to convert all web services involved in the S&R project and serve them from a single point of access.

As a conclusion the deliverable describes the platform as it is at the moment of writing the document. More work has to be done on the platform in order to integrate all other components that will be ready at later stages according to DoA. Also, some changes on the platform can be necessary based on use cases results.

Annex I: References

[1] "YAMAY Smart Watch for Android," [Online]. Available: https://www.amazon.com/Waterproof-YAMAY-Pedometer-Smartwatch-Compatible/dp/B07SK1BQNR
[2] "BROADCOM," [Online]. Available: https://www.broadcom.com/
[3] KT's Technical Documentation CONCORDE v0.2 https://alfresco.epu.ntua.gr/share/page/site/search-rescue/documentdetails?nodeRef=workspace://SpacesStore/bf0c1995-674c-4102-89be-e2f05b8a78d0
[4] "Django documentation," [Online]. Available: https://docs.djangoproject.com/en/3.2/
[5] "Django REST framework," [Online]. Available: https://www.django-rest-framework.org/
[6] "PostgreSQL Documentation," [Online]. Available: https://www.postgresql.org/docs/
[7] "Apache Hadoop 3.2.1," [Online]. Available: https://hadoop.apache.org/docs/r3.2.1/
[8] "USGS PAGER," [Online]. Available: https://earthquake.usgs.gov/data/pager/