eVACUATE

FP7-313161

A holistic scenario-independent situation-awareness and guidance system for sustaining the Active Evacuation Route for large crowds

The eVACUATE Emergency Operations Centre

Deliverable Identifier:	D.8.4
Delivery Date:	September, 2015
Classification:	Public
Editor(s):	Dimitris Petrantonakis, Thomas Dimakopoulos (EXUS)
Document version:	v0.99 - 2015
Contract Start Date:	April 1st, 2014
Duration:	48 months
Project coordinator:	EXODUS S.A. (Greece)
Partners:	EXO (GR), IT INNOVATION (UK), ICCS (GR), HKV (NL), TEL (GR), TEK (ES), AIA (GR), VITRO (IT), CDI (UK), INDRA (ES), KUL (BE), DXT (FR), POLITO (IT), STX-FR (FR), TUD (DE), TUC (DE), ASRS (ES), METB (ES), TIM (IT)

Project co-funded by the
European Commission under the
7th Framework Programme

SEVENTH FRAMEWORK PROGRAMME

Document Control Page

Title	D8.4 – The eVACUATE Emergency Operations Center	
Editors		
	Thomas Dimakopoulos	EXUS
	Dimitris Petrantonakis	EXUS
Contributors		
Peer Reviewers	Lazaros Karagiannidis	ICCS
	JoseMi Landeta	TEKNIKER
	Sandro Viola	VITROCISET
Format	Text - Ms Word	
Language	en-UK	
Work-Package	WP8	
Deliverable number	D.8.4	
Due Date of Delivery	September 2015	
Actual Date of Delivery	December 2015	
Dissemination Level	Public	
Rights	eVACUATE Consortium	
Audience	☒ public ☐ restricted ☐ internal	
Date	8/12/2015	
Revision		
Version	V0.99	
Edited by		
Status	☒ draft ☒ Consortium reviewed ☒ WP leader accepted ☒ Project coordinator accepted	

Revision History

Version	Date	Description and comments	Edited by
0.1	10/11/2015	ToC	EXUS
0.5	16/11/2015	1st revision from Coordinator	EXUS
0.99	8/12/2015	Comments received and added in 1st completed draft version	EXUS

List of acronyms and abbreviations

Acronym	Meaning
EOC	Emergency Operations Centre
CRUD	create, read, update and delete
COP	Common Operational Picture
GUI	Graphical User Interface
LAN	local area network
Voip	Voice over IP
PA	Public announcement
IPTV	Internet Protocol Television
FR	First Responder
API	Application Program Interface
H/W	Hardware
J2EE	J2EE is defined as Java 2 Platform, Enterprise Edition
RDBMS	Relational Database Management System
HDFS	Hadoop Distributed File System
DDS	Data Distribution Service
USB	Universal Service Bus
VGA	Video Graphics Adapter
HDMI	High-Definition Multimedia Interface
DB	Database

Table of Contents

1. **Introduction** 9
2. **EVACUATE Emergency Operations Centre (EOC)** 10
2.1 Purpose 10
2.2 EOC from User Perspective 10
2.3 EOC's position in eVACUATE Platform 11
2.4 EOC User/ System Requirements 12
2.5 EOC's Main Functionalities 13
2.5.1 Dashboard with device status and ability to recover/restart a device. 14
2.5.2 Generic overview of current crisis situation (display of reported events) 15
2.5.3 Delivery of TETRA messages to First Responders 15
2.5.4 Delivery of various alerts to First Responders and Passengers/Users mobile application. 16
2.5.5 Delivery of messages (Voice/Text) to DECT phones /ESPA/SIP/VOIP systems. 17
2.5.6 Management and monitoring of Digital Signs' status/content. 18
2.5.7 Management and monitoring of Exit Signs' status 21
2.5.8 Management and monitoring of active media files (Video / Pictures / audio) 22
2.5.9 Data analysis outcomes from deployed Sensors. 22
3. **EVACUATE EOC Implementation** 24
3.1 General information 24
3.2 EOC Architecture – Information Platform 24
3.2.1 Database of Data Access Layer 25
3.2.2 Data Access Layer 25
3.2.3 Business Logic/Controller Layer 26
3.2.4 User Interface Layer 26
3.3 EOC's Component Diagram 26
3.3.1 EOC Sofia Connection 27
3.3.2 Sensor Manager 28
3.3.3 Signalling Manager 32
3.3.4 Communication Manager 35
3.3.5 Alert Manager 36
3.3.6 Big Data 37
3.3.7 Operator/PPDR Gateway 39
3.3.8 Crowd Behaviour 41
3.3.9 Strategic evacuation 42
3.3.10 Smart Space Agents 42
3.3.11 SOFIA 43
3.3.12 DDS 44

3.4 Deployment 45
4. **Security Framework** 46
5. **Conformance with security requirements** 48
5.1 Authentication & Authorization 48
5.1.1 Credential-based authentication 49
5.1.2 Certificate-based authentication 50
5.2 Data Confidentiality & Integrity 53
5.3 Transparent Transactions 53
5.3.1 Data Confidentiality and Integrity 53
5.4 Non-Repudiation 54
5.5 Roles 54
6. **Network** 56
6.1 Network Topology 56
6.1.1 Communications - Internal Network Architecture 57
6.1.2 Local Area Network 57
6.1.3 IP Routing 58
6.1.4 Network Deployment 59
7. **EOC Beta Version Access details** 60
8. **Conclusion and Future Work** 61
ANNEX A - Hardware Inventory for EOC Implementation 62
ANNEX B References 63

List of Figures

Figure 1: Picture of already existing EOC at City of Minneapolis 10
Figure 2:Picture of already existing EOC in Maryland 10
Figure 3: EOC Overview (block diagram) 11
Figure 4: Screenshot of EOC's main dashboard 14
Figure 5: Screenshot of EOC's reported events 15
Figure 6: Screenshot of EOC's TETRA messages display 16
Figure 7: Screenshot of EOC's messages to FR's/Users through mobile Applications 16
Figure 8: Screenshot of EOC's messages transmission to DECT phones/ESPA/SIP/VOIP systems 17
Figure 9: Screenshot of EOC's monitoring capabilities over Digital Signs status/content (multimedia control) 18
Figure 10: Screenshot of EOC's monitoring capabilities over Digital Signs status/content (custom text) 19
Figure 11: Screenshot of EOC's add/remove capabilities over Digital EXIT Signs status/content 20
Figure 12: Screenshot of EOC's management and monitoring of Exit Signs' status 21
Figure 13: Screenshot of EOC's add/remove capabilities over Digital Media Files status/content 22
Figure 14: Screenshot of EOC's Data analysis from deployed sensors 23
Figure 15: 3Tier Architecture of eVACUATE EOC 25
Figure 16: EOC's detailed Component Diagram 27
Figure 17: EOC's Real Time Communication System 28
Figure 18: Connectivity of Dynamic Exit Signs 29
Figure 19: Environmental (Temp., Hum., Light) WSN Connectivity 30
Figure 20: Connectivity of RFID system 31
Figure 21: Building Management System Module (BMS) 31
Figure 22: Component Diagram of Media Manager Module 32
Figure 23: Digital Sign system architecture 33
Figure 24: connectivity of Digital Sign System 34
Figure 25: Connectivity of TETRA system 35
Figure 26: STX fire detection/phone system connectivity with communication gateway module 36
Figure 27: EOC's Alert Manager Module - component diagram 36
Figure 28: Data Analytics Process. 37
Figure 29: EOC's big data framework architecture – block diagram 38
Figure 30: Operator/PPDR gateway module (functional diagram) 39
Figure 31: Seamless failover and failback of Communication Servers 40
Figure 32: Security Application Stack 47
Figure 33: Evacuate Database Entities extending Authentication/Authorization information 49
Figure 34: Flow chart of the session creation 49
Figure 35: Flow chart service method invocation within a session 49
Figure 36: Successful user authentication with credentials 51
Figure 37: Failed user authentication with credentials 52
Figure 38: EOC Network 56
Figure 39 - High level view of the internal EOC architecture 57
Figure 40: Deployment view 59

List of Tables

Table 1: EOC Main Menu description 13
Table 2: EOC's Main dashboard devices description 14
Table 3: EOC's Main dashboard alerts description 14
Table 4: EOC's Main reported events description 15
Table 5: EOC's Tetra messenger description 16
Table 6: EOC's user card description 17
Table 7: EOC's DECT messenger description 18
Table 8: EOC's media signs messenger description 19
Table 9: EOC's digital signs messenger description 19
Table 10: EOC's list of digital signs description 20
Table 11: EOC's exit signs control description 21
Table 12: EOC's media file list description 22
Table 13: EOC's Components description 27
Table 14: Mapping of technology solutions used in Evacuate subsystem with security requirements 47
Table 15: Roles in the Evacuate security subsystem 55
Table 16: Access Rights per User Role 55
Table 17: EOC's hardware Inventory 62

1. Introduction

An Emergency Operations Center (EOC) is a central command and control facility responsible for carrying out the principles of emergency management, or disaster management functions at a strategic level during an emergency.

EOC is responsible for the strategic overview, or "big picture", of the disaster, and does not normally directly control field assets, instead making operational decisions and leaving tactical decisions to lower commands. The common functions of EOC is to collect, gather and analyze data, make decisions and disseminate those decisions to all concerned agencies and individuals. In most EOC's there is one individual in charge, and that is the Emergency Crisis Manager.

The first most critical component of an EOC is the individuals who staff it. They must be properly trained, and have the proper authority to carry out actions that are necessary to respond to the disaster. They also must be capable of thinking outside the box, and creating a lot of "what if" scenarios. The local EOC's function during an emergency is to support the Emergency Crisis Manager.

The second most critical component of an EOC is its communications system. This can be from simple word of mouth, to sophisticated encrypted communications networks, but it must provide for a redundant path to ensure that both situational awareness information and strategic orders can pass into and out of the facility without interruption.[1]

[1] https://en.wikipedia.org/wiki/Emergency_operations_center

2. EVACUATE Emergency Operations Centre (EOC)

2.1 Purpose

The EOC is the strategic level during evacuation operations, located preferable out of the crisis site at a crisis management center, which elaborates the response planning. The EOC takes into account regional effects of a crisis on a multidisciplinary basis and also deals with local effects of a crisis.

In eVACUATE, EOC provides the necessary executive support during crisis situations, thanks to the eVACUATE ICT systems, mainly computer workstations, communication infrastructure, COP tables, and all specific software and applications. EOC is located inside the end-users' premises and is operated from security personnel (see section 2.2) inside the venue.

The main difference between EOC and COP is that COP is mainly a viewing tool while EOC is a set of tools for the administration task and platform over rights. The operators of the COP can view a snapshot of the current crisis and be able to publish alerts. The EOC operator from the other hand can monitor the HW components, the sensors and be able to send alerts and messages to First Responders and other personnel (e.g. cruise ship passengers) and also to manage and manually control all devices in the eVACUATE platform.

Figure 1: Picture of already existing EOC at City of Minneapolis[2]

Figure 2: Picture of already existing EOC in Maryland[3]

2.2 EOC from User Perspective

In eVACUATE, EOC level includes operational strategic actors (operational intervention) from the first and second cores of intervention while it is located inside the end-user's premises. The EOC's team includes superior officers, communication experts and specialists needed in the crisis resolution. It has to be noticed that information and communication needs with authorities and media are managed at the EOC level.

The EOC consortium decides on the strategy to adopt in order to solve the crisis with a multidisciplinary expert point of view, taking into account potential regional damages at the largest scale. As at the EOC level, a Common Operational Picture (COP) is used as a basis of work and allows a multidisciplinary view, with the representation of the regional effects of the crisis, and the local details.

[2] source: www.avispl.com

[3] source: http://www.clearcom.com/news/maryland-emergency-management-agency-selects-clear-com

Finally, as different entities and different kinds of actors are present at the EOC, intuitive, easy-to-use and user-friendly interfaces are of most importance at this level. Of course usual actors are trained and are familiar with the EOC management and tools, but political staff or very specific specialists which are not necessarily familiar with, yet will need to understand at one glance, and communicate quickly and easily using eVACUATE support system.

2.3 EOC's position in eVACUATE Platform

EOC is the heart of the eVACUATE platform. In particular, EOC is used by the eVACUATE Emergency Officer to monitor and control the evacuate platform while intervening into its normal operations when he consider it as necessary (e.g. by over righting the media played in Digital Signs, while creating new messages to be displayed based on how the evacuation process is being evolved and/or manually control devices (Tickets machine, Fire alarm control) and legacy devices such as public announcement and DECT Phones (VOIP/SIP protocol).

EOC provides all physical connections between the modules (LAN/ WiFi), the infrastructure (Servers/Desktops), the communication software between the modules (Real Time Server, External Devices software modules and UI) and communication software with Sophia network communication layer. The overall diagram of the developed EOC is shown below:

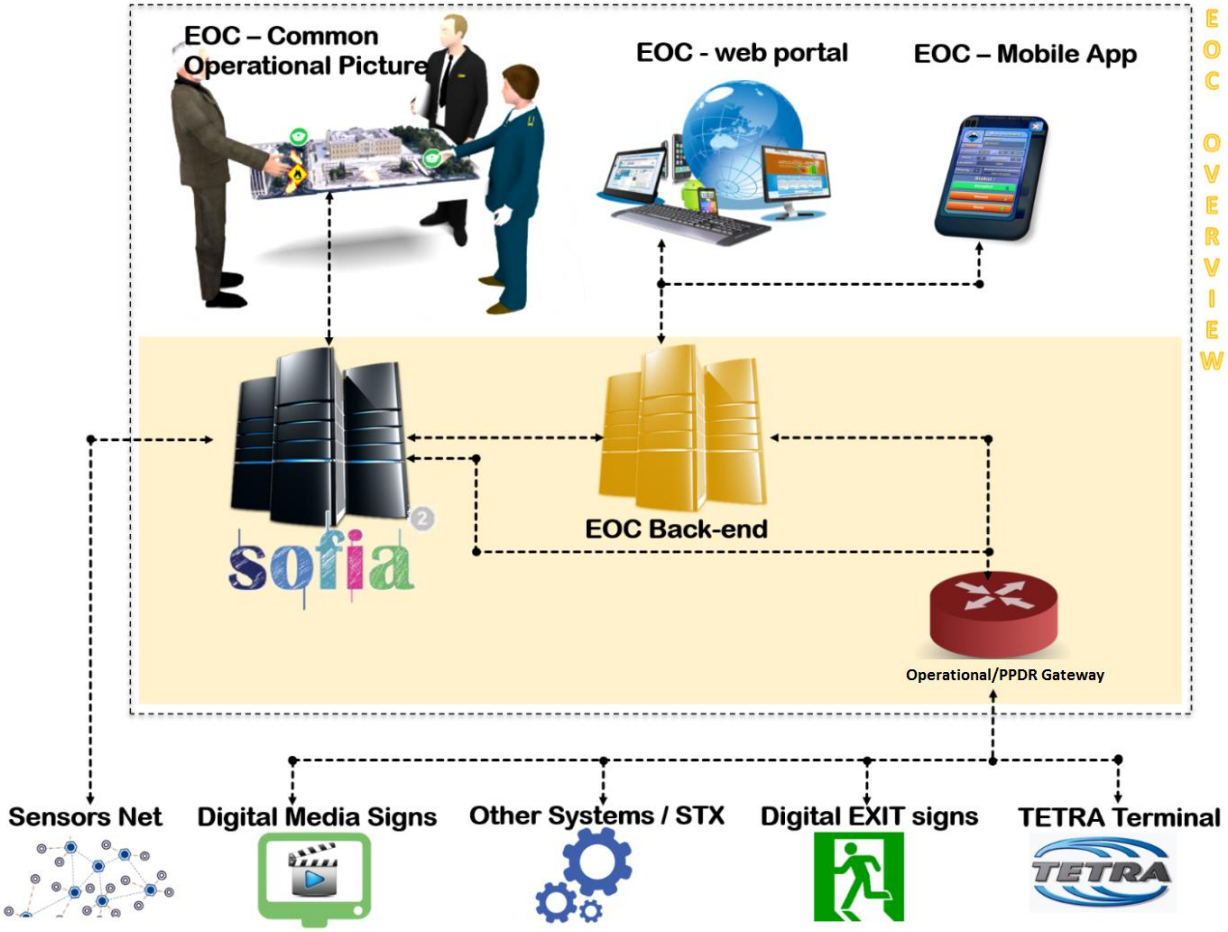

Figure 3: EOC Overview (block diagram)

2.4 EOC User/ System Requirements

Based on the defined user requirements from D.2.1 and D.2.2 reports, the eVACUATE EOC should provide the following functionalities, categorized as shown below:

- **Network infrastructure** for communication between Servers, Desktops, Sensors, Legacy and External devices and Mobiles phones. The communication can be LAN (Wired Ethernet) or wireless (Wi-Fi).
  - **Big Data** and **Cluster setup** offer best performance characteristics and using Ambari manager we have a Dashboard view of the **current** and **historical server performance**. View the historical log of the whole eVACUATE Platform.
- **Management of the Media File repository**. The Media File repository stores pictures, video and sound that are used from the Digital Signs. The operator should be able to add/edit/remove a media file from the file repository.
- **Capability to take over the control of the local Communication Gateway**. This is done with the use of Operator/PPDR Gateway and by providing a GUI to control all external and legacy devices. A list of all available over right functionalities is shown below:
  - Send an Alert/Text message to FR or User mobile application
  - Send an message to FRs TETRA device
  - Control the Exit Sign
  - Control the Digital Sign
    - STX PA System
    - STX IPTV System
    - METB Speaker System
  - Send a message to DECT Phone (in Cruise Ship)
- Provide the **status of all HW Components** of eVACUATE platform.
- Provide Interfaces for the following functionalities :
  - Send an Alert/Text message to FR or User mobile application
  - Send an message to FRs tetra device
  - Control the Exit Sign
  - Control the Digital Sign
  - Send a message to Dect Phone (In passenger suites)
  - View the historical log of the whole eVACUATE Platform
  - Perform Analytics on sensor data
  - Manage the Crisis details
- Provide the **infrastructure for eVAMAPP back office**.
- Provide the **infrastructure for COP**.
- API for **accessing the EOC Modules** using SOFIA and Restful API.
- **Ability to control the H/W modules** using a Restful API.
- Manage the **Users accounts and Users authorization**.

2.5 EOC's Main Functionalities

EOC's main functionalities can be accessed through the links existing in the main menu shown in the left side of each screenshot. The EOC Menu consists of the following categories/buttons/links.

Table 1: EOC Main Menu description

EOC MAIN MENU	Short Description
HOME	
Home	This menu item access the Dashboard where we can view the current status of all eVACUATE devices
Logout	Allow the current login user to logout
COMMUNICATION PUBLIC	
DECT Phones	Send a message to public DECT phones.
Mobile	Send a message to public mobile app.
COMMUNICATION PERSONNEL	
TETRA	Send a message to FR tetra terminal.
Mobile	Send a message to FR mobile app.
SIGNALING	
Manager	Manage all signalling devices, Digital media signs and Exit signs.
Control	Control all signalling devices, Digital media signs and Exit signs. By control mean to set a signalling device to a specific status for exit signs, or play a media file for digital signs.
CONTENT SERVER	
Media Files	Manage the Media files, direct access to the media server to add or remove a media file.
CRISIS	
Manager	Manage the crisis attributes. Future Work
Logs	View the eVACUATE log files

2.5.1 Dashboard with device status and ability to recover/restart a device.

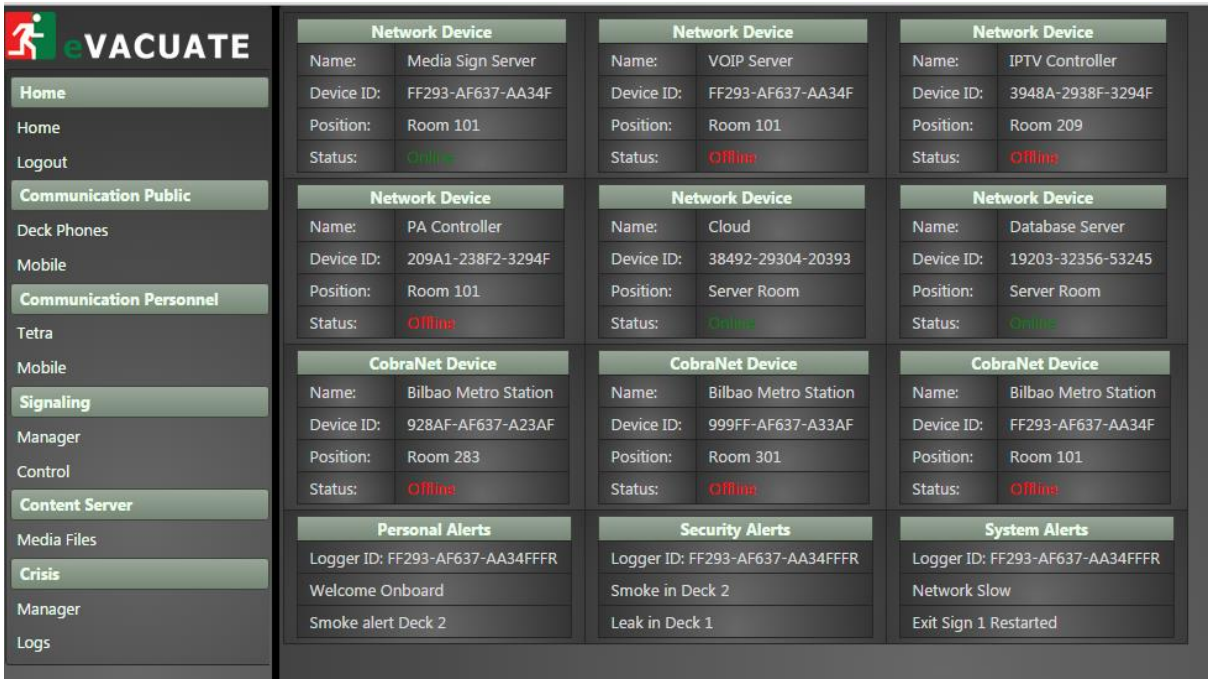

Figure 4: Screenshot of EOC's main dashboard

On this screen all the HW Devices and installed servers of the eVACUATE platform are presented to the EOC Operator while their current status is also displayed. For each component a table with the most important information is presented which can be exploited by the EOC Operator to locate the specific HW component and get an overall picture of the status of each connected subsystem/module and of the overall eVACUATE platform itself.

The main parameters displayed for each device are the following and they are split into two main categories, Devices and Alerts.

DEVICES	
Name	Name of installed device
Device ID	Unique ID reference number of installed device
Position	Location where the device has been installed
Status	Real-time monitoring of its current status (offline or online)

Table 2: EOC's Main dashboard devices description

ALERTS	
Logger ID	Unique ID reference number of the system logger. All Logs are files stored on EOC Server, while the Logger ID is used to identify the system logger.
Message(s)	List of alerts generated from the specific Logger

Table 3: EOC's Main dashboard alerts description

2.5.2 Generic overview of current crisis situation (display of reported events)

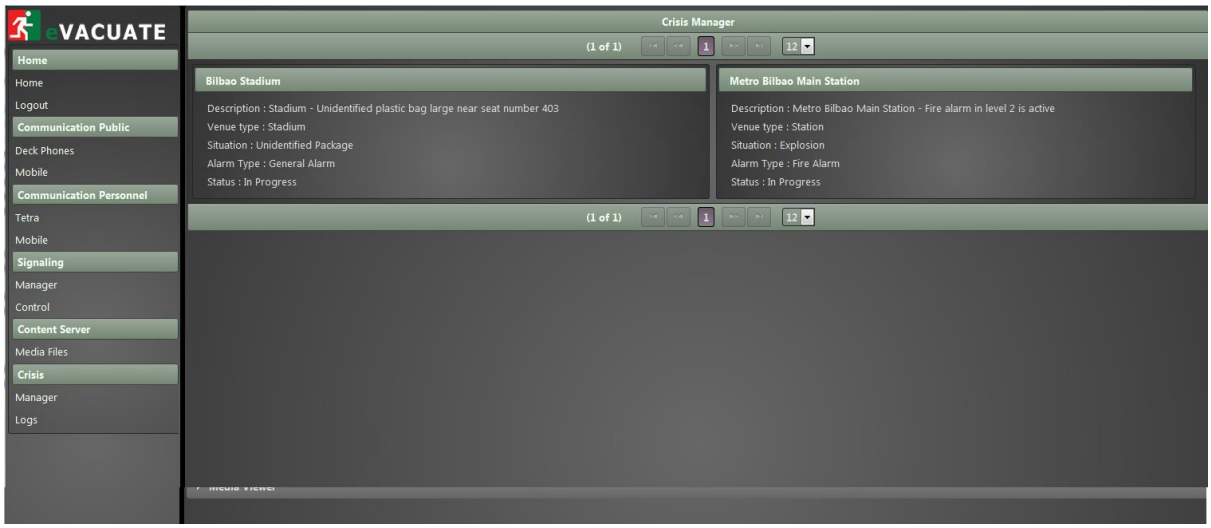

Figure 5: Screenshot of EOC's reported events

The EOC operator can view all current events associated with a crisis situation which are currently active or recently terminated. Each event has a short description of the triggered alert, the venue where the crisis is taking place, the situation which caused the crisis/triggered alert, the type of alert that was activated and the current status of the evacuation process.

CRISIS LOCATION - NAME	
Description	Description of the triggered alert
Venue type	Description of the location where the crisis event is taking place
Situation	Description of the cause that triggered the alert or initiated the evacuation process
Alarm Type	Description of the type of Alert that initiated
Status	Status of current situation. In progress, terminated, other…

Table 4: EOC's Main reported events description

2.5.3 Delivery of TETRA messages to First Responders

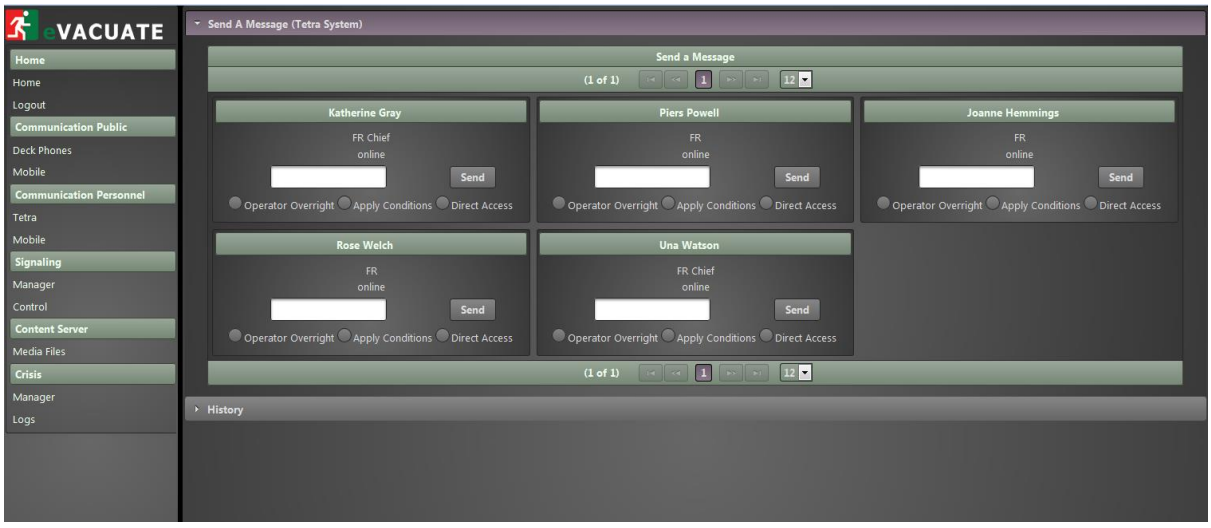

Figure 6: Screenshot of EOC's TETRA messages display

EOC's Operator has the option to deliver a short message to the First Responder's (FR) TETRA device in order to communicate directly with the specific FR or group of FRs.

NAME OF FIRST RESPONDER(s)/ SECURITY PERSONNEL	
Ranking	Displays the role of the specific person and its ranking. (E.g. FR Chief, FR, etc.)
Status	Displays its current Status, (e.g. Online or Offline)
Message	EOC Operator's message to be transmitted to the specific person
Transmission Options	• **Direct Access**: Direct communication with FR using Tetra devices. Messages are transmitted using direct communication with the H/W. This is the faster way to send messages as it bypasses all communication software and rules. • **Apply Conditions**: Controlled by SOFIA rules. Messages will be delivered based on SOFIA rules. • **Operator Over right**: Message is sent through the Operator Gateway bypassing the normal operation of eVACUATE Platform. Messages will be placed in a queue and will be delivered based on FIFO queue system.

Table 5: EOC's Tetra messenger description

2.5.4 Delivery of various alerts to First Responders and Passengers/Users mobile application.

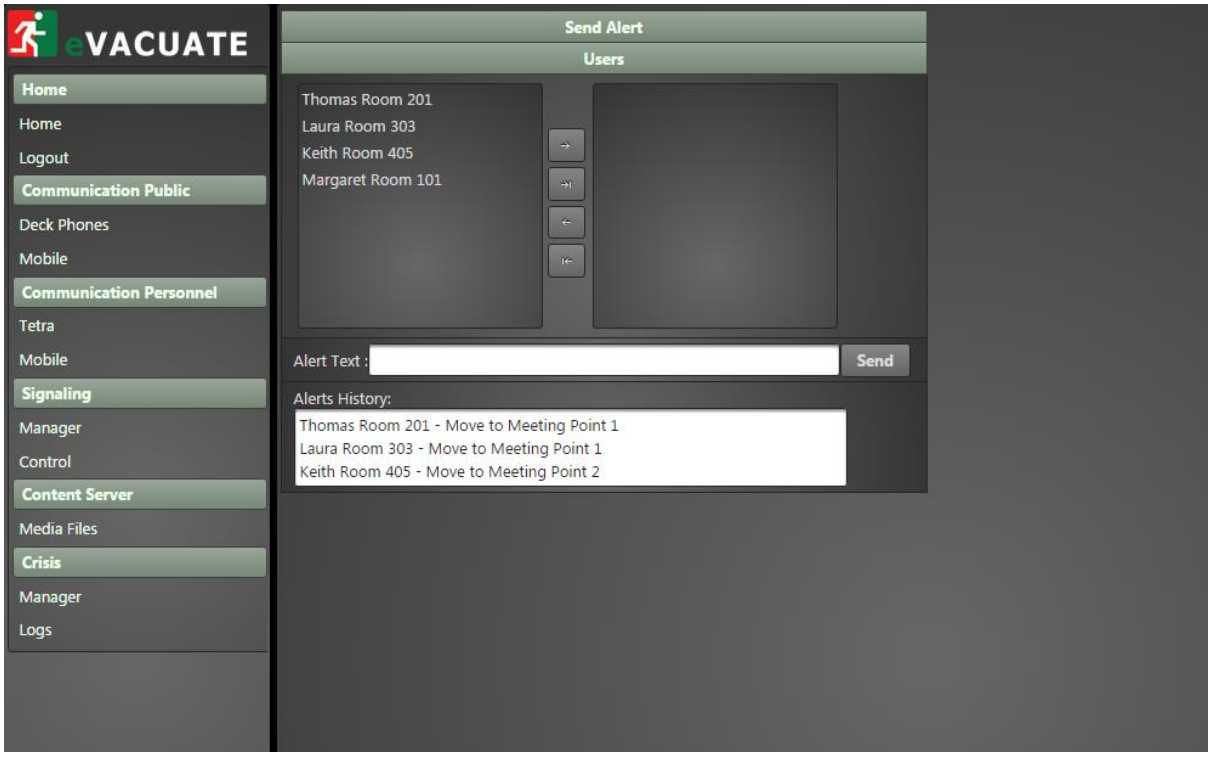

Figure 7: Screenshot of EOC's messages to FR's/Users through mobile Applications

Through eVACUATE's mobile Applications (FRs App and EVAMAPP), the EOC's Operator is also available to communicate with the FRs and any other Users. Once the EOC's operator select the user from the list box, he can send a text directly to him and inform him individually about any

kind of issue. Afterwards the received message/text is stored automatically to the Alerts History in order the Emergency Operator to be always aware of the performed communication between him and the corresponding FR or user.

NAME OF FIRST RESPONDER(s)/ SECURITY PERSONNEL/ USER NAME	
User ID name	Displays the name of the person that the EOC's Operator is willing to initiate a communication link through mobile application
Alert Text	Within this domain the EOC's Operator can type the message/alert that wants to disseminate to the selected User
Alerts history	The transmitted messages are stored in a list which is accessible from the EOC's Operator in order to check the overall chain of messages that have been sent till that time to different users through mobile app. (FRs /EVAMAPP)

Table 6: EOC's user card description

2.5.5 Delivery of messages (Voice/Text) to DECT phones /ESPA/SIP/VOIP systems.

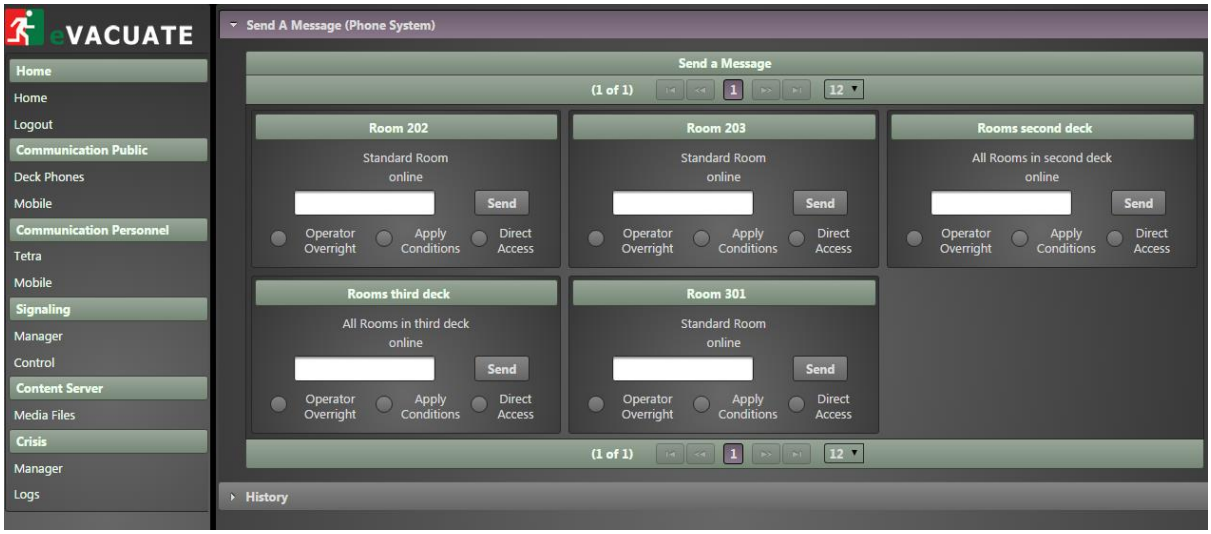

Figure 8: Screenshot of EOC's messages transmission to DECT phones/ESPA/SIP/VOIP systems

EOC allows the EOC's Operator to send an alert/message to the public using the DECT phone system already existing in end-users' premises. The alert/message is deliver over the ESPA/SIP network using the standard ESPA/SIP protocol. The alert/message can be either a predefined audio message (by attaching an audio file) or a text message depending on the situation and according to the Operator's final decision.

Checkpoint Name / Room Number	
Location ID Name	Displays the exact location where the EOC's Operator will send his message/alert
Status	Displays the current Status of the installed infrastructure (DECT phone), (e.g. Online or Offline)
Message	EOC Operator's message to be transmitted to the specific terminal
Transmission Options	• **Direct Access**: Direct communication with DECT phones using the DECT phone protocol. Using this option we have direct communication with the H/W. This is the faster way as bypass all software modules and SOFIA rules.

	• **Apply Conditions**: Controlled by SOFIA rules. Messages will be delivered based on SOFIA Rules. • **Operator Over right**: Messages are sent through the Operator Gateway bypassing the normal operation of eVACUATE Platform. Messages can have a delay as the Operator Gateway using a FIFO queue system.

Table 7: EOC's DECT messenger description

2.5.6 Management and monitoring of Digital Signs' status/content.

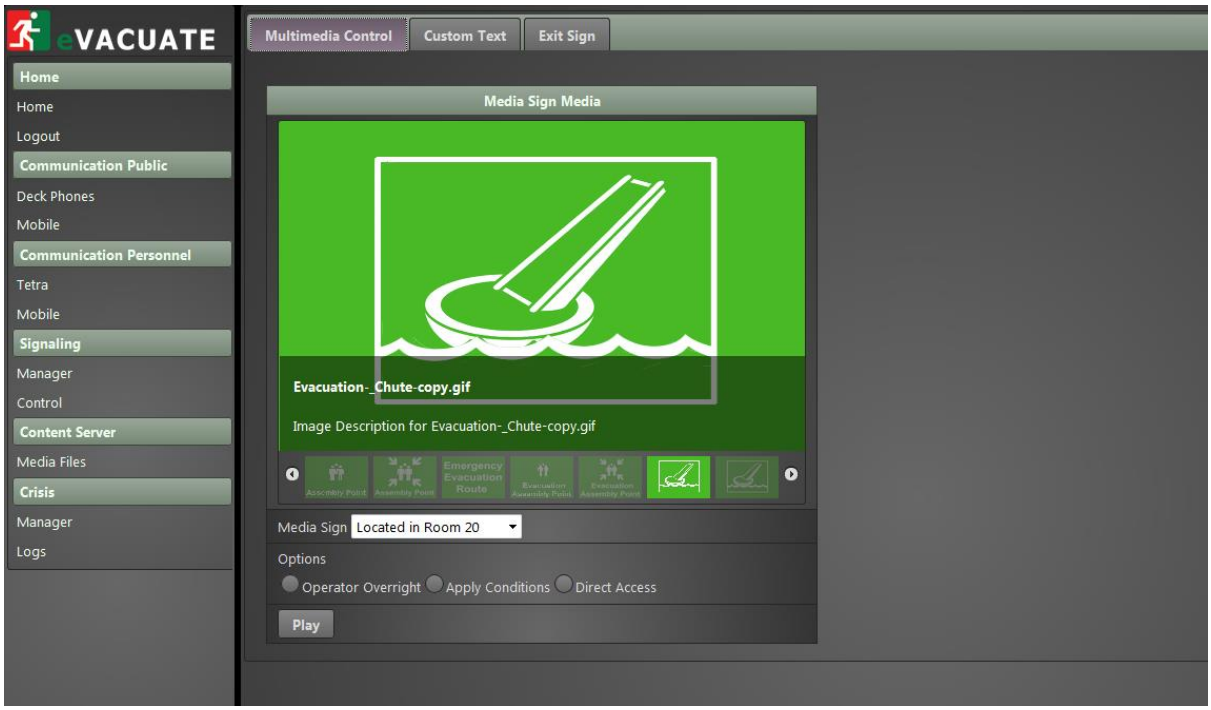

Figure 9: Screenshot of EOC's monitoring capabilities over Digital Signs status/content (multimedia control)

Within this page, the EOC's Operator can either activate a new digital sign or over right the media file currently played in an active Digital Sign by selecting a new multimedia file to be displayed. For selecting a new media file, a media gallery has been developed in which all media files are stored. The main options they are currently available in the media library are audio file, video file or static picture.

Digital sign – Media Files	
List of media signs (preview)	A list of media files is displayed under the chosen one, which can be used from the EOC Operator as media library to replace the already selected (the one shown in the central screen, maximized)
Media Sign location	Displays the location where the selected media sign is being displayed.
Transmission Options	• **Direct Access**: Direct communication with Digital sign devices. The file will be played instantly and this is the faster way to play a file on a specific Digital Sign. • **Apply Conditions**: Controlled by SOFIA rules. The command is controlled by SOFIA rules. • **Operator Over right**: Message is sent through the Operator Gateway bypassing the normal operation of eVACUATE Platform.

	Commands are transmitted on the Operator Gateway and will be executed based on a FIFO queue.
Play button	Initiates the uploading of the selected digital sign to the selected location

Table 8: EOC's media signs messenger description

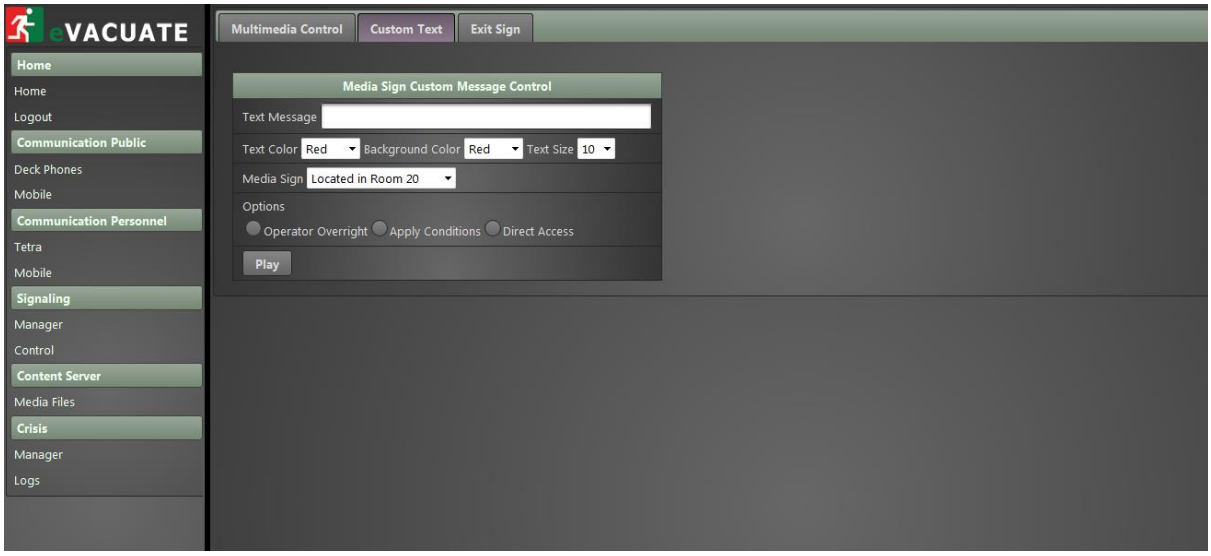

Figure 10: Screenshot of EOC's monitoring capabilities over Digital Signs status/content (custom text)

Apart from the media files, the EOC Operator can also write his own text in order to be displayed in the digital media signs installed in end-users' premises. This can be either random text or predefined text already stored in EOC's database. In particular for the predefined text, these can be e.g. instructions on how to reach a safety area or medical instructions to prevent possible injuries, etc. As in the previous page, also in this one the Operator can over right the already loaded file currently played in Digital Sign with a new one of his own will.

Digital sign - Text Files	
Text Message	Displays the content of the text that EOC's Operator wants to transmit
Text Colour/Background Colour/Text Size	Select various attributes such us Text colour, background colour and text size.
Transmission Options	• **Direct Access**: Direct communication with Digital sign devices. The file will be played instantly and this is the faster way to play a file on a specific Digital Sign. • **Apply Conditions**: Controlled by SOFIA rules. The command is controlled by SOFIA rules. • **Operator Over right**: Message is sent through the Operator Gateway bypassing the normal operation of eVACUATE Platform. Commands are transmitted on the Operator Gateway and will be executed based on a FIFO queue.
Play button	Initiates the uploading of the selected text file to the selected terminal (digital sign)

Table 9: EOC's digital signs messenger description

Figure 11: Screenshot of EOC's add/remove capabilities over Digital EXIT Signs status/content

The aforementioned screenshot displays a list of digital signs stored in EOC's database. The list consists of different digital signs of various types (text, audio, and video) as well as their status. In addition the Operator can add a new Digital sign, edit an existing one and even remove a digital sign from the system.

List of Digital EXIT Signs	
Location	Displays the location where the Digital Sign is placed
Description	Displays information about the selected Digital Sign (e.g. type)
Status	• **Active** when the digital sign is online, • **Disconnected** when there is a problem with the digital sign and • **Offline** when a digital sign is deactivated by the EOC Operator
Play now	Display the current media files that is uploaded and plays on the selected Digital Sign. In case no file is uploaded the status is shown as idle.
Actions	**Remove button**: This button gives the possibility to the EOC Operator to remove the selected digital sign from the existing list **Edit button**: This button gives the possibility to the EOC Operator to edit/modify the selected digital sign from the existing list

Table 10: EOC's list of digital signs description

2.5.7 Management and monitoring of Exit Signs' status

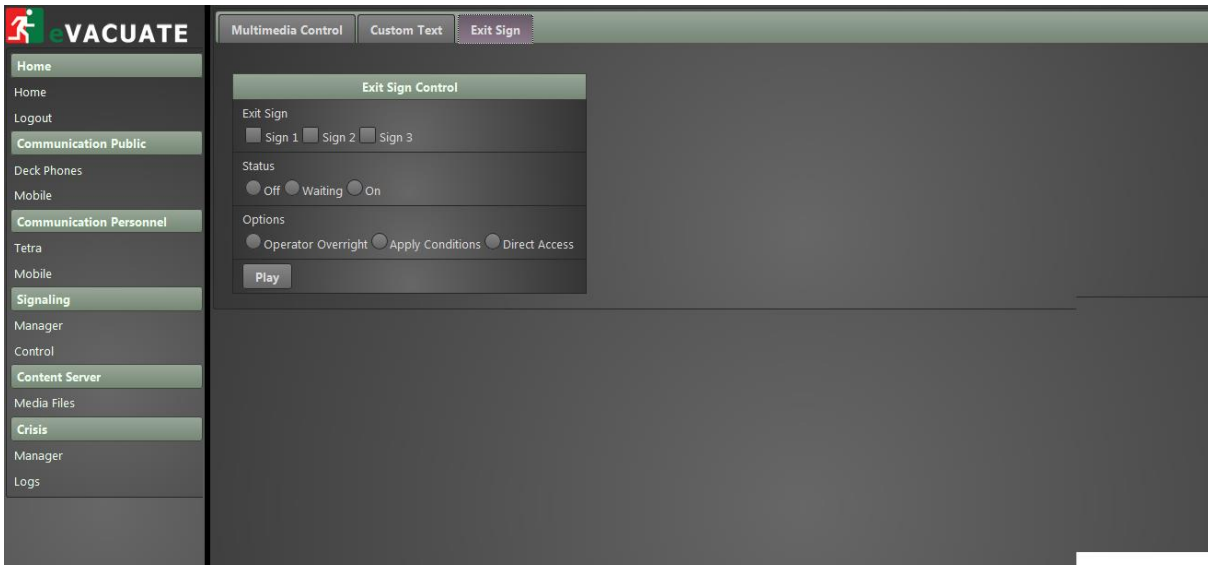

Figure 12: Screenshot of EOC's management and monitoring of Exit Signs' status

Through EOC the operator can also control the Exit Signs. A list of all the available Exit Signs is displayed in which the EOC Operator can choose between different statuses of operation. With this frame, an exit sign be either ON, OFF or even IDLE based on the outcomes of the current AER.

Exit Sign Control	
EXIT sign	List of available digital exit signs installed at end-users' premises
Status	• **OFF**: The digital EXIT sign is deactivated • **WAITING**: The digital EXIT sign is flashing • **ON**: The digital EXIT sign is activated
Transmission Options	• **Direct Access**: Direct communication with Exit sign devices. The command will be executed instantly this is the faster way to control an exit sign device. • **Apply Conditions**: Controlled by SOFIA rules. The command is controlled by SOFIA rules. • **Operator Over right**: Message is sent through the Operator Gateway bypassing the normal operation of eVACUATE Platform. Your command is transmitted on the Operator Gateway and will be executed based on a FIFO queue.
Play button	Uploads on system the selected mode of the specific EXIT sign that was chosen by the EOC Operator

Table 11: EOC's exit signs control description

2.5.8 Management and monitoring of active media files (Video / Pictures / audio)

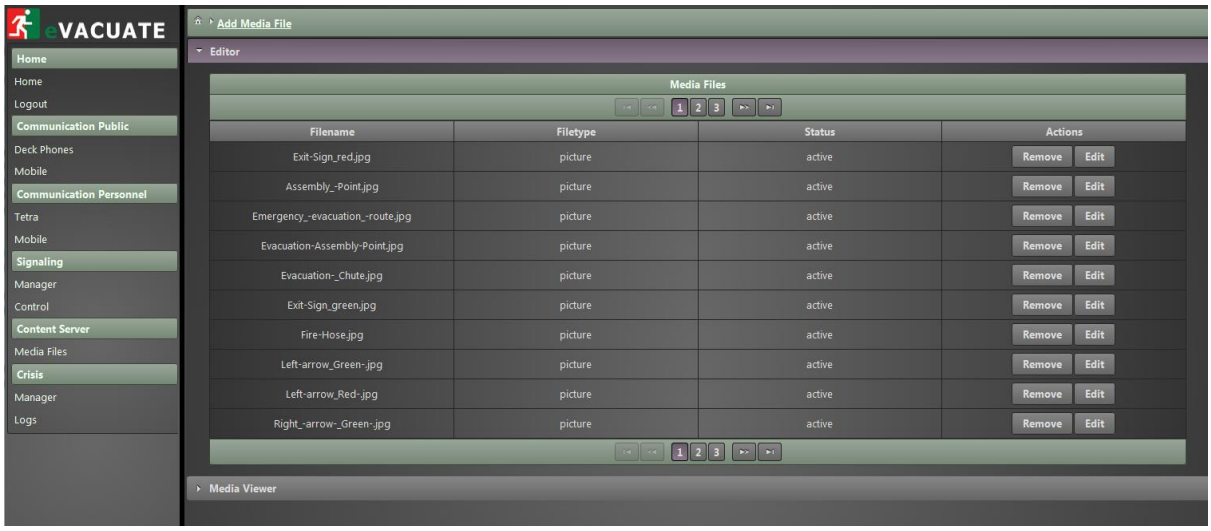

Figure 13: Screenshot of EOC's add/remove capabilities over Digital Media Files status/content

The screenshot above allows us to manage the media files that can be used for the digital sign. The Operator has the option to remove a file from the file server as well as to edit its attributes (file type and status).

Media Files	
File name	Display the name of the file as stored in the file server
File type	Displays the type of file (picture, video, audio)
Status	Displays the current status of the selected media file. • **Active** when the media file is in use, • **Offline** when a media file is not used
Actions	**Remove button**: This button gives the possibility to the EOC Operator to remove the selected media file from the existing list **Edit button**: This button gives the option to the EOC Operator to edit/modify the selected media file from the existing list

Table 12: EOC's media file list description

2.5.9 Data analysis outcomes from deployed Sensors.

The Data analysis is based on Kibana[4]. Custom queries and visualizations are pre saved and the operator can create a custom dashboard very easy. The figure below present the standard Dashboard view for the temperature and humidity sensors of eVACUATE platform.

[4] (https://www.elastic.co/guide/en/kibana/current/index.html).

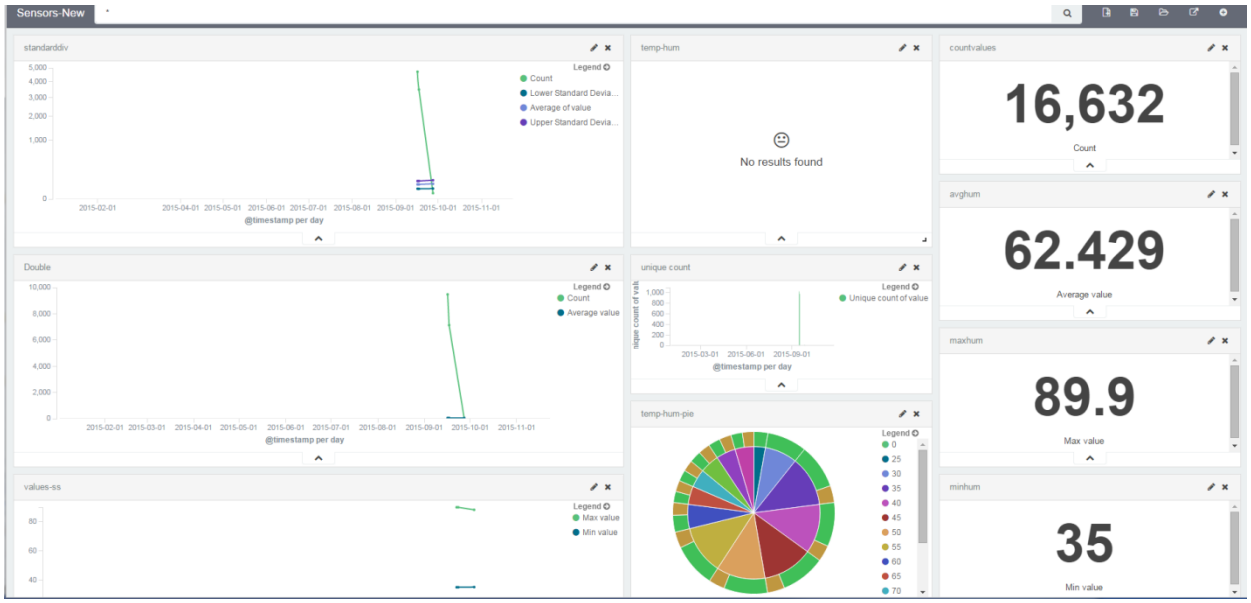

Figure 14: Screenshot of EOC's Data analysis from deployed sensors

With the exploitation of this feature, the EOC Operator is able to track the operation of all deployed sensors while keeping some historical data over the measurements that have been recorded by each sensor during the normal operation as well as when a crisis is evolving.

3. EVACUATE EOC Implementation

3.1 General information

eVACUATE is a platform integrating various advanced Information and Communications Technologies (ICT) in view of developing a data-centric platform fulfilling two core objectives:

1 **to collect** information from the crisis theatre – field of operations – e.g., from sensors, **process** it, **extract** inferred information and **present** it (in raw or inferred form) to people away from the field; the objective is to increase their awareness level concerning of what is actually happening at the field (Monitoring Flow) and thus aid them in decision making.
2 Secondly to **enable the dissemination** of appropriate information from central points (where decision makers are located) **towards the field of operations**, enabling in such a way the efficient organization of the resources involved in the actual emergency handling operations (Control Flow).

From the technological perspective, eVACUATE could be separated into two major sectors:

1 The information platform.
2 The real time communication infrastructure

The Information platform is the systemic part of the entire platform that is overlaid the communication infrastructure and is responsible for the actual processing of the data exchanged among the system nodes and their presentation to end users.

The real-time communication infrastructure bundles all communications hardware and software used to enable the communication among the various nodes of the system. eVACUATE is in principle an All-IP communication platform, with this meaning that an IP network is exploited over which voice, video and data communication services are offered.

The following section refers to this information management platform and more specifically to the presentation part for the Monitoring Flow on the one hand, and to the management of the disseminated information (mainly in the form of composition of appropriate messages) across the platform nodes on the other.

3.2 EOC Architecture – Information Platform

The eVACUATE EOC is a set of components based on 3 Tiers /Layers architecture:

- **Database or Data Model**
- **API which includes the Business Logic and Data Access**
- **User interface or UI**

The 3 Tier architecture is describes on the following sections. The diagram below shows a generic view of such an architectural design.

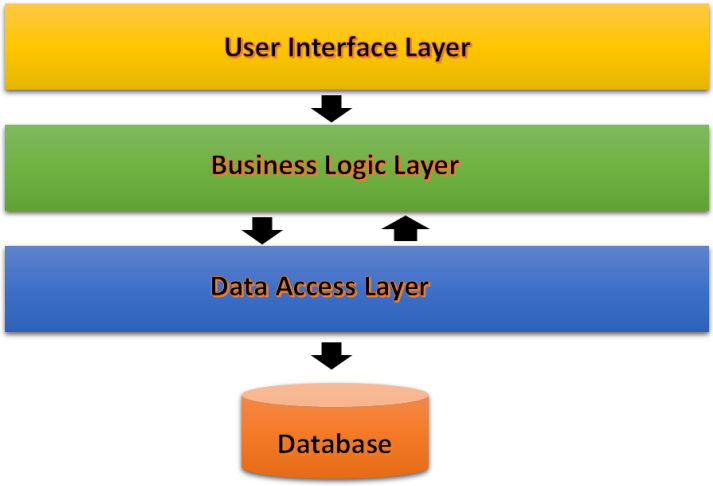

Figure 15: 3Tier Architecture of eVACUATE EOC

3.2.1 Database of Data Access Layer

The database layer is implemented with a Relational Database Management System (RDBMS), accessed via Java Persistence API. It is thus RDBMS vendor independent. With the use of Java Persistence API, introduced in Java 1.5 and J2EE version 5, no database-specific commands in SQL are used (thus the platform is easily portable from one database to another).The main pieces of information that need to be persistent in the system can be categorized as:

- **eVACUATE information**, which refers to metadata about the entities of the eVACUATE domain model, as well as the interrelations between them.
- **Rules information**, which refers to metadata about values or combination of values for the entities of the model (SOFIA rules).
- **Security information**, which refers to user management, private and public keys and the user authorization.

3.2.2 Data Access Layer

The data access Layer consists of web services which are designed to interoperate as a Service Oriented Architecture (SOA) system. This layer consists of a set of web services, which can be divided into the following subsets:

1. **Data Access Services**.
    a. For almost all entities defined in the Domain model, a series of RESTful operations have been defined for performing all necessary Create, Read, Update and Delete actions. For the implementation of each operation the respective POST, GET, PUT and DELETE RESTful methods have been used.
    b. Workflows for complex tasks.
2. **Security Services.**
    a. Authorization.
    b. Authentication.
    c. Encryption/Decryption.
    d. Digital Signature.
3. **SOFIA related Services**.
    a. Event listener Server.
    b. Event Persistence.

All web services are implemented by using APIs provided by the Restful Platform. The use of this Java platform allows for hardware and operating system independency as well as conformance to the Web Services and related standards. For the implementation of the eVACUATE platform the Restful Web Service stack [5]has been used. J2EE[6] version 7 annotations have been used, thus allowing for faster development and easier code management and maintenance.

3.2.3 Business Logic/Controller Layer

This layer is responsible for the implementation of the central, heavy-duty tasks of the system. It primarily consists of two modules; the Device module and the database (controller) module.

The Data Fusion Engine is mainly responsible for validating an event (sensor envelope or action), via the appropriate web service, and using the event details and a set of rules that apply to this event. For that reason the module must have access to the persistence layer to get the required information. The module is implemented in the Java programming language to ensure maximum platform independency and extensively uses the Drools framework. For the Business Logic Layer, while a set of java classes are used. A series of Flows and BPMN (Drools version) [7]enables the dynamic (on-the-fly) update of the business logic using graphical modelling tools, such as Eclipse with Drools, and deploy the new diagram into the Rules Repository (Drools Guvnor).

3.2.4 User Interface Layer

This layer is responsible for the implementation of the Graphical User Interface (GUI). Its core design is designed based on Europa's IPC Guidelines. It also be provided in multiple languages for the partner countries and easy interface for adding new countries if required.

Java Server Faces (JSF[8]) technology is used for the implementation of the web pages. JSF allow us to combine the power of JavaScript and the visual components of PowerFaces[9] and RichFaces[10] and other open source visual components to create a dynamic interface, which will ensure an easy to use implementation for the end user and more attractive visual design.

3.3 EOC's Component Diagram

EOC is a set of components that manage all the infrastructure of eVACUATE platform. Each H/W component communicates with SOFIA by the use of DDS and Smart Space Agents. Each component wants to have access to the device can communicate with it using the SOFIA API. Also the EOC GUI can gain direct access to the H/W Devices by the use of Operator/PPDR Gateway which takes over the control of the local communication gateway and gain direct access to most H/W Components. The Overall Component Diagram is presented below which is based on the diagram on section 2.1.

[5] https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html
[6] http://www.oracle.com/technetwork/java/javaee/overview/index.html
[7] http://www.drools.org/
[8] http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html
[9] http://www.powerfaces.org
[10] http://www.richfaces.org

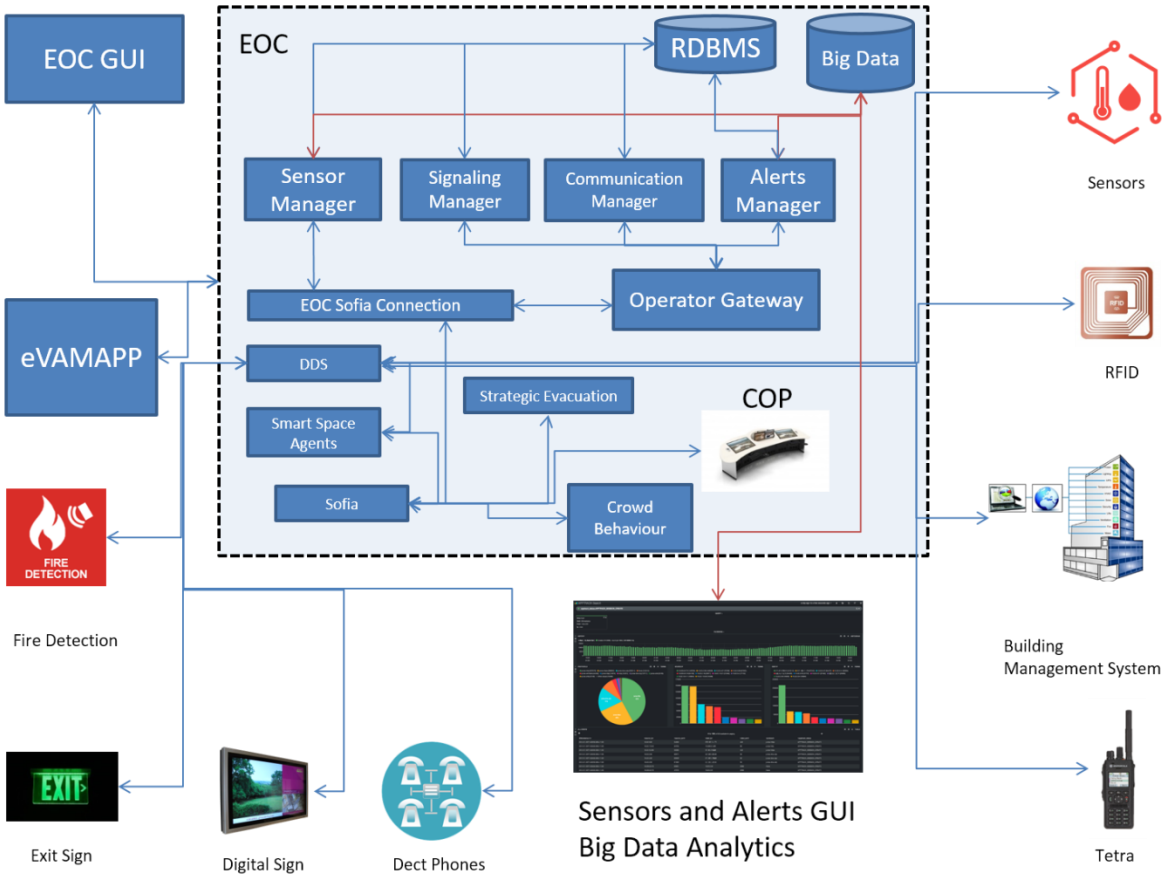

Figure 16: EOC's detailed Component Diagram

EOC's component	Description
RDBMS	
Big Data	See section 3.3.6
Sensor Manager	See section 3.3.2
Signalling Manager	See section 3.3.3
Communication Manager	See section 3.3.4
Alerts Manager	See section 3.3.5
EOC's SOFIA Connection	See section 3.3.1
Operator Gateway	See section 3.3.7
DDS	See section 3.3.12
Strategic evacuation	See section 3.3.9
Smart Space Agents	See section 3.3.10
SOFIA	See section 3.3.11
Crowd behaviour	See section 3.3.8

Table 13: EOC's Components description

3.3.1 EOC Sofia Connection

The SOFIA Connection handles the alerts, TETRA messages, DECT phone messages, and mobile applications. It provides a middleware layer which convert SOFIA calls to Restful calls and

vis versa. All internal communicate is realized with RabitMQ [11]and external communication take place using the Operational Gateway. The SOFIA Connection provides restful API for communication with the mobile apps and any other systems that want to have real time communication with the users/operators (e.g. COP).

Typical workflow allows the EOC Operator to publish an alert using the GUI or a mobile user to publish an Alert using the web service API.

The diagram below presents the component diagram for the Real Time Communication System.

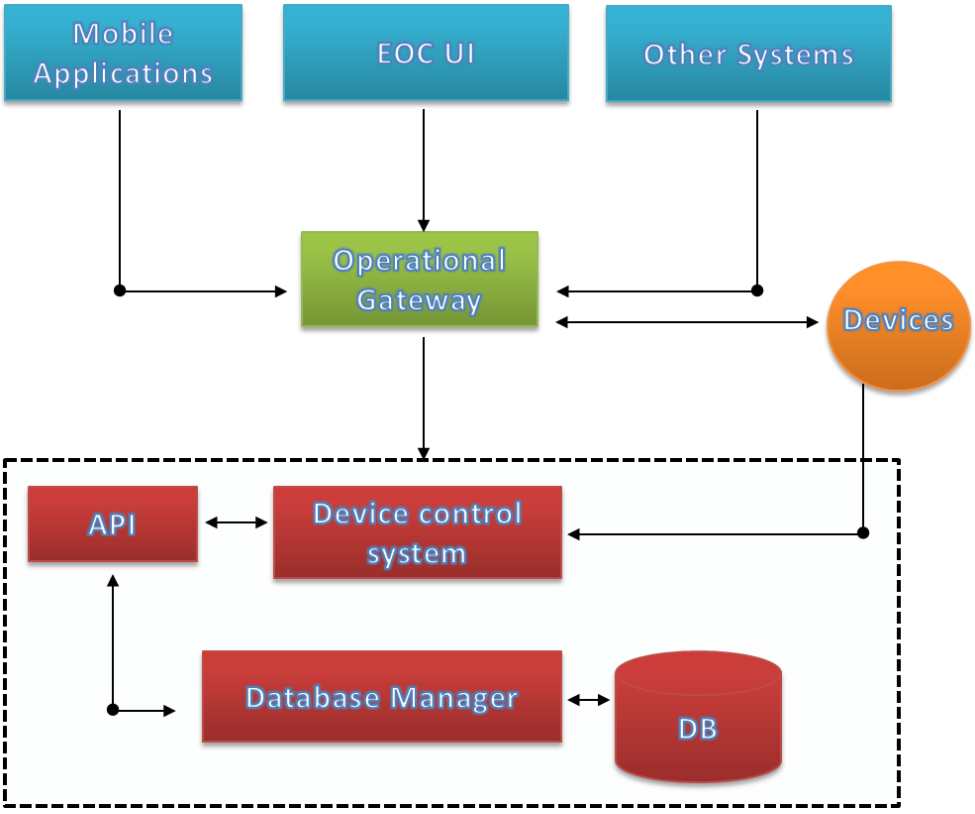

Figure 17: EOC's Real Time Communication System

3.3.2 Sensor Manager

The Sensor Manager is a set of components which are responsible for managing the Legacy and External devices. The main channels of communication is DDS and SOFIA. Each external device communicate with a gateway using a TCP/IP protocol and the gateway is responsible for storing and performing any necessary preprocessing before sending the data to SOFIA using the DDS protocol.

Dynamic Exit Signs Gateway Module

The following diagram describes the internal architecture of the Exit sign's subsystem. Based on the figure 17, the EOC Real Time communication System receiving the Sophia messages as the Exit sign is part of the other systems as shows in Figure below.

[11] https://www.rabbitmq.com/

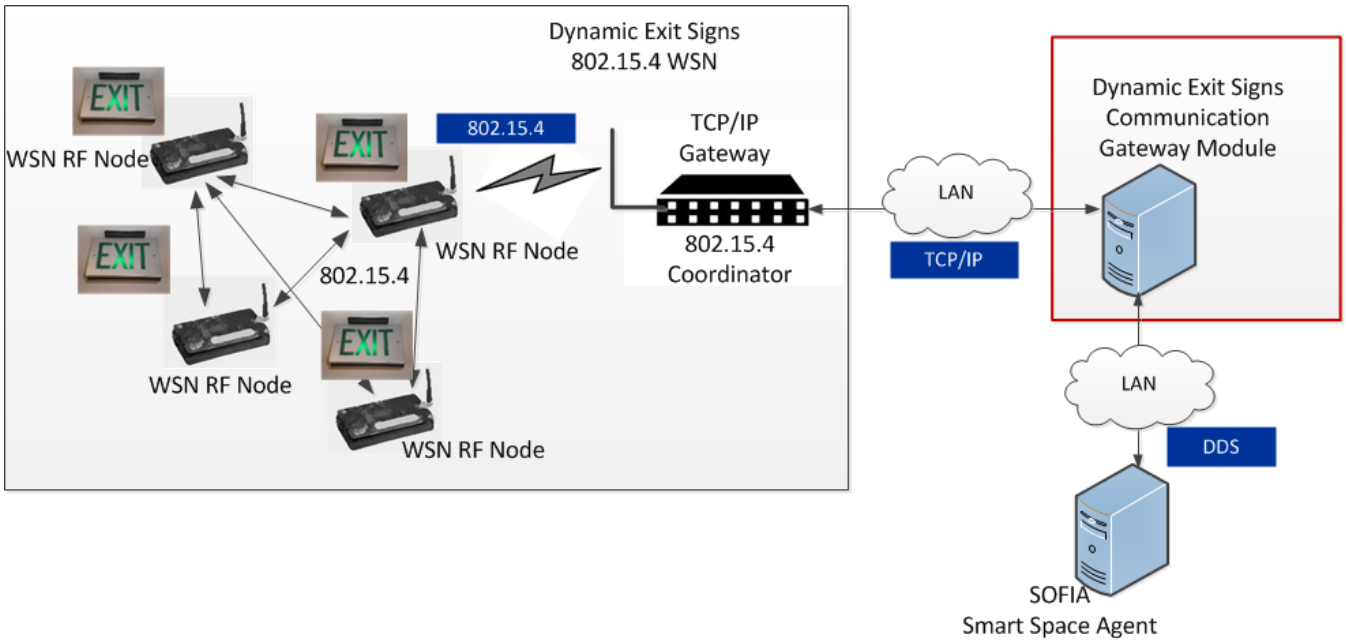

Figure 18: Connectivity of Dynamic Exit Signs

Communication protocol among the WSN RF nodes and WSN coordinator/gateway is based on 802.15.4 protocol. The communication between the WSN coordinator/gateway and the Dynamic Exit Sign Communication Gateway module is based on TCP/IP. The digital exit signs communicate with the Dynamic Exit Signs gateway with TCP/IP, the gateway store and performed basic preprocessing of the data and then transmit the data using the DDS Protocol to Sophia Network. The Dynamic Exit Sign Communication Gateway module is an application developed in C++. The Dynamic Exit Sign Communication Gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

Environmental Sensor Communication Gateway Module

Based on the figure 17, the EOC Real Time communication System receives the SOFIA messages as the Sensors are part of the other systems shown in Figure 19. An overview of the architecture is presented with the form of a diagram.

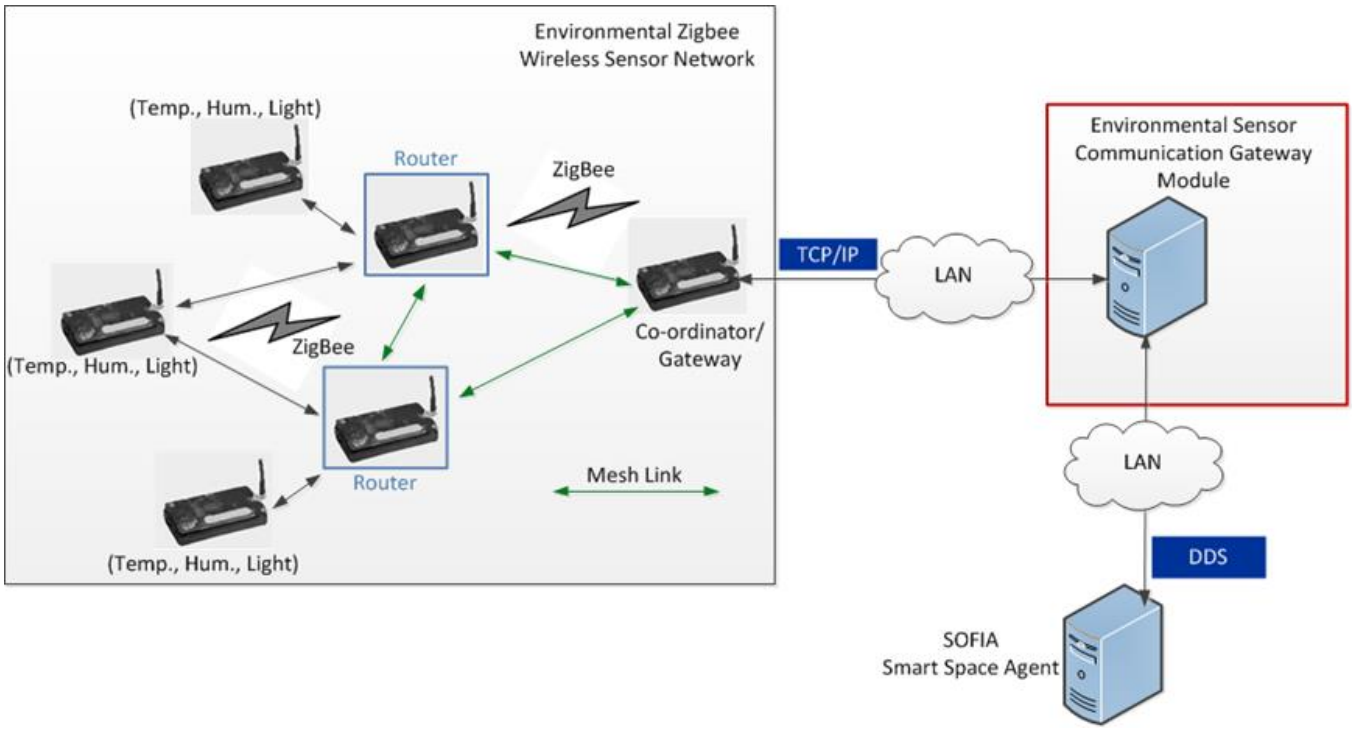

Figure 19: Environmental (Temp., Hum., Light) WSN Connectivity

Communication between the WSN coordinator and the Environmental Sensor communication gateway module is based on TCP/IP. At the Environmental Sensor Communication Gateway module a TCP/IP client has been embedded that uses the "Kaleidos[12]" profile to communicate with the Environmental Co-Ordinator/ Gateway.The Environmental Sensor communication gateway module is an application developed in C++.

The Co-ordinator/Gateway communicates with the Environmental Sensor Communication Gateway Module with the use of TCP/IP, the Gateway module store and performed preprocessing of the data and with the use of DDS transmit the data to SOFIA network.

The Environmental Sensor communication gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

RFID Communication Gateway Module

The overview of the architecture is presented below with the form of a diagram.

[12] https://github.com/kaleidos/grails-admin-interface

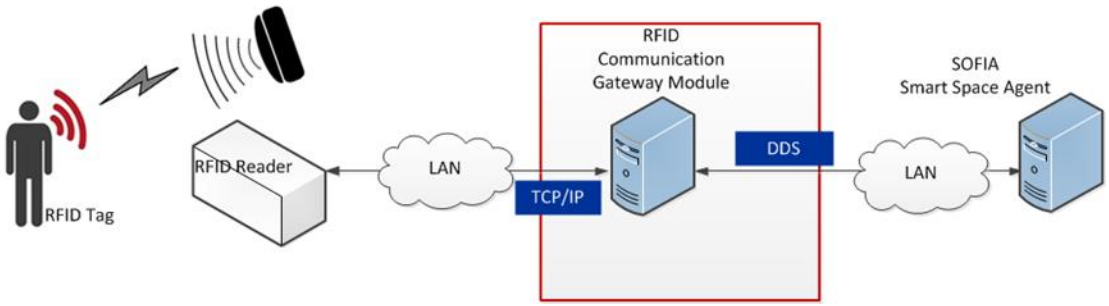

Figure 20: Connectivity of RFID system

The RFID communication gateway module consists of a TCP server that receives information from the RFID reader's TCP client. The RFID communication gateway module is an application developed in C++. The RFID communication gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

Building Management System Module

The Architecture and communication diagram of the management of the building is show below with a form of a diagram.

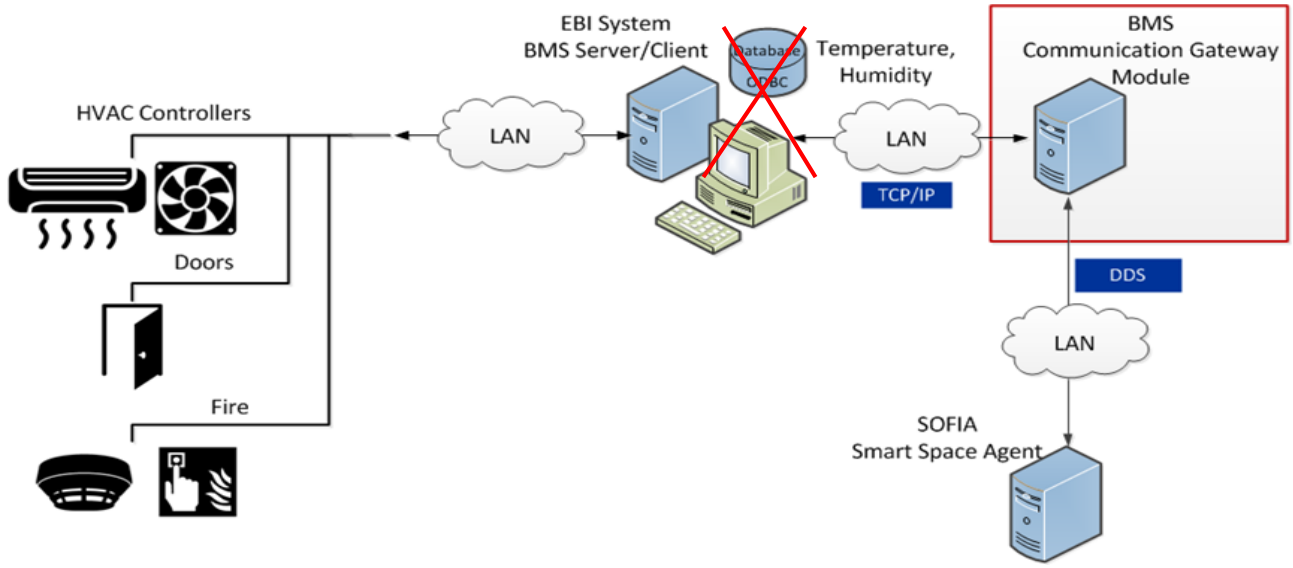

Figure 21: Building Management System Module (BMS)

The BMS communication gateway module is running on the BMS server and retrieves the data from the BMS by accessing a locally stored file (e.g. excel). (Note this is different from the above figure).The BMS communication gateway module is an application developed in C++.

The BMS communication gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

3.3.3 Signalling Manager

The Signalling Manager consists of the media manager and the device manager. The media manager is a media server that allows the operator to store and manage multimedia files. Multimedia files can be video files, images and sound files. The video and image files can be used by the digital signs and IPTV and the sound files can be used by the public announcement.

The Media manager consists of the following components:

- File repository, based on HDFS files are stored on a cluster based format.
- API to manage the files and be able to stream the files. This is a restful and soap API providing all functionality for manage and streaming.
- A GUI which allow us to upload a file, download and perform CRUD (Create – Read - Update - Delete) operations.

The diagram below presents the component diagram for the Media Manager module.

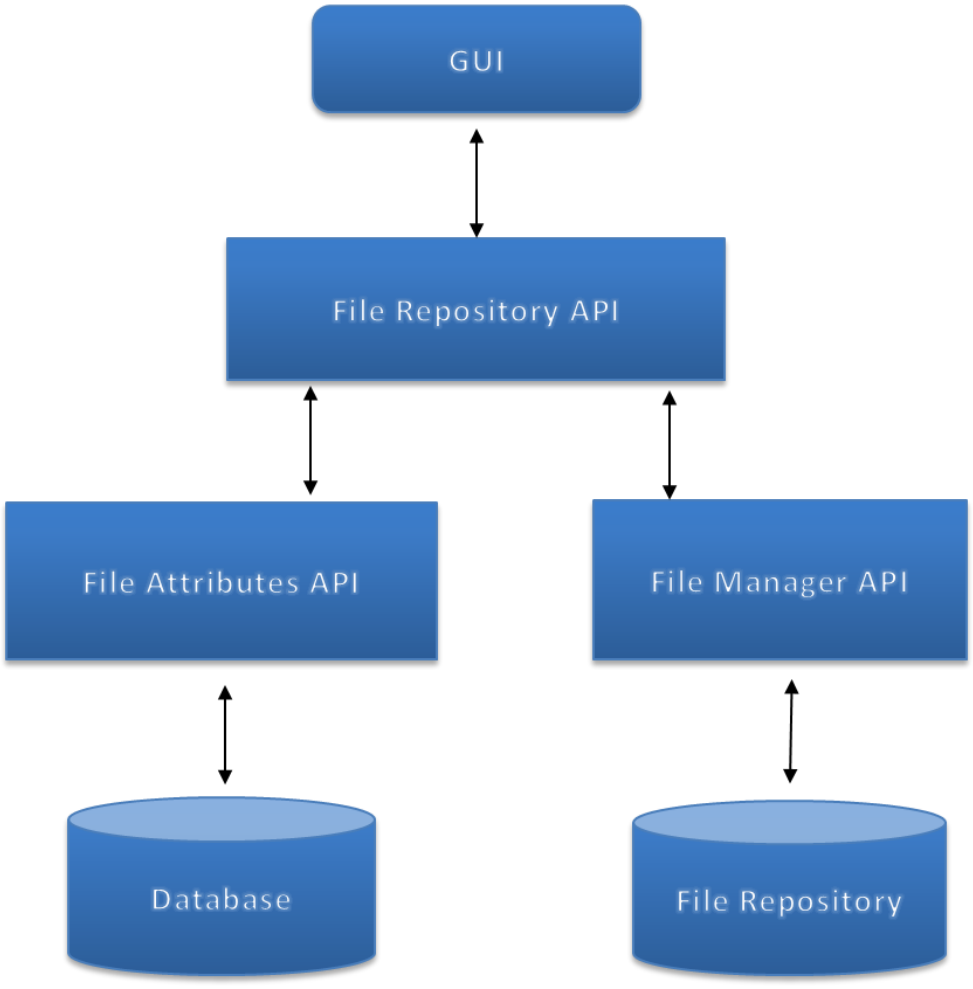

Figure 22: Component Diagram of Media Manager Module

The device manager system comprises of following hardware and software components:

- Digital sign content server that stores the media content (predefined text, audio, image or video messages)
- Open Source HW Media Player based on a dual core ARM processor (Advanced RISC Machine), running a modified version of Ubuntu (cubieboard3/cubietruck[13]) that can connect to LCD screens, TV systems and audio devices
- Digital Sign web service API's
- SW on the media player that enables the media player to play multimedia files on request

The Digital Sign system uses web services for its software modules and its backend is based on a three-tier architecture:

- Database model which is managed by a series of hibernate classes
- A web service API, which is developed as a series of SOAP Web services to perform CRUD (Create, Read, Update and Delete operations) on the entities.
- Business logic API, which is developed as a series of SOAP Web Services that allow us to control the play/stop functionality of the Digital Signs.

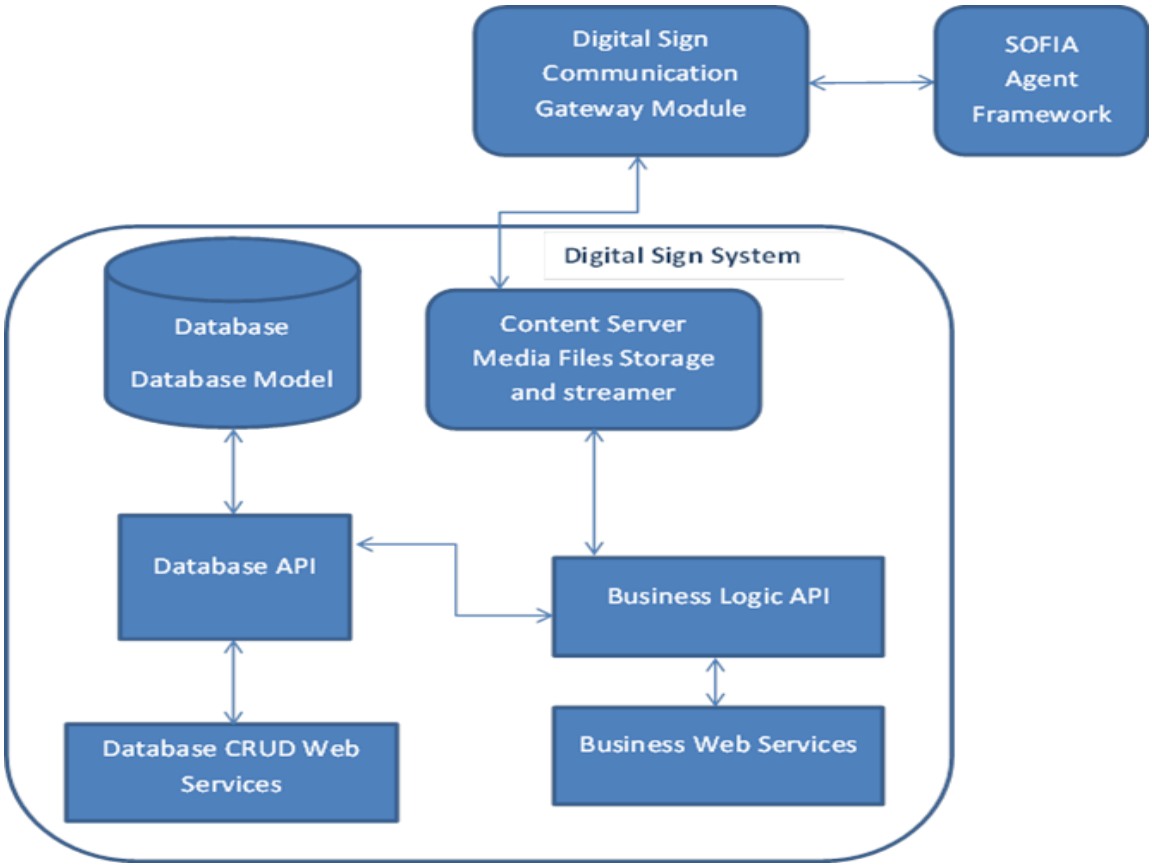

Figure 23: Digital Sign system architecture

The following diagram describes the architecture of the Digital Media Sign.

[13] http://www.cubietruck.com/

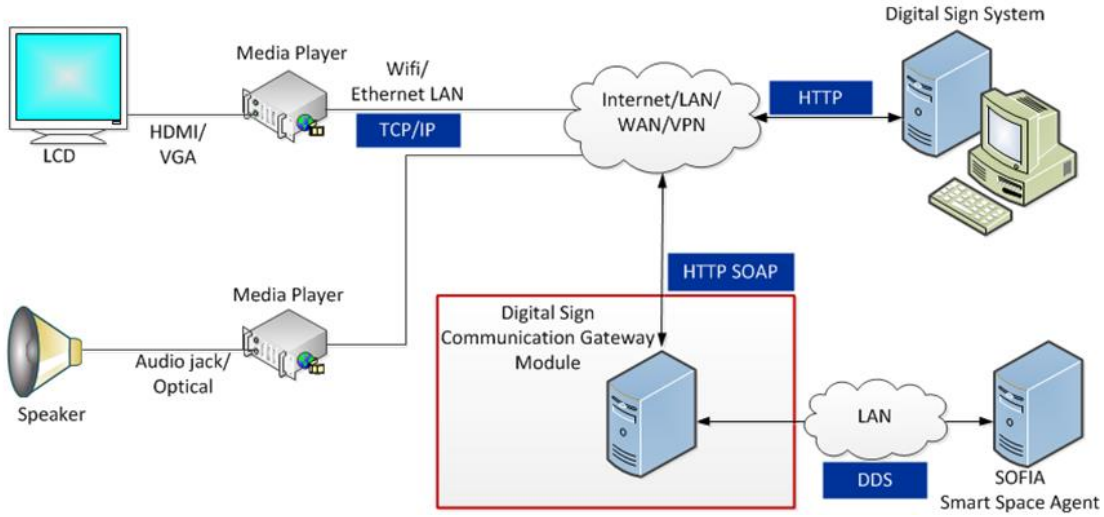

Figure 24: connectivity of Digital Sign System

The media player has a WiFi and LAN network connectivity for connecting to the Digital Sign system as well as Bluetooth and 2 USB ports for mouse/USB disk. Connectivity to the multimedia devices (such as LCD screens, monitors) is supported by VGA and HDMI interface. Connectivity to audio devices is supported by an audio jack and optical interface.

Each digital sign communicates with the Digital Sign Gateway Module which is responsible for translating the data using the DDS and then transmitting to SOFIA Network. Also the Gateway Module provides the API for the Digital Sign GUI.

The digital sign system is connected to the following systems:

- METB speaker system
- STX IP TV system
- STX Public address system

The Digital Sign Communication Gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

3.3.4 Communication Manager

The Communication manager is responsible for the tetra and DECT phones communication. In the following sections we present the two options for communication.

An overview Architecture of the tetra communication is show below with the form of a diagram.

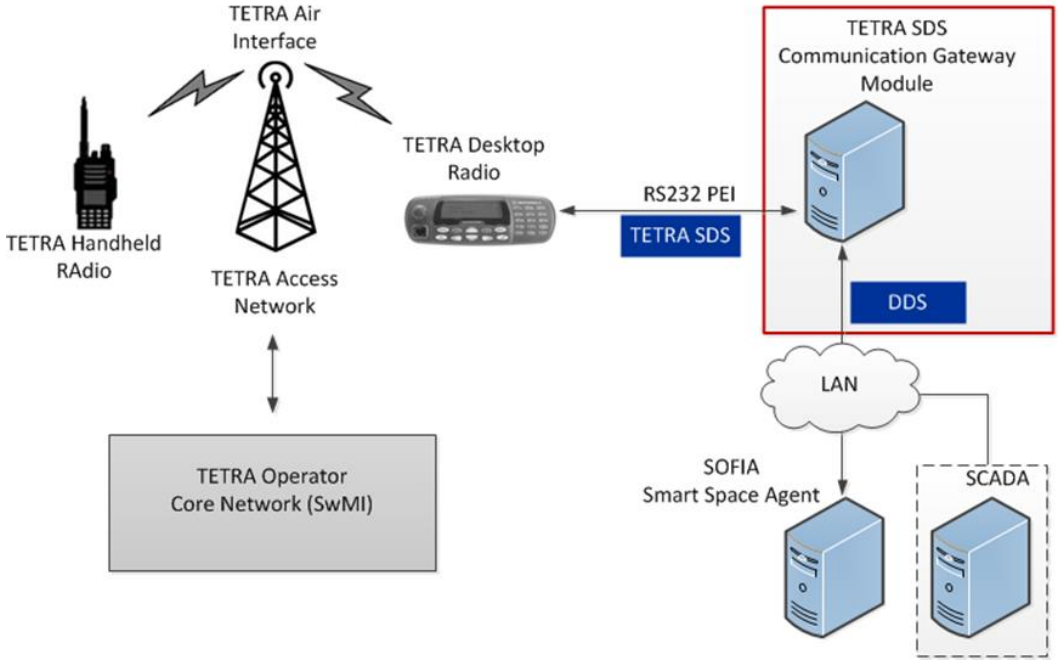

Figure 25: Connectivity of TETRA system

TETRA desktop radio is connected via RS232 interface to the TETRA SDS communication gateway module. The TETRA SDS communication gateway module is an application developed in C++. The TETRA SDS communication gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

STX Fire detection System & DECT phones

Below is presented the Connectivity diagram for the STX fire detection system & Dect Phones.

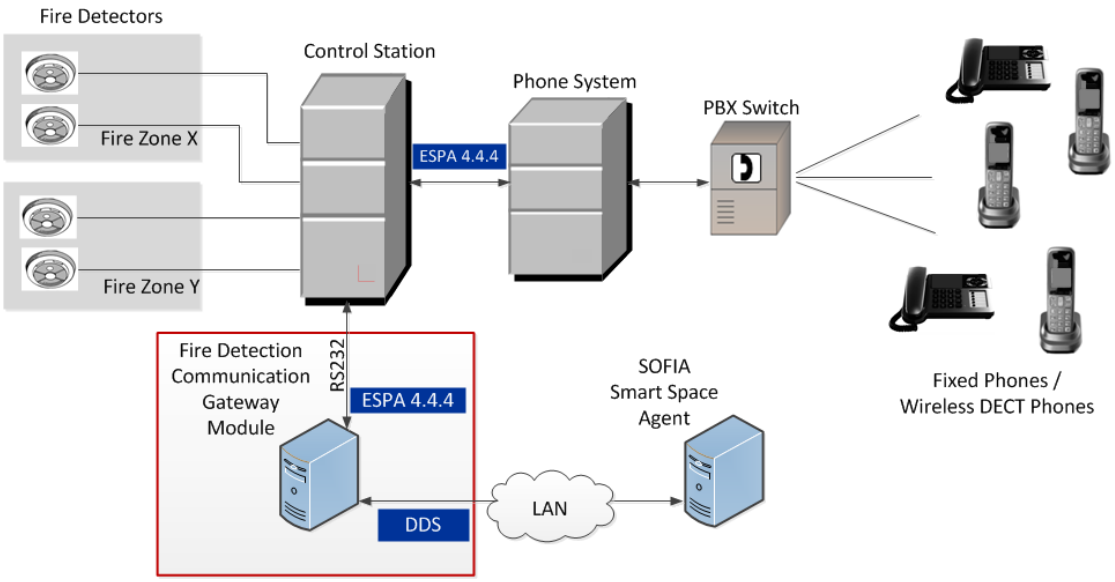

Figure 26: STX fire detection/phone system connectivity with communication gateway module

The connection between the fire detection system/phone systems and the fire detection/phone communication gateway module is based on RS232. The fire detection/phone communication gateway module is an application developed in C++.

The fire detection/phone communication gateway module communicates with Smart Space Agent via TCP/IP, and the messaging technology is based on DDS (http://www.opendds.org/).

3.3.5 Alert Manager

The Alert manager is responsible for receive Alerts that been publish in the eVACUATE platform and to publish an Alert in the eVACUATE platform. The module communicate with SOFIA and Operator Gateway. Also the eVAMAPP communicates with the Alert manager to publish an Alert.

All Alerts are stored into the Big Data component and they are accessible for data analysis and searching using the Kibana Big Data GUI.

The Internal component diagram is presented below.

Figure 27: EOC's Alert Manager Module - component diagram

3.3.6 Big Data

The eVACUATE platform is dealing with a large set of Data coming from sensors. This data are classified as Big Data and to store and analyze such datasets we need to develop a Framework and infrastructure to store and analyze this volume of data.

The EVACUATE Big Data Framework (EBD) is a set of technologies, platforms and tools used for big data analytics to store, analyze and visualize data. For storage a HDFS, Hadoop[14] database is used, Storm[15] is used as Analytics framework and also R[16] is used in a close cooperation with Storm. For visualization/Indexing elastic search and Kibana[17] has been adopted.

The sensor reading acquire and processing can be summarized in the following steps:

1. Data Import in Hadoop
2. Data Analysis and Transformation using Spark or ad-hoc tools
3. Metadata stored in Hadoop, using Hbase, Hive or Elastic Search index
4. Create dashboard and visualize for mass consumption

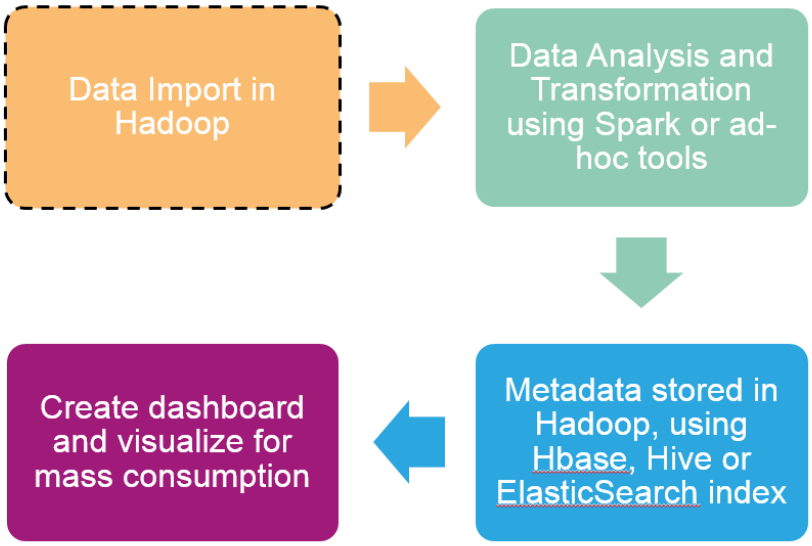

Figure 28: Data Analytics Process.

For the architecture of the eVACUATE big data framework we follow the 3Tier architectural model. The 3 steps are **data storage**, **analysis** and **logic** and final the **visualization**.

- **Data storage**: We use two repositories. One for unstructured data and one for structured data. Unstructured data can be stored in Hadoop and structured data in Hive or HBase[18].
- **Analysis and Logic**: Analysis and process can be performed using storm, spark and R.

[14] http://hortonworks.com/
[15] http://hortonworks.com/
[16] https://www.r-project.org/
[17] https://www.elastic.co/guide/en/kibana/current/index.html
[18] http://hortonworks.com/

- **Visualization**: Visualization can be archived using kibana and elastic search.

The following diagram describes the internal architecture:

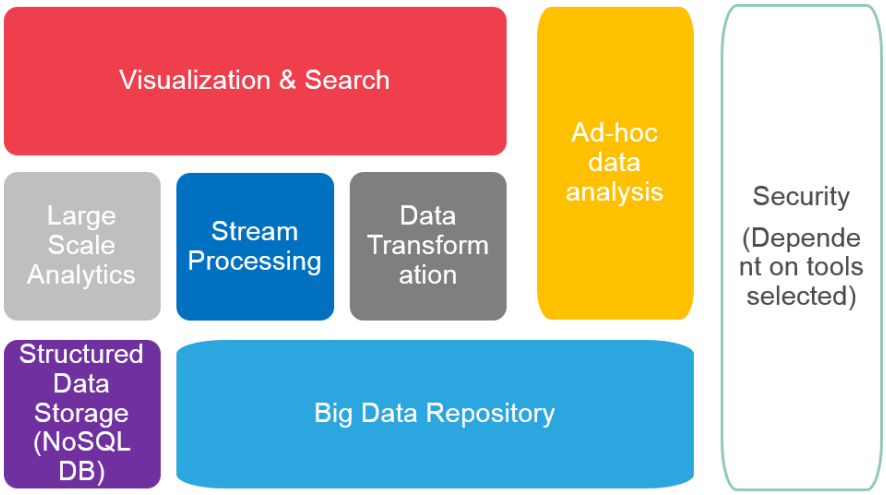

Figure 29: EOC's big data framework architecture – block diagram

The Structured Data is a Mongo DB database which is stored a Metadata and file attributes. The Big Data Repository is based on Hadoop in distribution environment and support scaling up to 128 nodes. The Large scale analytics is based on Storm and support analysis in historical data and in current data (Live data stream). Stream Processing is done with the use of Storm and is responsible for the live streaming of sensor reading processing. Finally the Data Transformation is done with the use of Storm and R and is basic a unit transformation from Celsius to Fahrenheit and other unit conversion. For Ad-hoc we use elastic search which is create indexes for fast search and helper for the visualization tool. And finally to visualize the data we are using kibana.

3.3.7 Operator/PPDR Gateway

The Operator Gateway is the main communication channel for EOC Portal, allow the operator to take over the control of the local communication gateway that is controlled by SOFIA and manually control each of the following devices:

- Exit Sign. Allow to set an exit sign to On/Off or flashing
- Digital Sign. Allow to play a specific file to Digital Exit Sign.
- DECT Phone system. Allow to play or send a message to a DECT Phone.
- TETRA Devices. Allow to send a message to specific TETRA terminal.

The internal Architecture of the Operator and PPDR gateway is presented below with a form of a diagram.

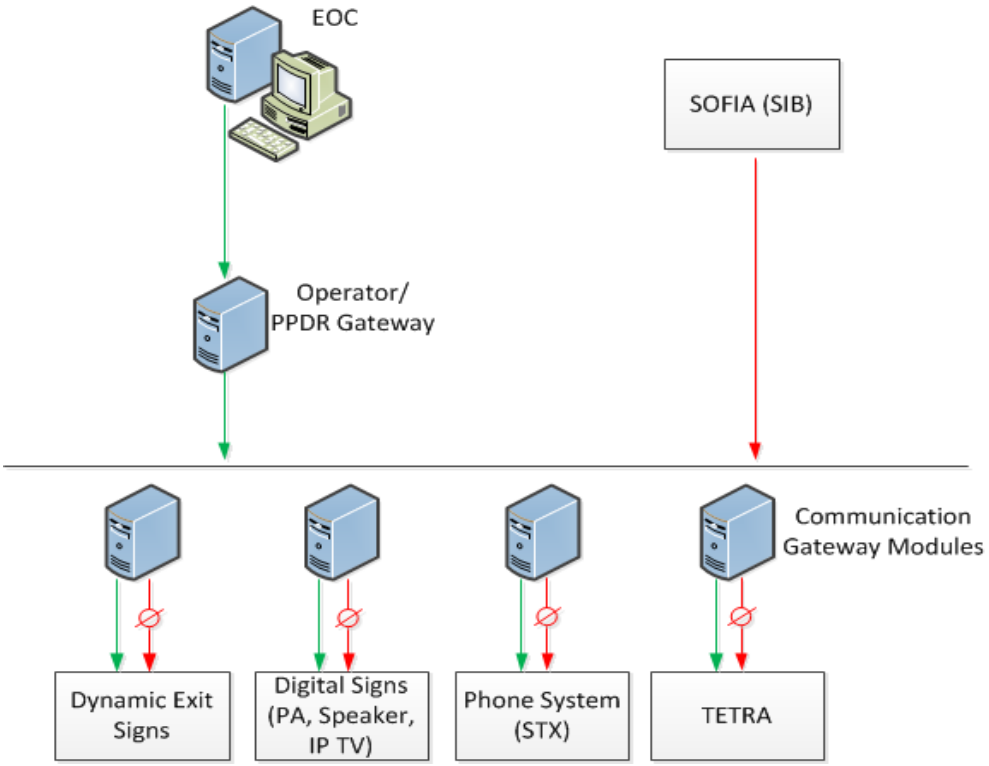

Figure 30: Operator/PPDR gateway module (functional diagram)

Operator/PPDR gateway can be accessed using RESTful web services. The RESTful web services may be deployed in any of the widely used servlet containers. In our case Tomcat 7 has been chosen. EOC, and any other component, can securely communicate with Operator/PPDR gateway using HTTPS.

The network architecture of communication gateway module is presented on the figure below. One physical server acts as a host for virtual machines (VMs), using the virtualization capabilities of modern CPUs, and one physical backup/mirror server for seamless failover in case of a malfunction in the first server. The architecture will be based on high availability server cluster architecture.

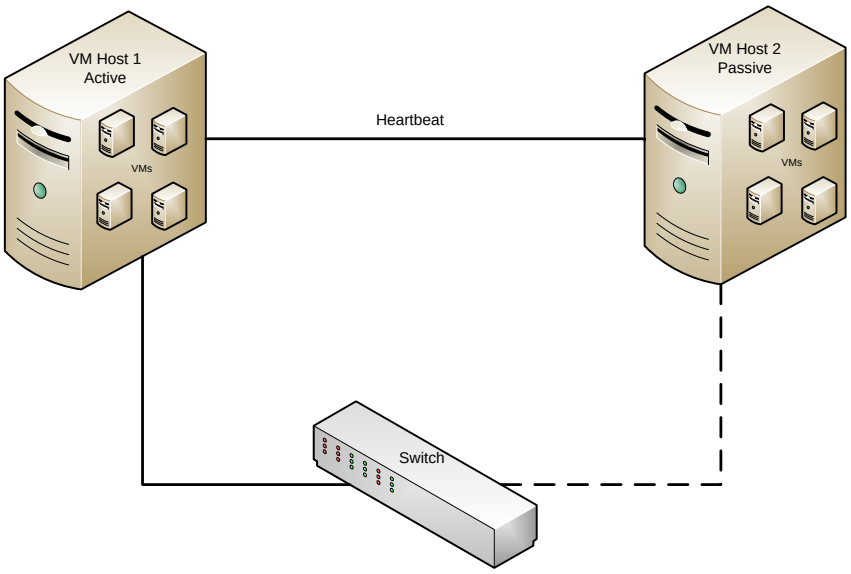

Figure 31: Seamless failover and failback of Communication Servers

One physical server acts as a host for virtual machines (VMs), using the virtualization capabilities of modern CPUs, and one physical backup/mirror server for seamless failover in case of a malfunction in the first server. The architecture will be based on high availability server cluster architecture.

All communication gateway modules presented above are applications running on separate VMs on the host server.

The server specifications are as follows:

- CPU: Intel Core i7
- RAM: 8 to 16 GB DDR3
- HDD: 2 x 1TB in RAID 1 (Mirroring)
- Ethernet: 2 x 10/100/1000 Mbps
- Wi-Fi: 802.11a/b/g/n
- Serial: 2 x RS-232/422/485 ports
- USB: 4 x USB 3.0/2.0 ports, 2 x USB 2.0/1.1 ports
- OS: Ubuntu Server 14.04 LTS

As far as the various supported interfaces are concerned (Ethernet: 2 x 10/100/1000 Mbps, Wi-Fi: 802.11a/b/g/n Serial: 2 x RS-232/422/485 ports), the Ethernet and Wi-Fi interfaces are obligatory in order for the server to be connected redundantly to the pilot site's LAN. Furthermore, the serial port is the interface used by components like the Fire detection system.

3.3.8 Crowd Behaviour

Crowd behavior detection and recognition in crisis situations is the main focus of the current research work. Its foundation and further development are described in this second version of this project deliverable. The work encounters multiple technical challenges which are addressed and refined in phases throughout the duration of this project. The main target is to achieve acceptable rates of correct detection and classification of crowd motion and behavior respectively. However, the performance of the developed intelligent algorithms, put in place, for the detection of crowd motion and behavior greatly depends on the quality of observation data and ground truth on crowd, acquired during the project. In addition, it is of great importance to acquire good contextual and spatial details of the pilot confined environments from which crowds are evacuated. This concerns crowd evacuation from 1) Stadium in Bilbao, 2) Athens Airport, 3) Cruise ship and 4) Underground train station in Bilbao.

The technical methodologies, using vision observation data, are described in this first version of the deliverable. They concern the following:

1. Detection of crowd physical motions at macro-, meso- and micro-scales using a number of approaches including ones based on statistical thermodynamic principles.
2. Detection of behaviour at individual, group and overall crowd levels.
3. Use of hyper-spectral and thermal image analysis for further detection of crowd motions and behaviour through the exploitation of wider multiple spectral bands

Deliverable D3.4 takes on board Tasks 3.1, 3.2, 3.3 and 3.4 research activities for the development of highly advanced intelligent algorithms on the detection of crowd motion and behaviour in confined environments. These shall be capitalized from the respective research activities on the above mentioned tasks to potentially propel the behaviour algorithms towards understanding seed behaviour propagation within crowds. These shall be fused with other sources of information potentially. For example, those from smart sensing in WP7 to advance the deep learning and situation awareness in crisis situations that is currently performed under WP3.

3.3.9 Strategic evacuation

Evacuation strategies are important for any building or venue. In general, they currently exist in a static form, with a fixed plan that is complemented by training and 'table top exercises' where different scenarios can be played out hypothetically. During crises themselves, the strategy tends to change for any particular incident; the situation is often unique and has not arisen during training exercises. Indeed, there are an infinite number of possibilities for such incidents, when one considers the variables – type of threat, location of threat, combinations of threats, current population of building, behaviour of crowds etc.

The eVACUATE system aims to sustain an active evacuation route for the actual situation that has unfolded during a crisis. Further to this, it aims to provide an optimum evacuation strategy for any given situation, in respect of the information available. This document describes the analysis undertaken in work package 4 regarding current evacuation strategies, how eVACUATE optimises strategy and the analysis of methods for providing optimum evacuation routes using predictive software.

The presence of the eVACUATE system itself provides additional information during crisis situations, and includes predictive crowd models to forecast the likely outcome of a given evacuation process before it has taken place. This enhances the current static evacuation strategies, making them much more dynamic and able to respond to a unique situation where decisions can be made based on much greater situational awareness.

3.3.10 Smart Space Agents

The agent's role is completed at this stage if we are willing to accept the agent simply as an intermediary between two black boxes; but, to see the real situation in more detail, we will understand that the SIB then communicates with the highly-qualified human user through the Common Operational Picture (COP) developed in WP4 Advanced Strategic Spatial Evacuation and the system will also contact with WP3 Crowd Behaviour Detection & Recognition in Crisis Situations - but the agents will not contact with these two last WPs by themselves, meaning that we do not have to deal directly with these WP's in the current document.

The Agents communicate with the SIB by means of ontologies. At the time of writing this deliverable the eVACUATE ontologies (developed within "T7.4 Ontologies") are not fully defined. This means that some of the information presented in this deliverable may be completed or modified in the future (especially data models) although the core functionality of the Agents will not be affected. In this sense, the deliverable "D7.6 Smart Space Information Model: Data Models and ontologies", to be delivered on Month 30, can be considered as an extension for this deliverable.

The architecture of the Agent Framework has been designed in layers and components in order to allow modifications to the ontologies without affecting the overall functionality of the applications. Actually, the ontology concept has been hidden from the application layer. Specifically, the modification of the ontologies may affect the "Ontologizer Library", "Configuration Manager Library" and "eVACUATE Data Model Library". The implications will be explained in the corresponding sections within this deliverable.

The short definition of agent, previously approved in the project, is the following one:

An agent is a piece of server-based software that centralizes all the information from a given kind of device.

For every specific kind of device, only one agent will be implemented. Each instance of the device will communicate with the agent, and the agent will know that each instance is different - for instance, two exit signs in opposite sides of the room are controlled by the same agent, but that agent may want to turn on only one of the exit signs if the other one leads to an unsafe area.

Not all the technologies within eVACUATE will use agents to communicate with the SIB. As we have already seen, the highly-qualified controller deciding whether to launch an alert does not have an agent - that person communicates with the SIB through the COP, as described in section "2.2 Relationship with other modules in D7.4". There are other exceptions: Video streaming and social networks follow a different communication path due to a number of reasons. In other cases, a smart phone application used to collect information about its immediate environment may communicate with the SIB using an agent, although a software application is only arguably a device.

Some agents may have specific functionalities because of the nature of the device they are related to.

3.3.11 SOFIA

The SOFIA (Smart Object for Intelligent Applications) platform is a platform that focuses its efforts in this kind of architectures based on smart objects cooperating together into a smart space. SOFIA provides an interoperability platform, where devices are able to discover and to connect to a smart space in order to use and provide cooperating services. Connection at the lower level is managed by a middleware layer that uses legacy connectivity and communication protocols. At a higher level of abstraction, SOFIA uses semantic models as mechanism of knowledge exchange between devices. Each smart space defines its own ontology that describes all relevant elements into its domain. So, each device connected to a smart space can use the ontology concepts that need to manage in its application logic. Thus, all devices connected share the same vision of the domain where they are deployed.

The smart space vision of SOFIA is based on a client-server architecture:

• The server side is a knowledge database that contains the most updated status of all devices deployed in the smart space. This part is called SIB (Semantic Information Broker). It uses a RDF triples to store the information of status as semantic elements.

• The client side are one or more devices, these devices add themselves to a smart space by connecting to the SIB of that smart space. These devices are called KP (Knowledge Processor).

The communication between the KPs and the SIB is managed by a message protocol defined in the SOFIA project. This message protocol is based on XML and is called SSAP. The SSAP protocol provides to the KPs (clients) methods to connect/disconnect to the SIB, insert/update/remove its own information, query information inserted by other KPs, and to subscribe to specific kinds of information to be automatically notified by the SIB when any event occurs.

The smart spaces proposed in eVACUATE, can be easily instantiated as SOFIA smart spaces. All devices such as sensors, actuators, monitoring elements will be developed as KPs that will cooperate together through a SIB, creating the smart space. SOFIA provides the middleware libraries to the KPs. As mentioned above, in the lower layer, the communication protocol uses legacy transport protocols such as TCP/IP, Bluetooth. But SOFIA is enough modular to support new transport protocols, only by the adding the necessary connectors, without modifying the middleware libraries. In the top level of the smart space, SOFIA only requires the development of an ontology for each specific scenario (smart space). This ontology will describe all the relevant elements into the domain of the scenario, and will be shared by all devices (KPs) connected to the same smart space, this way all devices will have the same vision of the elements into the smart space. Each KP will instantiate its own elements using their description in the ontology, and will publish/update them in the SIB using a RDF triples mechanism. This mechanism is language independent, allowing the interoperability of KPs developed in different programming languages. In addition, the KP can query/subscribe to the SIB, receiving the current status of any other elements published in the SIB by other KPs. In summary, using the SOFIA smart space platform in the scenarios proposed in eVACUATE only requires the definition of the ontology for each scenario (stadium ontology, ship ontology, etc.) and the development of the transport connectors not provided currently by the SOFIA platform and used in those scenarios (RFID, PLC, etc).

3.3.12 DDS

DDS is networking middleware that simplifies complex network programming. It implements a publish/subscribe model for sending and receiving data, events, and commands among the nodes. Nodes that produce information (publishers) create "topics" (e.g., temperature, location, pressure) and publish "samples". DDS delivers the samples to subscribers that declare an interest in that topic.

DDS handles transfer chores: message addressing, data marshalling and demarshalling (so subscribers can be on different platforms from the publisher), delivery, flow control, retries, etc. Any node can be a publisher, subscriber, or both simultaneously.

The DDS publish-subscribe model virtually eliminates complex network programming for distributed applications.

DDS supports mechanisms that go beyond the basic publish-subscribe model. The key benefit is that applications that use DDS for their communications are decoupled. Little design time need be spent on handling their mutual interactions. In particular, the applications never need information about the other participating applications, including their existence or locations. DDS transparently handles message delivery without requiring intervention from the user applications, including:

- Determining who should receive the messages
- Where recipients are located
- What happens if messages cannot be delivered

DDS allows the user to specify Quality of Service (QoS) parameters to configure discovery and behavior mechanisms up-front. By exchanging messages anonymously, DDS simplifies distributed applications and encourages modular, well-structured programs.

DDS also automatically handles hot-swapping redundant publishers if the primary fails. Subscribers always get the sample with the highest priority whose data is still valid (that is, whose publisher-specified validity period has not expired). It automatically switches back to the primary when it recovers, too.

3.4 Deployment

All EOC modules are deployed on tomcat [19]and we use Mysql[20] database. For the big data we use the ambary cloud manager [21]to manage the deployment of Hadoop and Spark [22]and we store all data in Hadoop as unstructured data.

The EOC Portal is deployed in Glassfish [23]and the Big Data portal is based on kibana with custom Dashboards and search filters.

The EOC Cloud is based on a master node and two client nodes all manager by Ambari[24].

[19] http://tomcat.apache.org/
[20] https://www.mysql.com/
[21] http://hortonworks.com/
[22] http://hortonworks.com/
[23] https://glassfish.java.net/
[24] http://hortonworks.com/

4. Security Framework

The specification of the eVACUATE security framework has been designed and implemented so that it complies with the following security requirements (The security framework follows the evacuate security architecture and security modules specified in D6.3):

- **Authentication**
  Access to system services should be restricted to authenticated users only.
- **Authorization**
  Access to system services should be authorized according to the role of the user.
- **Transparent Transactions**
  Key system transactions involving access to sensitive data or system functions should be recorded. Audit Trails associating actions and actors should provide tracking and accountability of key user transactions.
- **Secure Transactions**
  Online Transactions involving sensitive data must be secure. This involves preventing session hijacking, message and network level encryption.
- **Data Confidentiality**
  The content of transactions between a client and eVACUATE server cannot be intercepted by a third malicious party (man in the middle attack). The same applies to data stored in the system's persistent storage.
- **Data Integrity**
  The content of transactions between a client and eVACUATE server cannot be tampered by a third malicious party (man in the middle attack). The same applies to data stored in the system's persistent storage.
- **Non-Repudiation**
  According to this requirement, a user/actor cannot deny any of his transactions/actions within the eVACUATE system.

The eVACUATE Security platform consists of the following key components:

- LDAP Framework, using OpenLDAP
- PKI Framework
- JAAS/JDK 1.5+ framework provided by the Application Server

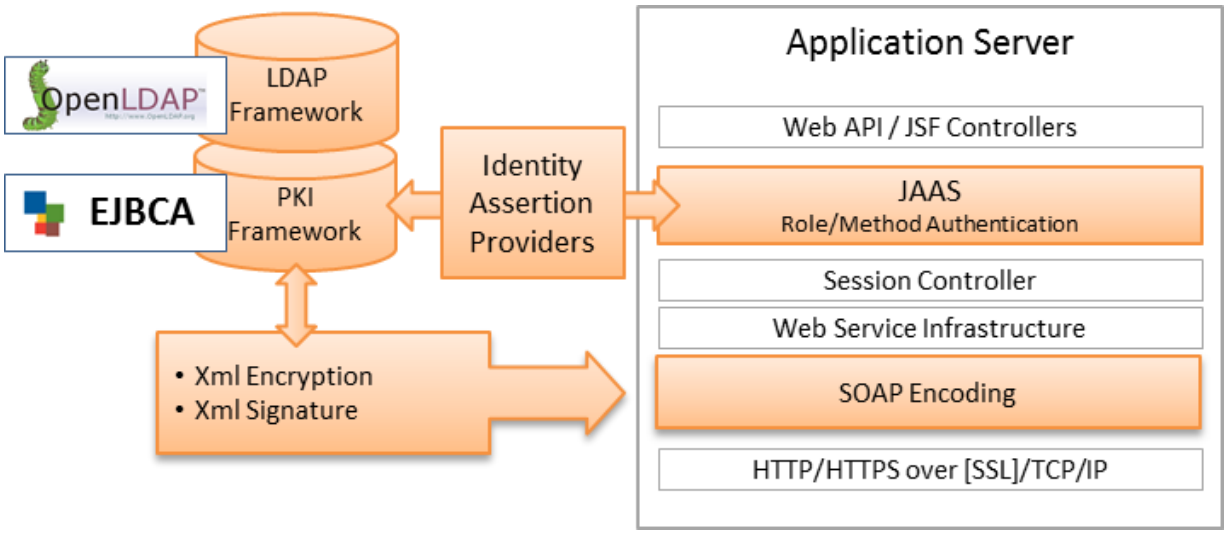

Figure 32: Security Application Stack

Table 14 provides an overview of the technologies used and their mapping towards fulfilling eVACUATE security requirements.

	Authentication	Authorization	Secure Transactions	Transparent Transactions	Data Confidentiality and Integrity	Non-repudiation
JAAS	✓	✓				✓
HTTPS			✓		✓	✓
XML Encoding			✓		✓	✓
XML Signature			✓		✓	✓
DB Data Encryption					✓	
EJBCA-PKI	✓		✓		✓	✓
Audit trails				✓		✓

Table 14: Mapping of technology solutions used in Evacuate subsystem with security requirements

In the following of this subsection, we will explain how the eVACUATE subsystem addresses the security requirements.

5. Conformance with security requirements

5.1 Authentication & Authorization

The system builds upon the framework provided by the Java Authentication and Authorization Service (JAAS)[25] in order to ensure that access to any system service and applications is limited only to authenticated users.

Authentication in the eVACUATE security subsystem is provided in two ways:

- User credential-based authentication (user name/password)
- Security certificate-based authentication

The first method is used for authenticating users accessing the Web Portal application while the latter is used for authenticating access via the mobile application (mobile application-backend communication). The use of each method is explained in the next subsection.

In order to meet this requirement the eVACUATE Security subsystem builds on two core modules:

- OpenLDAP[26], which is an open source implementation of the Lightweight Directory Access Protocol and
- EJBCA (Enterprise Java Bean Certificate Authority) which is an open source software public key infrastructure (PKI) certificate authority software package.

These two modules integrate easily through configuration in order to be able to exchange information and allow EJBCA to use the LDAP module to manage and store user/key associations operating as a “keystore”.

JAAS then uses the concept of **Identity Assertion Providers,** which are configuration associations which allow to route processing to the appropriate module depending of the kind of credential used in an incoming message ie X.509 PKI certificate or username/password pair.

- An Open LDAP Authentication provider and
- An EJBCA/X.509 LDAP Authentication provider

Additional tables in the backend persistent storage allow the addition of domain specific extensions with metadata to be used various processing levels.

Table “Users” contains among other information the login credentials of a user. I relates with a [1:n] relationship to the Tables “Roles” and “PKI Keys”.

The “PKI Keys” table contains PKI keys which are used in the Security certificate-based authentication scenario. It is applicable only for “Patient” users.

[25] http://docs.oracle.com/javase/7/docs/technotes/guides/security/jaas/JAASRefGuide.html
[26] http://www.openldap.org/

These keys are issued from the Administration Portal and once made available to the patient user, they are installed on the smartphone device and subsequently used for any transactions of the Smartphone app with the eVACUATE Backend.

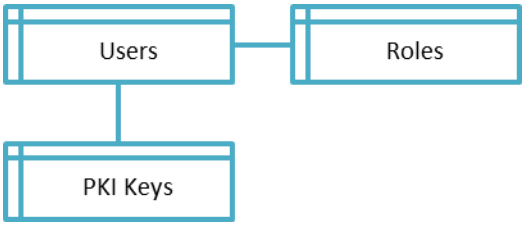

Figure 33: Evacuate Database Entities extending Authentication/Authorization information

5.1.1 Credential-based authentication

Credential-based authentication is used for authentication through the web portal application.

When a user accesses the web portal application; the login sequence is secured by HTTPS/SSL transactions. Figure 36 presents the sequence diagram of a successful login process and Figure 37 the sequence diagram of a non- successful login process. In the case of a successful login, the user session is created as shown in Figure 34. The created user session is associated with the role(s) of the user role in order to provide role-based authorization, which we be discussed in section 5.5 Roles

Figure 34: Flow chart of the session creation

Access to the methods of the accessed service(s) within the session is regulated by JAAS making use of the role information stored in the session and the role-method association, as shown in Figure 35.

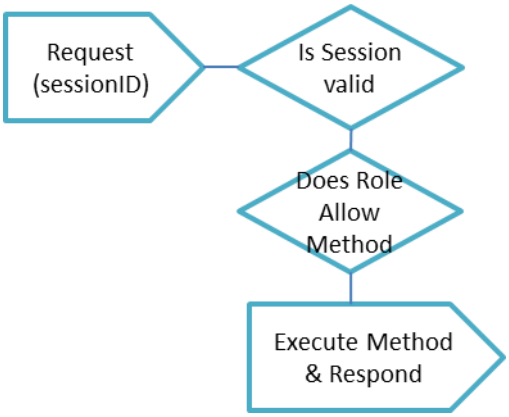

Figure 35: Flow chart service method invocation within a session

5.1.2 Certificate-based authentication

Certificate-based authentication is used for system access through the smartphone device via the smartphone service interface (from users with 'Patient' role) and the Inter-System Service Interface (from users/external systems with 'System' role).

As an integral part of establishing the right to access this interface, it is necessary to install on the smartphone device or the external system module, a unique certificate which is issued by an administrator through the Administrator Portal. Service invocation through this mode of operation is by definition session-less, at least in terms of application server functionality. Therefore each web method invocation contains authorization information. These will be asserted in terms of authentication through JAAS and the X.509 IAP from EJBCA/LDAP and then, in terms of authorization using the configured Method/Role mapping.

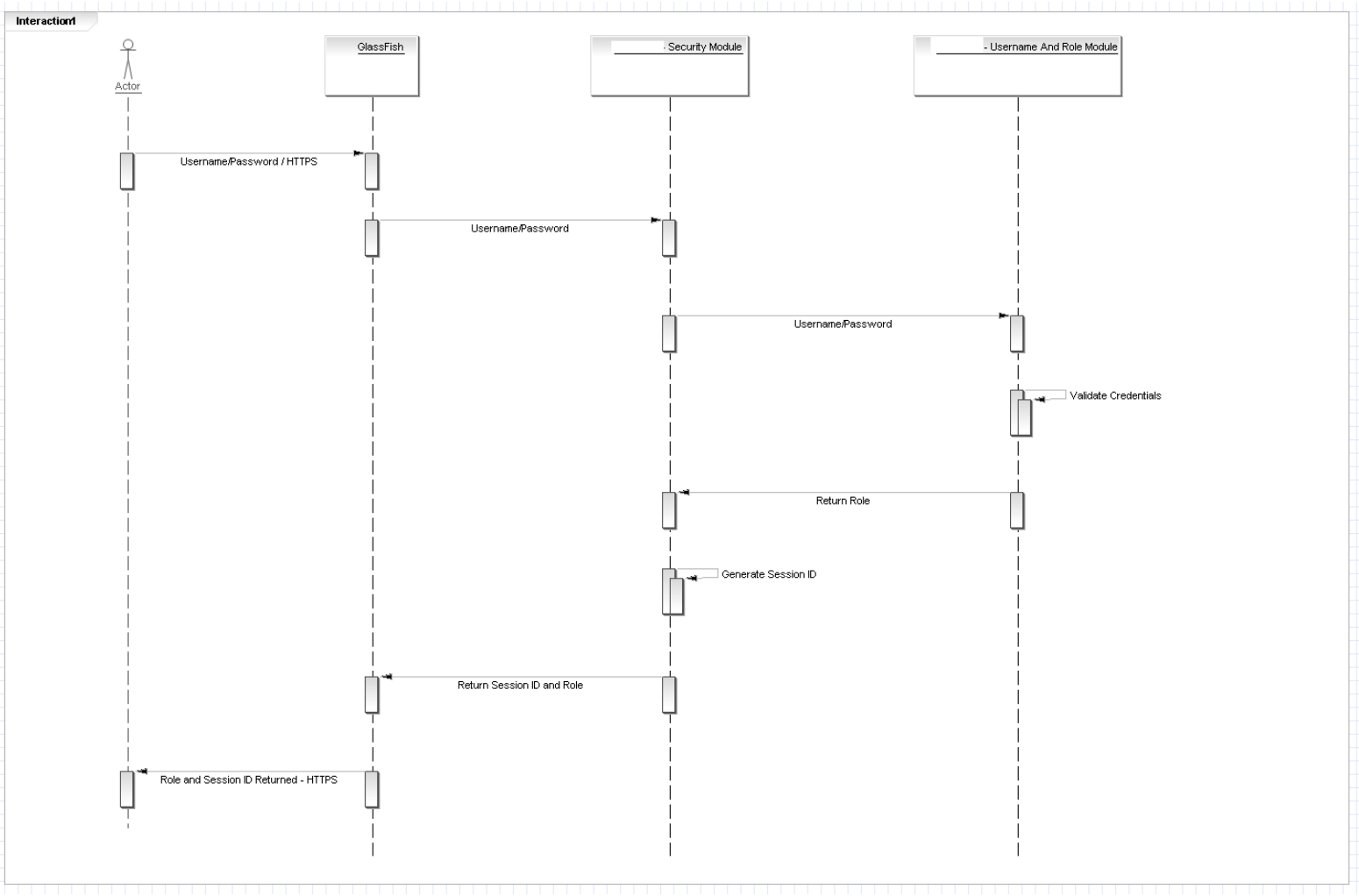

Figure 36: Successful user authentication with credentials

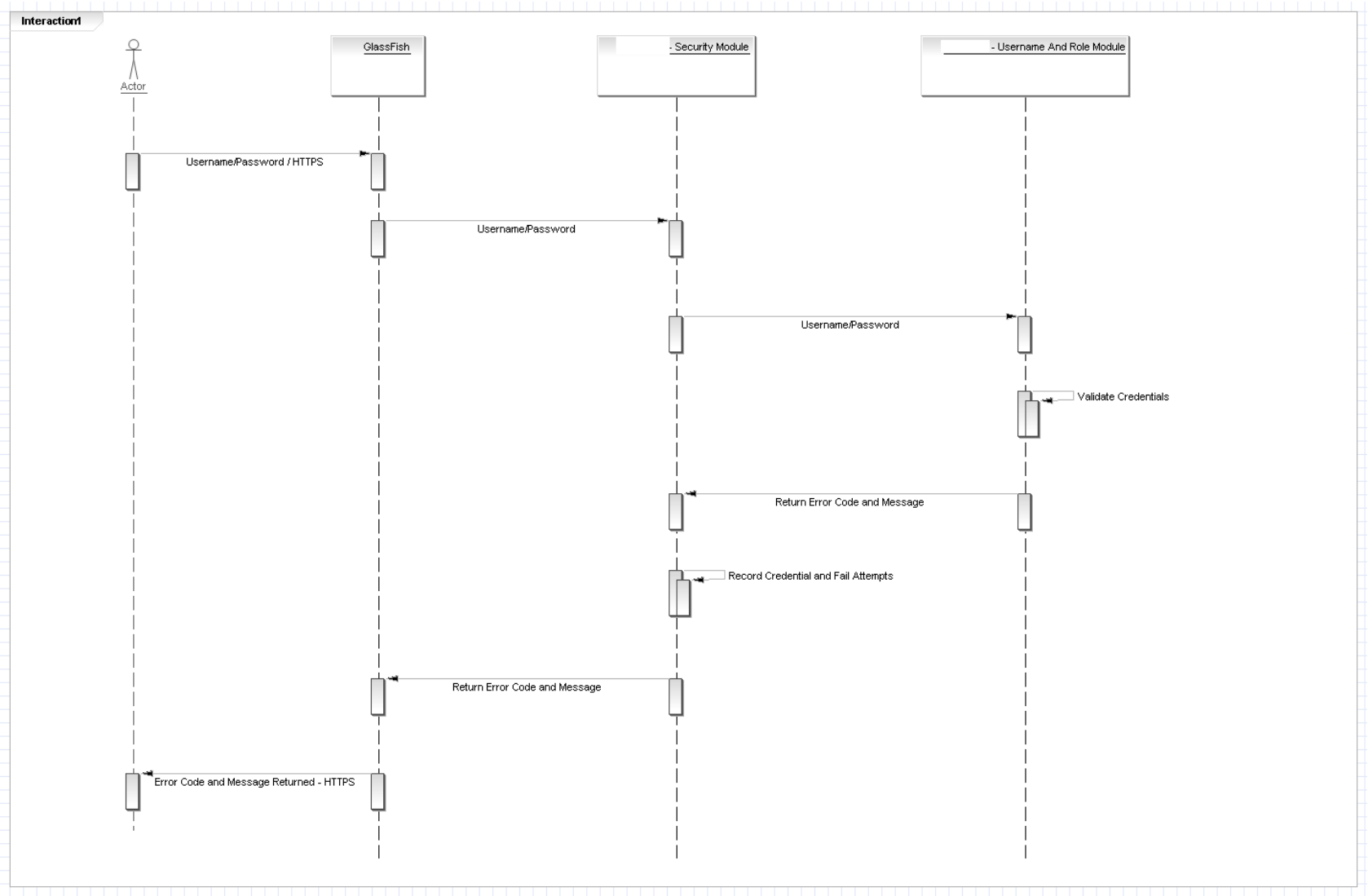

Figure 37: Failed user authentication with credentials

5.2 Data Confidentiality & Integrity

To address this requirement, all transactions through the service interfaces are secured through the use of XML Encryption and XML Signature techniques.

XML Encryption [27]is a method for guaranteeing the confidentiality of the data. The XML Encryption specification being developed by the W3C's XML Encryption Working Group provides a framework and procedures for encrypting XML messages to ensure that only the intended recipient can read them. Encryption can be applied at document or element level and use different encryption keys, a feature that can be exploited if in a feature system design message processing is dispatched to multiple processing entities each with different access rights.

In the current implementation we use encryption at document (message) level which means that all the message body is encrypted.

A scheme of mutual authentication is used which means that each communicating entity manages a pair of Private and Public keys. From the backend perspective, the system owns a Private Key which it uses to decipher incoming messages and a user specific Public Key which uses for ciphering outgoing messages for the specific user/server interaction. From the user perspective the respective software module (smartphone application or 3rd party software) owns a private key, it received from a eVACUATE administrator and a public key of the eVACUATE backend.

XML Signatures [28]provide integrity, message authentication, and/or signer authentication. The concept is similar to the one described above for the Xml Encryption feature. Mutual Public/Private key scheme is employed to create unique hash signatures, which are embedded within the document (when signature applies on the whole document) or specific elements (when applies on specific xml elements).

In the current implementation Xml Signature is applied at document level. Both features use Java Xml Encryption API.

5.3 Transparent Transactions

All transactions create comprehensive audit trails providing accountability for selected actions and access to data entities, services and methods. The backend system provides a custom implementation of audit trailing which records a vector of user id, entity code, operation code, event date, and change data.

5.3.1 Data Confidentiality and Integrity

Data encryption at transmission level allows the system to prevent any unauthorized access and tampering, interception of sensitive data elements. The use of state-of-the art security deployment schemes such as Digital Signature, XML Encryption on a PKI infrastructure, combined with proven Authentication/Authorization mechanisms as

[27] https://en.wikipedia.org/wiki/XML_Encryption
[28] https://en.wikipedia.org/wiki/XML_Signature

described in the previous paragraphs fortifies the system against a wide range of threats such as session hijack, man-in-the-middle attack etc.

Data encryption at storage level is a new feature introduced in the latest version of the backend system. This allows a section of the persistent storage (DB) to be encrypted using system scope encryption key. This means that only software modules can have full access on sensitive private information. This prevents e.g. a system administrator to be able to browse sensitive information through queries with reporting tools. In this way all access to the systems physical data is regulated through the authorization/ authentication mechanism of the eVACUATE backend, and direct access to the database, in effect allows access only to anonymised information, which is still suitable for statistical analysis reporting etc.

5.4 Non-Repudiation

Non-Repudiation refers to the need that a trustful system should make sure that an actor of a specific action cannot deny the fact of this action.

This can be broken down to the following:

- The system should be able to provide a level of authentication that can be asserted to be genuine with high assurance.

- The system should be able to prove the integrity and origin of data.

This requirement then is met by the effectiveness of all the measures described in the above sections.

Message/Document Integrity is proven by PKI managed digital signatures.

Additional Audit Trails introduce accountability for operations performed on the system and allow administrators to identify suspicious patterns, breaches, and malicious practices allowing the deployment of anti-measures in other security levels (firewalls, etc.)

5.5 Roles

The description of the roles that we defined in the eVACUATE security subsystem is provided in Table 15. Table 16 provides an overview of the access rights associated to each defined role. Note that role administration is provided through the "Administration Portal" which was presented in section 2.

Role	Role Description
System	The 'System' role has full access to all methods. This role is mainly used for communication with external systems. The system user is not exported to the outside world, and can only access the web services from the same domain. The list of methods accessed by this role is provided in Table 6.

Administrator	The 'Administrator' role has access to the Administration portal with then main task of managing users and their profiles.
Crisis Manager	The 'Crisis Manager' role has the right to manage a crisis, he is the EOC Operator and can perform all available operations.

Table 15: Roles in the Evacuate security subsystem

	Crisis Manager	System	Administrator
Manage Users Security Keys		✓	✓
Manage FRs	✓	✓	
Manage Devices	✓	✓	
Manage Smartphones	✓	✓	

Table 16: Access Rights per User Role

6. Network

6.1 Network Topology

eVACUATE network is based on a network switch(s) for connecting all cable connected devices, a router for routing all information faster and secure across all devices and a wireless access point for wireless access to the eVACUATE network. Below we present an overview of the eVACUATE network, more details are presented on the network deployment section below.

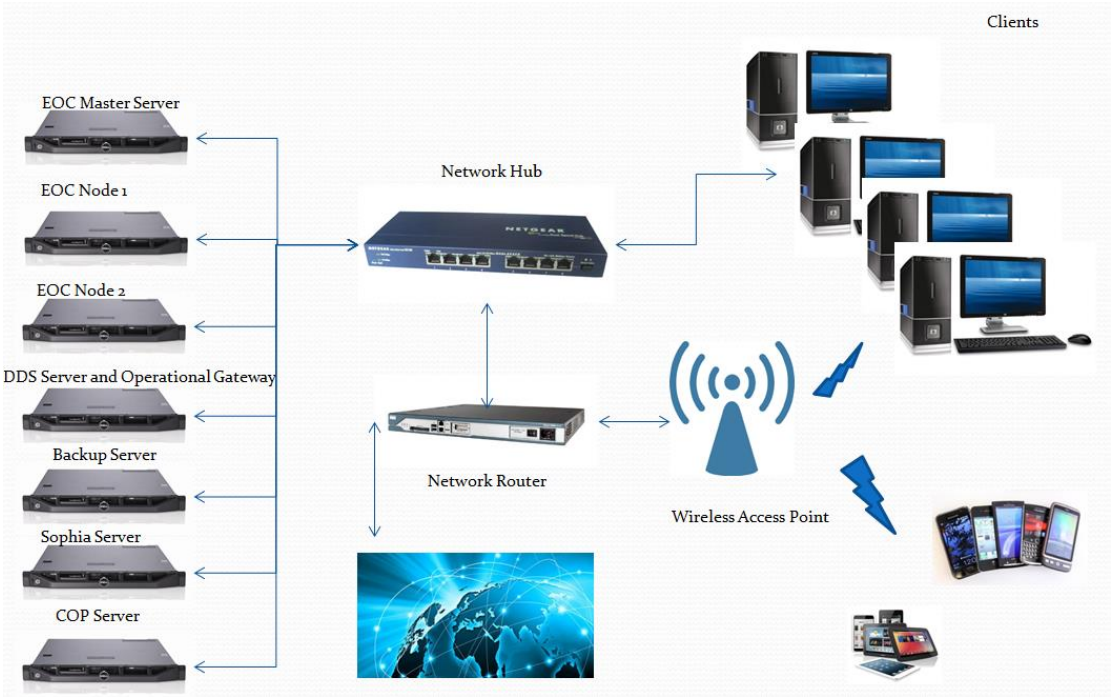

Figure 38: EOC Network

6.1.1 Communications - Internal Network Architecture

The internal working between these technologies is made available at the EOC thanks to a wired Ethernet-like LAN that, moreover, provides network connectivity to the EOC's internal IT systems. The high-level picture of the EOC's architecture is reported in the following figure

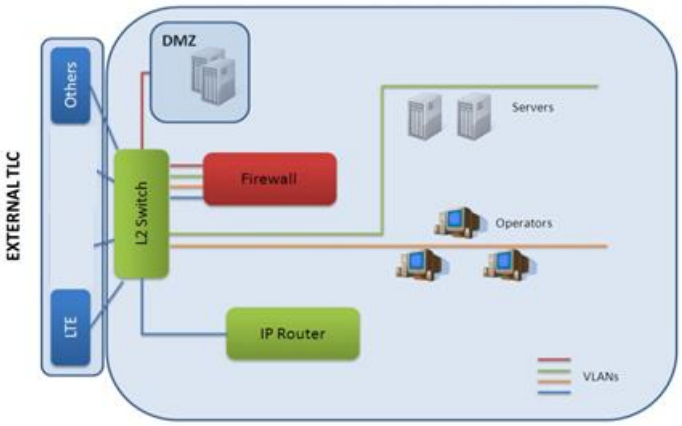

Figure 39 - High level view of the internal EOC architecture

Since Ethernet technology is adopted to provide a flexible and interoperable interconnection, it becomes vital that every wireless network interface used at the EOC is supplied with an Ethernet interface. This, in turn, will be used to interconnect each interface with a layer-2 bridge (or a switch), internal to the EOC. In addition to the Ethernet-based LAN, the EOC's internal ICT equipment is composed by an IP router, the security subsystem and the set of computer machines required to host or support the applications.

6.1.2 Local Area Network

As for the interconnection part of the LAN, namely the bridge/switch hardware, there is the need to have a gigabit ethernet compliant device. Regarding the number of its ethernet ports, this depends on:

1. the actual number of the network interfaces used at the EOC to support communication to the rest of the eVACUATE network (e.g., legacy devices,external devices,etc);

2. the actual number of computer systems used to implement the security subsystem;

3. the actual number of computer machines deployed at the EOC, each one requiring (at least) one Ethernet port.

Based on the design proposed in previous documents [network design, cop, smart spaces], the total number of ethernet ports required for a real-world implementation of the EOC can be easily satisfied with commercial off-the-shelf products (e.g., CISCO devices or dell routers/switches).

6.1.3 IP Routing

An IP router is required at the EOC to perform the routing of IP datagrams across the different network segments composing the eVACUATE network (e.g., to route and forward packets from the lan network, to the IEEE 802.16 one).

A number of alternatives are available regarding the actual implementation of the IP routing function. The simplest approach will require the implementation of a commercial off-the-shelf router, but other solutions are possible, too (e.g., the use of a software router to allow the network administrator to implement specific traffic shaping policies).

The IP router should provide several features, such as:

1. VLAN/IEEE 802.1Q tagging support to manage different VLANs;
2. Static/dynamic routing support, depending on how the actual Evacuate network has been implemented;
3. IPv4/IPv6 support.

6.1.4 Network Deployment

For the connection of the different sensors, actuators and physical servers hosting the communication gateway modules, the operator/PPDR gateway module and the smart space servers two 24-port managed Gigabit Ethernet Switches (10/100/1000) will be employed. Following a layered network architecture one L2 switch will be used as an access layer switch to connect the Communication Gateway Modules and the different sensing and actuating device interfaces. A layer 3 switch will act as an aggregation and core layer switch/router that will interconnect and route the traffic among the access layer devices and the core layer elements such as the EOC, the COP and SOFIA and their respective servers.

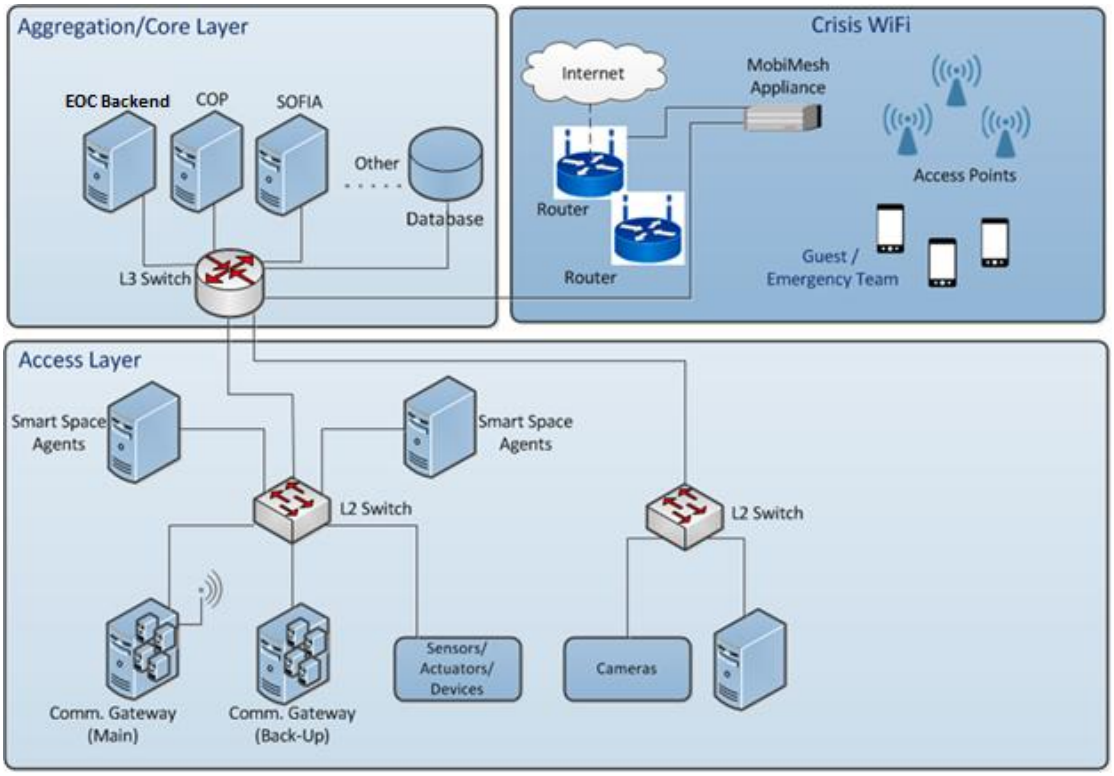

Figure 40: Deployment view

7. EOC Beta Version Access details

To access the eVACUATE EOC portal you will need access to the following ip address 193.92.114.115 and 193.92.114.117 and ports 8080 and 5601 open.

- The EOC Portal can be accessed from the following URL: http://193.92.114.115:8080/EOCUI/

- The Big data framework can be accessed from the following URL: http://193.92.114.117:5601/

8. Conclusion and Future Work

Currently the EOC is a complete subsystem consisting of a back-end and a front-end infrastructure. All system are deployed in EOC infrastructure and EOC can communicate with the other modules using SOFIA messages, Restful API or DDS networking protocol.

The Security is only supporting Username/Password currently. As a future work we need to switch to https calls, integrate with a certificate authority such us EJBCA[29] and implement encryption.

Communication with the eVAMAPP (Mobile application) will be handled by EOC, a dedicated set of components will be developed to manage Alerts/TETRA Messaging and User handling.

The EOC portal is currently a beta version as it needs further stylistic improvements in its interface, addition of some extra functionalities for Management of the Public Announcement system and implementation of a Crisis Context management where the Operator can modify the Crisis Attributes. All these additions will be reported in the updated (v.2.0) version that is scheduled to be submitted at the end of M33.

[29] https://www.ejbca.org/

ANNEX A - Hardware Inventory for EOC Implementation

Equipment	Description	Photo
Dell R-210 Server Single CPU Xeon 32 GB Memory 2 TB Hard disk space	Part of EOC Cloud (Master node)	
Dell R-210 Server Single CPU Xeon 32 GB Memory 2 TB Hard disk space	Part of EOC Cloud (node 1)	
Dell R-210 Server Single CPU Xeon 32 GB Memory 2 TB Hard disk space	Part of EOC Cloud (node 2)	
SWITCH PLANET 16PORT GIGABIT	Part of EOC Network infastructure	
MikroTik RB2011iL-RM 5xEthernet 5xGigabit Mainboard : 600MHz, 64MB PoE out L4	Part of EOC Network infastructure	
Flight Case 19" rack 3x Large catches 6x handles Medium tour label dish Casters	EOC Rack total of 19U and table.	

Table 17: EOC's hardware Inventory

ANNEX B References

[1] IDABC - Content Interoperability Strategy, Working paper, September 2005

[2] IDABC - European Interoperability Framework For Pan-European Government Services (version 1.0), EC 2004.

[3] The Web Services-Interoperability Organization (WS-I), Basic Security Profile version 1.0, Working Group Draft (17-08-2006) available at http://www.wsi.org/Profiles/BasicSecurityProfile-1.0.html

[4] The Web Services-Interoperability Organization (WS-I), Basic Profile version 1.1, Final material (10-04-2006) available at http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[5] Inspector IST project (IST-2000-26347) Deliverable D1.2 A data and process model of the architecture of the validation system.

[6] Bass L., Clements P., Kazman R. Software Architecture in Practice, Addison Wesley, April 2003.

[7] Bass L., Clements P., et al., Documenting Software Architectures, Addison Wesley,May 2003.