beAWARE

Enhancing decision support and management services in extreme weather climate events

700475

D7.4

Integrated beAWARE platform 1st version and Technical Evaluation Report

Dissemination level:	Public
Contractual date of delivery:	Month 18, 30 June 2018
Actual date of delivery:	Month 18, 30 June 2018
Workpackage:	WP7: System development, integration and evaluation
Task:	T7.3 - Overall Technical Testing of beAWARE platform
Type:	Demonstrator
Approval Status:	Final
Version:	**Error! Reference source not found.**
Number of pages:	42
Filename:	D7.4_beAWARE_1st_technical_evaluation_report_2018-06-30_v0.5

Abstract

This document describes the technical components and the technical evaluation of the first version of the integrated beAWARE platform implemented within Work Package 7 "System development, integration and evaluation". It provides an overview of the integration status of the different components the hosting infrastructure and the improvements that are performed since the operational version. The First Prototype is the updated version of the platform that integrates most of the services as a baseline version according to the

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 700475

requirements in Milestone 3.
The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Co-funded by the European Union

History

Version	Date	Reason	Revised by
V0.1	08/06/2018	Document initiation and assignments distribution	CERTH
V0.2	15/06/2018	Second version with skeleton information within	CERTH
V0.3	20/06/2018	Incorporate information	UPF, IOSB, CERTH, IBM
V0.4	28/06/2018	Internal Evaluation	CERTH
V0.5	29/06/2018	Internal Review	CERTH

Author list

Organisation	Name	Contact Information
CERTH	Ilias Koulalis	iliask@iti.gr
IOSB	Philipp Hertweck	philipp.hertweck@iosb.fraunhofer.de
ISOB	Tobias Hellmund	tobias.hellmund@iosb.fraunhofer.de
IBM	Benjamin Mandler	MANDLER@il.ibm.com
CERTH	Efstration Kontopoulos	skontopo@iti.gr
CERTH	Gerasimos Antzoulatos	gantzoulatos@iti.gr
CERTH	Stelios Andreadis	andreadisst@iti.gr
CERTH	Emmanouil Michail	michem@iti.gr
CERTH	Panagiotis Giannakeris	giannakeris@iti.gr

Executive Summary

The main goal of this document is to present the technical assessment of the 1st version of beAWARE platform through performance benchmarking experiments performed on system's components in order to verify its correct operation and interoperability of the different integrated modules to meet the requirements of the use cases for the end users. These requirements have been defined in D2.3"Pilot User Cases for the 1st prototype". This document upgrades the previous version D7.3"Integrated operational beAWARE platform".

The development status of each component is updated and summarised along with the integration approach and infrastructure that was used towards the implementation of the platform. The deliverable also demonstrates the operational use case scenarios that have successfully been implemented in order to test the interoperability of the different integrated modules.

Specifically three different use cases have been implemented, based on hypothetical storylines taking example from real crisis events:

1. Flood: A hypothetical storyline is assumed, taking example from the great flood of November 2010 when the Vicenza city centre, a very busy residential and commercial area, was submerged.
2. Fire: The territory of La Devesa de l´Albufera (Valencia) has been affected by some fires in the past. This use case refers to a simulated forest fire in the area.
3. Heatwave: The area of Thessaloniki was chosen since according to various studies conducted the city is ranked at the top of the heatwave scale.

Use cases depict the process flow through the platform based on its most likely use.

Overall, end-users' feedback has confirmed the initial approach and the relevance of the predefined requirements. Technically, all modules have been deployed at a basic level fulfilling the requirement as referred in D7.2 "System requirements and architecture". The general assessment from this deliverable is that the evaluation showed positive feedback and clearly pointed out the planning path towards the 2nd prototype.

Functional results and screenshots are presented in this deliverable.

Abbreviations and Acronyms

API Application Programming Interface

ASR Automatic Speech Recognition

CDR Central Data Repository

CI Continuous Integration

DTr dynamic texture recognition

DTstL Dynamic Texture spatio-temporal localization

GUI Graphical User Interface

IoT Internet of Things

JSON JavaScript Object Notation

K8s Kubernetes

KB – Knowledge Base

KBR – Knowledge Base Repository

KBS – Knowledge Base Service

M2M Machine-to-machine

MRG Multilingual Report Generator

MSB Message Bus

MTA Multilingual Text Analyzer

ObjD object detection

OGC The Open Geospatial Consortium

PaaS Platform as a Service

PSAP Public-safety answering point

SOA Service-oriented architecture

STL Spatio-Temporal localization

TLc traffic level classification

WP Work Package

Glossary

BlueMix – a public cloud hosting platform provided by IBM

Cloud Foundry – an open and extensible Platform as a Service offering

Consumable - Designing an easy to use product

Mongo DB – Scalable document oriented NoSQL data store

JSON - (JavaScript Object Notation) is a lightweight data-interchange format often used in web applications

Kafka – A messaging middleware mostly used to connect between different components

Kubernetes – Containers orchestration service: an open-source system for automating deployment, scaling, and managing containerized applications.

OWL - The Web Ontology Language is a family of knowledge representation languages for authoring ontologies

PSAP – Command and control centre to serve authorities, first responders, and citizens, mainly during crisis situations

Table of Contents

List of Figures

List of Tables

1 Introduction

In this document we provide the output of the efforts of creating the integration, testing and technical evaluation of the 1st version of beAWARE platform (P1), due M18 (June, 2018). For the fulfilment of the integration plan partners collaborated in many different occasions, especially for the definition of the necessary integration points between components, but also for a common approach regarding testing and evaluation with main goal to cover the functionality defined by the end-users. The integration of this 1st version is a continuation of the plan defined in D7.3 “Integrated operational beAWARE platform” and the Architectural Design, continued with the specification of the mechanisms described in D7.2 “System requirements and architecture”. The goal of revising the integration plan is to support the development by guiding the integration of the software components to the final version of the beAWARE platform, with the agreement on identified interfaces.

The document is structured in 5 sections: Section 2 provides a complete synthetic summary of the development status of each component. Section 3 details the process flow through the use cases as sequential steps followed for the interoperability assessment Section 4 presents the methodology followed for validation and some quantitative evaluation of the beAWARE key Factors. Finally, Section 5 summarizes the conclusions.

Overall, this report shows that the beAWARE project has achieved its objectives so far and fulfilled its key requirements. In addition, the beAWARE technical evaluation activities have allowed identifying and quantifying the strengths as well as some weaknesses of the platform and planning the path for the 2nd prototype.

1.1 Integration Overview and current status

In D7.1, a technological roadmap was established. This roadmap determines the main attributes and timelines for the development of the different components of the beAWARE platform and describes an iterative approach from the initial operation prototype towards to the final version of the beAWARE platform. The “walking skeleton” for this technical roadmap is presented below:

During the first year of the project the main goal was to provide an initial operational version of the Prototype which could allow a fast and easy integration framework including dummy services and services with limited functionality on them.

In this second step of implementation and according to the expected timeline for MS3, a first version of a functional Prototype is presented, followed by improvement of the functionalities

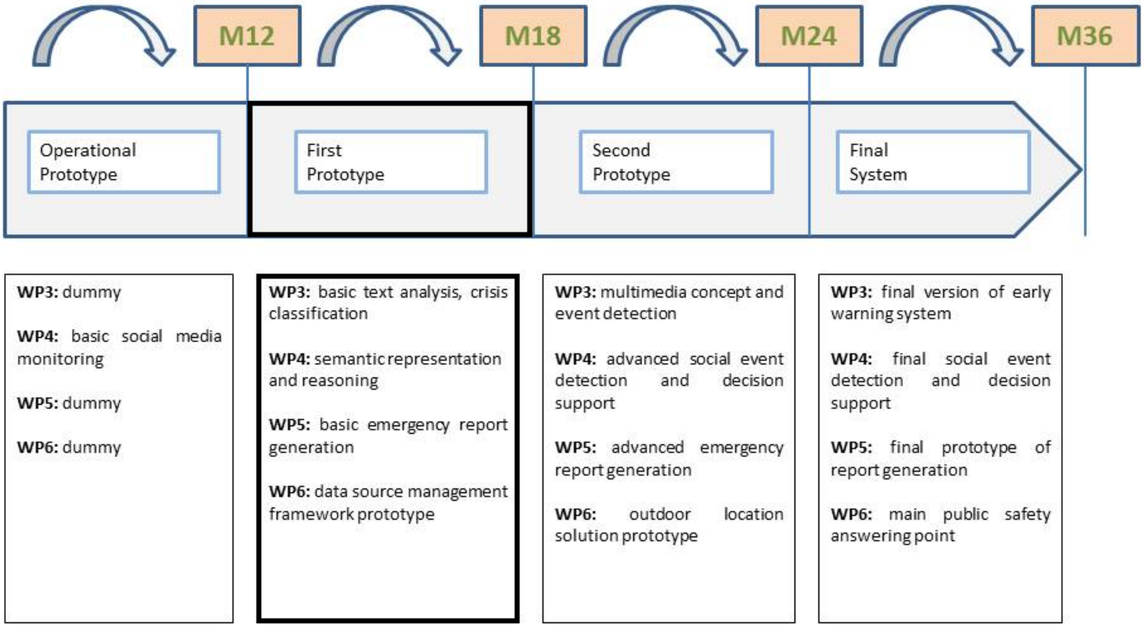


Figure 1: beAWARE walking skeleton

2 First Prototype Architecture

In this section is provided a high-level overview of the beAWARE platform architecture, explaining the components forming the system from a functional standpoint.

The 1st beAWARE prototype has successfully integrated all modules, as described in the DoW extending the functionalities of the Operational Prototype. Particularly it integrates (i) the basic module of text analysis (ii) the crisis classification, (iii) the semantic representation and reasoning module (iv) the emergency report generation modules (v) and the data source management framework prototype.

2.1 Global view

The global architecture for the beAWARE platform has been discussed at length in D7.1, D7.2 and D7.3. The architecture is roughly made up of the following layers:

1. **Ingestion layer**, containing mechanisms and channels through which data is brought into the platform;
2. **Internal services layer**, is comprised of a set of technical capabilities which are consumed by different system components. This layer includes services such as generic data repositories and communication services being used by the different components;
3. **Business layer**, containing the components that perform the actual platform-specific capabilities;
4. **External facing layer**, including the end-users' applications and PSAP (Public-safety answering point) modules, interacting with people and entities outside the platform (end-users of the platform).

An overview of all the components, in this first version of the platform, with a functional description and their dependencies is given in the following subsections. The complete set of functional dependencies between all the implemented components and their technical description can be consulted in the corresponded Deliverables.

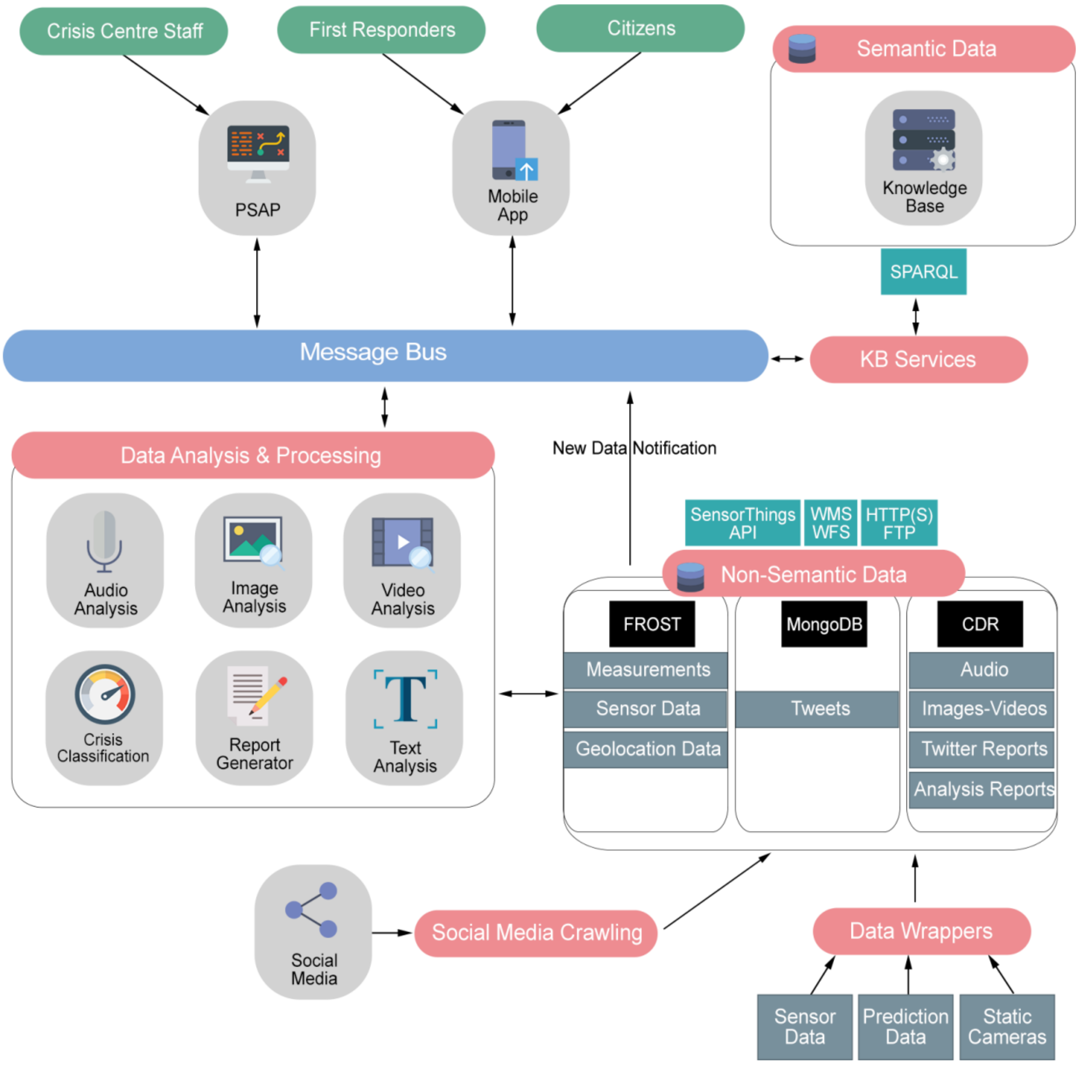


Figure 2: Architectural high-level view

2.2 Ingestion Layer

This layer serves as the input mechanism into the platform. Within this layer we can classify two modules: the Social Media Monitoring and the Monitoring machine sourcing information.

2.2.1 Social Media Monitoring module

The Social Media Analytics (SMA) component serves two different purposes. On one hand, the module crawls social media content exploiting Twitter's Streaming API[1] and after it classifies the posts as relevant or irrelevant; it forwards them to the Knowledge Base Service (KBS) to populate corresponding incident reports and to the Multilingual Text Analyzer (MTA) for further analysis. On the other hand, the module receives the analyzed texts from MTA, clusters the tweets based on time and location and sends a Twitter report per cluster

[1] https://dev.twitter.com/streaming/overview

to KBS. This second service is not delivered yet since retrieving location from tweets is a complex technique which is still under study and will be implemented under the MTA component on the next step of integration.

2.2.2 Monitoring machine sourcing information from IoT and M2M platforms module

Meteorological forecasts as well as hydrological and hydraulic data with forecast information about risk of flooding are important to the anticipation of extreme natural events such as flood, fire or heatwave. Furthermore, sensor data is crucial for monitoring the onset and progress of a crisis.

Crisis Classification module, currently implemented within this 1st prototype, collects all these data from heterogeneous resources ,through data wrappers that mediates, aiming to estimate the level of severity and the risk for an forthcoming or ongoing crisis. Thus, it utilises data from: the SensorThings Server API module which gathers and stores sensing information from physical sensors. Those sensors have been settled to weather stations in the region of interest. In order to acquire these data, CRCL module formulates and sends requests to SensorThings Server API.

The query's reply contains the AMICO forecasts regarding the Water Level over the river sections for a specific period of time.

Exploiting the Weather Forecast Models, like HIRLAM. Crisis Classification module formulates and poses requests to Finnish Meteorological Institute (FMI) data services.

The extracted data are in the Geography Markup Language (GML), which is the XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. The data present the parameters values (Air Temperature, Humidity and Wind Speed) to a specific location.

2.3 Business Layer

This layer encompasses the components that provide the actual platform specific capabilities.

2.3.1 Semantic modelling, integration and aggregation

To facilitate the usage and exchange of information, beAWARE deploys a semantic representation model, i.e. an ontology. For the creation of the ontology, we analysed important domain information regarding its semantics and formalized this information in OWL (Web Ontology Language), giving it well-defined meaning. The concepts that the ontology encompasses include: vulnerable objects, like people or assets, damages and related impacts, the description of incidents and associated climate conditions. The ontology also contains instantiated information. The model is fully described in beAWARE D4.2.

The ontology is hosted on a knowledge base (KB), and is implemented using the Fraunhofer IOSB developed system WebGenesis®, which is a content, community, and knowledge management system. It supports the user by providing several options and tools for managing, requesting, inserting, and visualizing data. Unlike many other content management systems, it is also able to work with ontologies. Since WebGenesis® separates content and layout it offers standardized templates for information categories and harmonizes the appearance.

On top of beAWARE's KB lays the Knowledge Base Service (KBS), a sophisticated middleware that handles storage, processing and retrieval of system data. Moreover, it acts as a communication proxy, receiving inputs from various components and informing other components. The interaction of the KBS with other system modules is summarized in Figure 3.

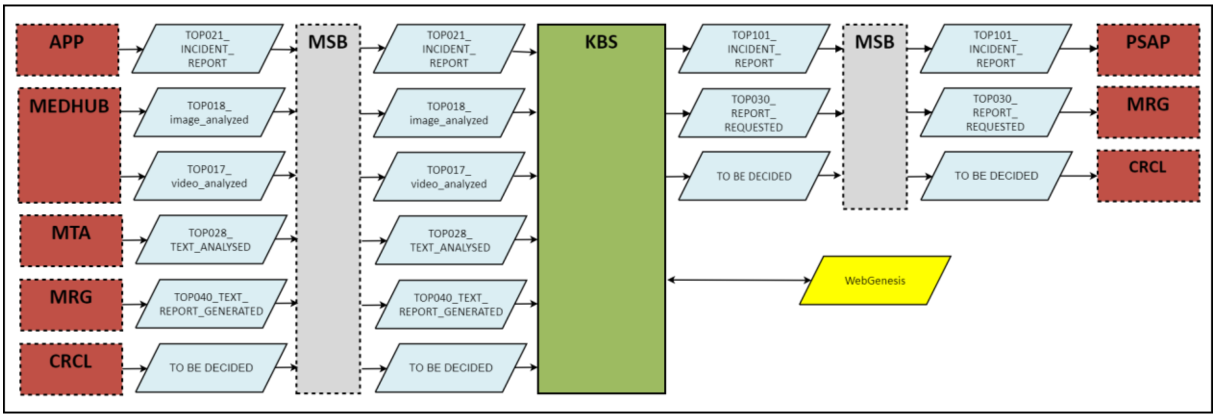


Figure 3 – KBS connectivity diagram

More specifically, the KBS engages in the following pipelines within the system's communication bus (MSB):

- The dedicated beAWARE mobile application (APP) publishes new incident reports and their updates as message in topic *TOP021_INCIDENT_REPORT*. The KBS parses the provided information (incident location, timestamp, attachments, etc.) and populates it to the appropriate ontology classes. Then, the KBS assembles and produces corresponding messages (*TOP101_INCIDENT_REPORT*) to notify the system's user interface (PSAP).
- New incident reports are also received by the social media components in topics *TOP001_SOCIAL_MEDIA_TEXT* and *TOP003_SOCIAL_MEDIA_REPORT*, and handled by the KBS in a similar way.
- Upon the announcement of new incident reports in the aforementioned topics, the two analysis modules MEDHUB and MTA (for image, video, audio and text analysis) are triggered and produce analysis results for reported attachments. These results are published in topics *TOP017_VIDEO_ANALYZED*, *TOP018_IMAGE_ANALYZED* and *TOP028_TEXT_ANALYSED*. The KBS monitors these topics and, when an analysis has been received, it associates the detected entities with the analysed media items and the original incident reports that had been previously created in the KB (see former flows).
- After each reception of analysis data, the KBS needs to update the PSAP with the newly discovered information. This also happens by producing update messages to topic *TOP101_INCIDENT_REPORT*. These messages contain two fields, *Title* and *Description*, which are considered crucial for the short, intuitive presentation of incidents at the user interface. Thus, the KBS firstly needs to consult the dedicated report-generation component (MRG). A message in topic *TOP030_REPORT_REQUESTED* is produced by the KBS, where all incidents and incident participants are enumerated, along with additional details, such as incident severity, timestamps, etc. Subsequently, the MRG generates a

report with a title and a description, as the summarization and verbalization of the enumerated entities. These values are returned to the KBS through the topic *TOP040_TEXT_RERORT_GENERATED*, and are then integrated in the *TOP101_INCIDENT_REPORT* PSAP-update message.

- The initialization of the beAWARE system is signalled by a message in topic *TOP111_SYSTEM_INITIALIZATION*, which contains information about the place of interest and various other parameters. This message triggers the KBS to search the KB for registered places of relief (e.g. public facilities) within a certain radius (e.g. 50km). For each such place, a new incident report is sent to the PSAP (*TOP101_INCIDENT_REPORT*). Moreover, the availability of relief places can be updated via the same topic, once the analysis components discover relevant information within the various attachments.
- As a future task, the KBS will establish communication with the Crisis Classification component (see section 0) in order to feed it with existing knowledge and support its functionality.

Besides monitoring the bus and storing/exchanging information, the KBS incorporates a semantic reasoning mechanism to support and enrich incoming data, by inferring underlying knowledge and discovering interlinkages between incidents during a crisis. This mechanism is rule-based, implemented with a combination of Python code and an elaborate SPARQL ruleset. Specifically, some tasks handled by this reasoning mechanism are the calculation of the *severity index*, the *clustering of incidents* and the *monitoring of safe locations*.

2.3.2 **Crisis Classification**

The Crisis Classification component will encapsulate the necessary innovative solutions that will allow to beAWARE platform to provide early warning and decision support to the Public Safety Answering Point (PSAP).

Crisis Classification (CRCL) system aims to support the beAWARE platform functionalities in two folds. Firstly, as an Early Warning System in order to warn the authorities and first responders of an upcoming extreme event such as the hazard of flood (pre-emergency phase). Secondly, during the crisis, to operate as a real-time monitoring and risk assessment system and to support local stakeholders, authorities and rescue teams to make accurate and timely decisions and take actions.

The components of the CRCL system are connected via Web Services to exchange data and messages. The backbones components of the CRCL system are:

- The Early Warning System which assesses the Overall Crisis Level based on the forecast hydrological and weather data in the pre-emergency phase;
- The Real-time Monitoring & Risk Assessment module which assists authorities and local stakeholders to evaluate the evolution of the flood by estimating its overall crisis level and its severity in order to make efficient decisions and timely actions.

So far, the Crisis Classification system that has already implemented is used to provide early warnings to PSAP and also real-time tracking and assess the risk of a crisis event.

Specifically, for the flood use case, it grabs data from SensorThings Server API which refer to the AMICO forecasts of Water Level over different River Sections in Vicenza district. After that, the Early Warning module compares the water level forecast value with the predefined alarm thresholds of each one of the river sections and estimates the level of crisis based on the outcome of the comparison. The pre-set values of alarm thresholds have been stored to SensorThings Server API and extracted with other data as mentioned in the above section. Combining the crisis levels of the river sections, Crisis Classification module is used to estimate an overall indicator of the forthcoming flood event. The system forwards appropriate messages to the PSAP to notify the authorities and decision makers for the onset extreme flood event. The Real-Time monitoring and Risk Assessment module is triggered when a flood crisis event takes place aiming to track its progress and estimate its severity. In this case, the system grabs sensor data from specific Weather Stations and follows similar process as in pre-emergency phase, the system estimates the level of flood crisis. The system creates and forwards appropriate messages to PSAP aiming to inform the authorities and decision makers for the progress of the flood crisis event.

In the fire use case, the Early Warning System of beAWARE collects aggregated data (EFFIS) which describes the forecasts of Fire Weather Index (FWI) for a period of 10 days. The FWI index is based on the Canadian Rating System and indicates the forecasted fire danger in a specific area. Seven (7) pre-specified points of the Valencia district have been chosen in such a way to enclose the region of interest (Parc Natural de l'Albufera). System estimates the fire danger in each point and the overall predicted crisis level for the whole area for each day. When a risk of fire is high appropriate messages are created and forwarded to PSAP in order to alert the authorities for the upcoming fire hazard in the Parc Natural de l'Albufera area.

In the heatwave use case, the Early Warning System operates in similar way, as above, by collecting hourly forecasting weather data (Air Temperature and Humidity) from six (6) pre-set points in the region of Thessaloniki. The goal is to estimate the hazard of heatwave per day in Thessaloniki district. Thus, Crisis Classification system calculates the Discomfort Index (DI), which is useful to evaluate how current temperature and relative humidity can affect the discomfort sensation and cause health danger in the population. DI is estimated over the six points of interest by employing the hourly weather forecasts. Then, the system aggregates the DI estimations in order to predict the heatwave danger per day in the whole district. The appropriate messages are generated and processed to PSAP when the Discomfort Index per point and predicted heatwave crisis level exceed the high level, in other words when most population feels discomfort and deterioration of psychophysical conditions.

2.3.3 **Concept and event detection from multimedia**

Concept extraction from visual content (image/video) in the beAWARE project is supported by two separate components, namely IMAGE ANALYSIS and VIDEO ANALYSIS. Those currently have integrated modalities that include a fire and flood detection system, as well as functions for detecting and estimating the severity of people and vehicles that are in danger.

For those tasks the image and video analysis components include several interoperable modularities that deploy an array of cutting-edge computer vision techniques:

- Image classification so as to determine **which images/video frames contain an emergent event or not** (i.e. a fire of flood event)

- Object Detection so as to **find people and vehicles that exist in the images/videos**.

Each one of them is assigned to process an image or a video frame separately from the others in order to decide about the existence of fire and flood concepts and objects that are of particular interest like people and vehicles and later locate their position inside the image. Then, a severity level estimation module is assigned with the task of deciding about the danger that the people and vehicles undergo based on their proximity to the emergent event.

This procedure is triggered by the MEDIA HUB component that is responsible to inform the IMAGE/VIDEO ANALYSIS listener about new incoming analysis requests. A link referring to the location of the media to the OBJECT STORAGE (CDR) is provided in order to download the appropriate media file and start the analysis. To handle an arbitrary number of simultaneous incoming calls every analysis request is placed last into a FIFO queue. The FIFO queue is processed in a sequential manner in order to provide results more efficiently. After the analysis is complete for each item, an analysed version of the media fire is uploaded to the OBJECT STORAGE and provided back using the appropriate MEDIA HUB response connection. An example of an analysed image is shown in Figure 4.

Figure 4: Example of Analyzed Image

Expected meaningful outcomes from the analysis for the 1st prototype, that are passed through the MEDHUB into the KB service, include the information about the type of emergency present in the media file (i.e. flood or fire) as well as the location and class of the objects of interest that have been found.

2.3.4 **Automatic speech recognition**

An automatic speech recognition (ASR) component has been developed and integrated into the operational beAWARE platform, in order to transcribe audio recordings from citizens and first responders in four languages (English, Spanish, Italian, Greek). The ASR communicates with the Media Hub component through socket messages. The Media Hub subscribes to TOP021_INCIDENT_REPORT and triggers the ASR module in case the attachmentType in this topic is set to "audio". The socket message form the Media Hub to ASR is a JSON message with the link to the audio file, the language, and the timestamp and the Incident ID. The input audio should be “RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, monophonic, 16000 Hz”. However, an encoder has also been included, in case the input format is not the appropriate. By reading the language information, ASR component selects the corresponding language model, invokes the transcriber, creates the transcription and sends a JSON message back to the Media Hub, with the transcription text. Subsequently, the Media Hub creates a TOP010_AUDIO_ANALYZED topic, containing, among other fields, the transcribed text and the language of the user. MTA, which is subscribed to this topic, receives this information in order to further analyze the transcription.

Speech Recognition is performed by taking an audio waveform, splitting it at utterances[2], by utilizing silences and then trying to recognize what is being said in each utterance, by matching all possible combinations of words with the audio. The extraction of the best matching combination is based on three entities:

- An **acoustic model,** which contains acoustic properties for each basic speech segment.
- A phonetic **dictionary,** which contains a mapping from phones (speech segments) to words. A dictionary can contain alternative pronunciations for the same word.
- A **language model,** which is used to restrict word search. Language models contain statistics of word sequences and define which word could follow previously recognized words. They help to significantly restrict the matching process by stripping words that are not probable.

These three entities are combined together in an engine to recognize speech (Transcriber).

In order to assist MTA (see section 2.3.5) in the semantic analysis of the trasncriptions, we developed a simple post-processing step in order to split text into potential sentences, by using the time durations of the silences between words, in order to provide automatic punctuation.

2.3.5 **Multilingual Text Analyzer**

The beAWARE platform can process natural language text obtained either from social media and textual messages sent to the platform by user using the mobile application, or from the transcription of audio messages also sent using the app. Text in any of the project languages –English, Greek, Italian and Spanish—is analysed using a set of natural language processing and information extraction tools, which produce language-indented representations of the information conveyed in each analysed message. These representations express relevant events, concepts, entities and relations identified in the text. The output of the analysis of each message is passed to semantic processing components for its integration with information obtained from other sources.

2.3.6 **Multilingual report generation**

Starting from contents in the knowledge base, the multilingual report generation modules produces multilingual texts providing to the users of the platform with relevant information about an emergency. Two types of report are supported, short reports to provide situation awareness related to one or multiple incidents, and wrap-up summaries. While situational reports are updated often to provide recent information available to the beAWARE system, wrap-up summaries are to be generated at the end of a crisis in order to provide authorities

[2] Chunks of speech between pauses, containingwords and other non-linguistic sounds, which are called fillers (breath, um, uh, cough).

with a general overview of how incidents unfolded during the crisis. Reports are first planned by selecting and ordering contents made available to it by the KB services and obtained from the analysis of media collected by the system or as a result of reasoning and semantic enrichment tasks. A text is then produced that communicates the selected contents in any of the beAWARE languages (English, Greek, Italian and Spanish). The module implements various knowledge-related and linguistic strategies to ensure that the resulting summary is relevant, concise, non-redundant, coherent, grammatical and fluent.

2.3.7 Social Media Analytics

The Social Media Analytics tool is responsible for crawling, storing and analysing Twitter content that could be relevant to the use cases examined by the beAWARE project. With the usage of Twitter's Streaming API[1], we collect posts in English, Italian, Greek, and Spanish, which contain predefined keywords that refer to flood, fire or heatwave incidents. Then, classification techniques exploit the text or the image of each tweet to label it as relevant or not. All posts are stored in a Mongo database, but only relevant ones are used in the flow of the beAWARE system.

2.4 Internal Services Layer

These services are used internally for the proper functioning of the capabilities provided by the various components and represents the middleware communication services which are tailored for the specific use of the beAWARE system

2.4.1 Communication Bus (MSB)

The main purpose of this component is to provide generic communication capabilities among different beAWARE components and participants. It can be used to send messages and notification among components and to share information among various entities. In a microservices based architecture, such as beAWARE has adopted, there is a need for communication among different microservices, and one of the dominant manners to achieve that is by using a distributed publish / subscribe mechanism. The beAWARE used Communication Bus provides the ability for different entities to send and receive messages without having to be aware specifically of each other. The agreed upon pieces of information to enable such an integration are the topic name which shall be used for a specific kind of interaction, and the message format, such that different entities will be able to understand each other. Extensive work has been done in beAWARE to reach an agreed upon list of topics and their corresponding formats (which are specializations of a generic message format including a header and a message body).

The dominant communication paradigm used within beWARE is the publish/subscribe pattern leading to event-based communication among collaborating partners by registering interest in particular events. Event-based communication is often used in enterprise architectures (SOA) as it decouples any producer and consumer in terms of location and time (asynchronous communication).

A typical beAWARE flow, based on the communication bus, is for a component holding a new piece of information that needs to be processed by another component to store the information to be shared in a temporary raw data store, publish a message on the corresponding message bus topic, providing in it a link to the current location of the information to be processed. The receiving component in turn receives the message, parses it, retrieves the stored data via the supplied link, processes it and in turn may produce a new piece of information that need to be passed to yet another system component. All further interactions will follow a similar flow.

The communication bus is configured, upon deployment, with the necessary set of topics as agreed upon between the different components. In addition, the message structure of each message in each topic is agreed upon and documented by the cooperating components. The communication bus supports the number of different topics required for a beAWARE installation, along with the associated aggregated throughput in all topics.

The communication bus is realized by using an instance of a MessageHub service, deployed in IBM's BlueMix cloud. The back-end is based on a Kafka cluster, and the interaction with the service is realized using standard Kafka clients.

The communication bus has been deployed as a central component of the beAWARE platform for over a year. It is being extensively used by most components on a regular basis.

2.4.2 Data management

The data management layer consists of several components, like shown in the overall architecture (see Figure 2). Semantic data is handled inside the knowledge base. All time series based information, like sensor data (measurements from weather stations, measurements of the river level), weather forecast as well as river level predictions are stored inside the SensorThingsAPI-Server. The SensorThingsAPI-Server is a Fraunhofer IOSB open-source server implementation of the OGC SensorThingsAPI standard. The SensorThingsAPI standard defines a data model and offers a REST-API. The data model consists of several entities and describes the sensor data, as well as geographic metadata (e.g. location). The REST-API and the server implementation facilitate the integration of heterogeneous data sources. For example, the raw data in the SensorThingsAPI Server can be visualized in the knowledge base and is also at disposal for all project partners. All other data types e.g. images, videos, audio files are stored inside the central data repository, which offers the possibility to store single files and therefore is the choice to store multimedia files. All these data sinks have been deployed to the beAWARE platform and are integrated into the first prototype. These storage solutions are passive components, which means they cannot sent notifications for new events. Other components e.g. the crisis classification queries these databases to extract the needed information, processes them and notifies the other components via the message bus. To import newly existing data, import components have been developed and deployed to always have the latest data inside the beAWARE system.

2.5 External Facing Layer

2.5.1 PSAP

The objective of this component is to serve as a means for public safety answering points (PSAP) to obtain situational awareness and a common operational picture before and during an emergency, and to enable efficient emergency management based on a unified mechanism to receive and visualize field team positions, incident reports, media attachments, and status updates from multiple platforms and applications.

PSAP's Functional Characteristics:

- ✓ Displays metrics on the map.
- ✓ Displays teams on the map.
- ✓ Displays incidents on the map.
- ✓ Display critical assets (e.g., places of relief) on the map.
- ✓ Sends public alerts from a fixed list of texts.

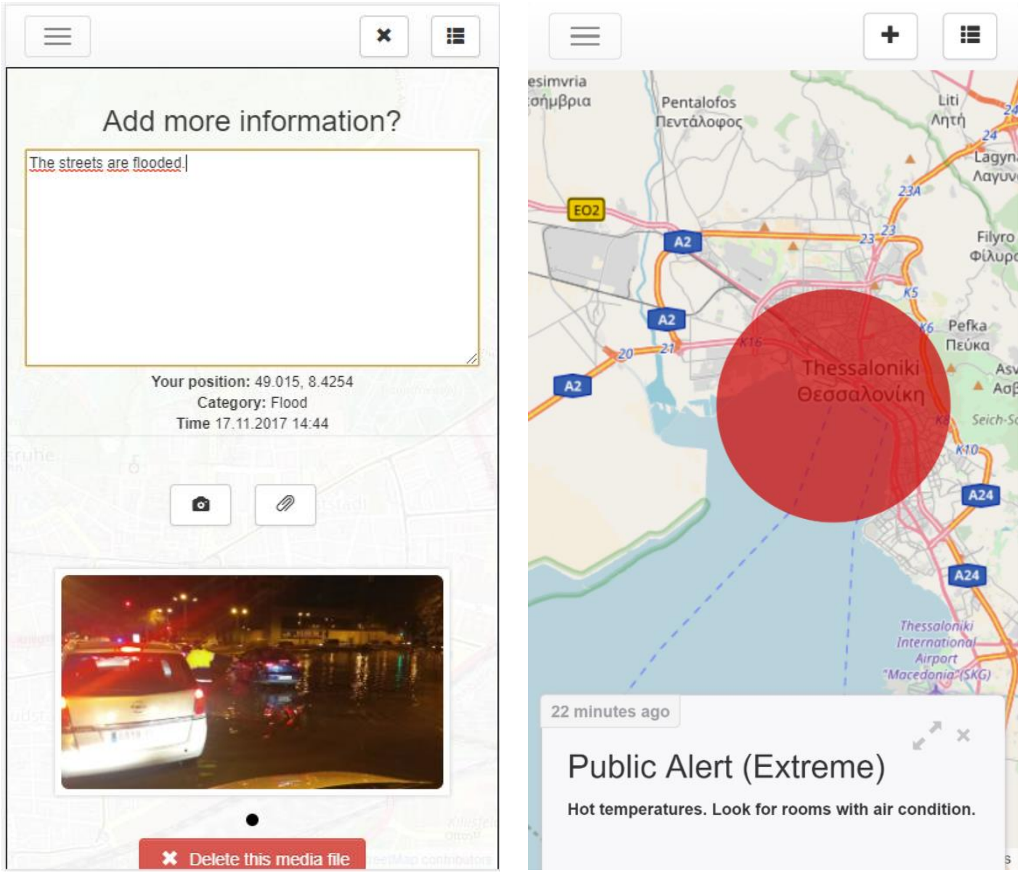
Functional Flows

- **Metric Flow:** A critical aspect in the metric flow is the ability to visualize measurement series on the map, such as sensor readings or flood forecasts. These metrics are sent to the PSAP from the crisis classification module (*TOP104_METRIC_REPORT)*
- **Incident Flow:** The incident flow is one of the fundamental information flows in the beAWARE platform. PSAP receives and displays incident information from first responders in the field or bystanders (citizen) who provides the information and footage from their mobile application or through social media posts. These reports are clustered by the KBS and associated with previous reports as it has been described on a previous section and displayed respectively on the map.
- **Asset Flow:** The asset flow allows for defining and displaying locations of critical assets on the map (e.g places of relief). The initialization of those assets is triggered by message that invokes the KBS to generate incident messages, one for each asset, and send to the PSAP to display. The status of those assets can be updated via the same topic message as it has been described previously.
- **Team Flow:** The team flow supports the monitoring and tracking of first responder teams' positions and status in the field by the authorities.
- **Public Alert Flow:** The public alert flow allows the authorities to warn the public and first responders by sending public alerts regarding incidents and general risk assessments on their mobile devices.

2.5.2 End-users applications

The first prototype of the end user applications are developed and integrated into the overall platform. They are available for Android based smart phones as well as a web version, which is accessible by every internet device, capable of displaying web sites. This offers the possibility to use the main functionality of the mobile application on other than Android devices, e.g. iPhones or desktop PCs. For the first prototype version there is no distinction between a citizen and a first responder application. In a later step, first responders can sign in to enable further functionality, like receiving tasks to handle.

The main functionalities of the first prototype are the creation of multi-modal incident reports and the receiving of public alerts, send by the authorities. All interactions with the application are triggered through a map-based interface. On this the incident reports send by this mobile device are displayed, as well as the public alerts.



a) b)

Figure 5: a) Interface for creating an incident report. β) Example of a Public Alert.

Figure 5 (a) shows the interface for creating an incident. If no specific information is given, the location of the incident is equal to the current position of the user of the application. To report incidents on specific places it is also possible to double tap on the map to send a report happening at that place. Each report can contain photos, videos and audio recordings as well as a textual description. Multimedia files can be selected from existing files on the device (e.g. a photo has been taken before and the report is created after a person got to a safe place) or they can be captured directly. These reports are sent to the analysis components and are displayed in PSAP.

In the first prototype authorities have the possibility to send public alerts to the users of the mobile application. Public alerts are created in PSAP and are sent to the mobile application. The device of the user shows a notification that there is a new public alert. Figure 5 (b) shows the affected region of the public alert together with the message of the authorities: “Hot temperatures. Look for rooms with air condition.”

3 Use Case Scenarios

beAWARE is a use-case-driven project that aims to cover the functionality required by three relevant operational scenarios: flood, fire, heatwave. The use cases serve as an important validation point to test the communication flow, the level of integration, to reveal any weaknesses or shortcomings found in the tryouts and to validate that platform meets the requirements set by the end users.

In order to demonstrate the whole function of the first prototype and the implementation of all three use cases as they are defined in D2.1, a demonstrator video has been created, capturing the whole function of the platform throughout the sequential steps of the UCs. This is a public report, thus only the non-confidential parts of the implementation are described in the following sections.

The video is available in the following **link**.

3.1 Flood Use Case

For the demonstration and evaluation of the features implemented in P1, a hypothetical storyline is assumed, taking example from the great flood of November 2010 when the Vicenza city centre, a very busy residential and commercial area, was submerged. The main objective of the demonstrator is to highlight that, in every phase, the beAWARE technology implemented in P1 is able to provide a detailed and updated outline of the situation, with the incident reports, physical sensor measurements, forecasts and additional available information. Thus, it offers a great decision support tool to the Authority and also helps the rapid diffusion of public alert to all citizens. The storyline for this scenario is composed of four phases.

- **PHASE 1: Pre-emergency phase**
  The Flood Forecasting System AMICO generates periodically a flood prediction based on the most recent available weather forecast provided by FMI. The forecast is transmitted, stored, analyzed and visualized by the SensorThingsApi. This module defines a data model consisted of entities which describe the sensor data, as well as geographic metadata. The Crisis Classification component acquires this data to classify the crisis event and provides early warnings that are sent to the PSAP. For activating this pre-emergency phase an iconic flood forecast is revived based on historical data from the great flood of November 2010. The PSAP receives these TOP104_METRIC_REPORT via the message bus. Upon user request, a metric map and a dashboard interface are displayed to the user and allow several ways of interaction.
  According to the prescribed scenario, after receiving on the PSAP the indication of the forecasted risk levels, the decision maker notifies the public sending a public alert. Public alert flow allows the authorities to warn the public and first responders about hazards before or during an emergency by sending notifications on their mobile devices. By sending this alert the Authority upgrades the level of emergency.

  Real-time Monitoring & Risk Assessment layer within the crisis-classification module is activated evaluating the evolution of the flood by estimating the overall level and the severity of the ongoing event. The estimated factors are sent to the PSAP to support the constant monitoring of the forthcoming event by the authorities.

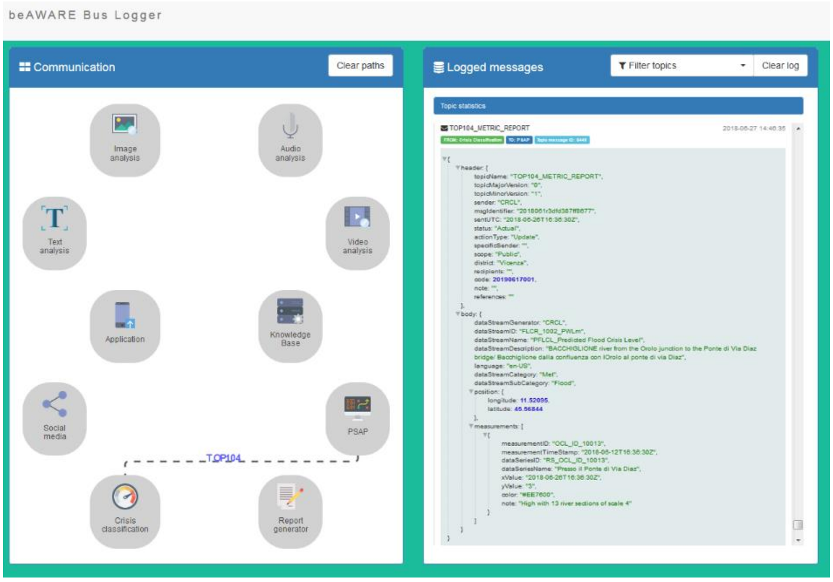


Figure 6: Metric Flow created by Crisis Classification component

➢ **PHASE 2: Image & Video Analysis**

During the second phase of the Flood UC, the first incident report by the mobile application is provided by a citizen in Matteotti Square, with an image attached about some cars submerged by floodwater. An incident report message is sent to the message bus notifying the KBS and Image-Analysis module for the new data ingestion (*TOP021_INCIDENT_REPORT*). The KBS in turn produces corresponding messages to notify the PSAP about the new incident (*TOP101_ INCIDENT_REPORT*).

Consequently, the Image-Analysis module analyzes the image to extract conceptual information regarding the incident. The analysis results are saved in the KB and the incident severity is updated. The KBS produces new message to notify the PSAP about the incident update. Additionally, the KBS requests new title and description from the MRG and updates the incident. The relevant messages and flow can be seen in Figure 7

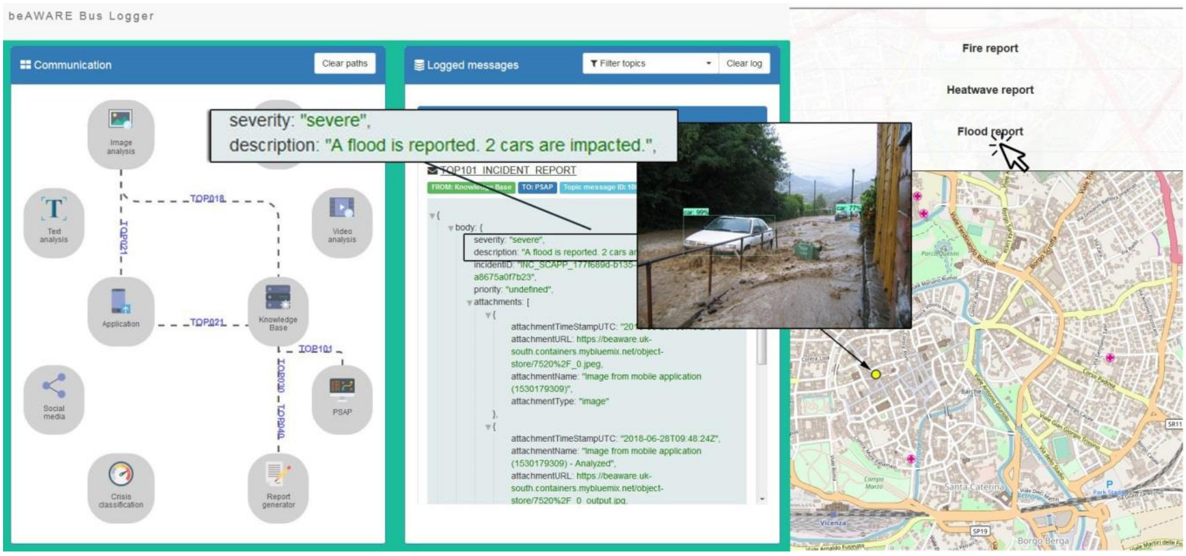


Figure 7: image analysis, along with the corresponding information flow. A new incident is reported.

The same procedure is being followed to simulate a second incident sent on a different location.

Right after a video is sent in a location relatively close to the first reported incident documenting scenes of flooded streets. The KBS gathers the additional information under a common incident ID performing a proximity-based clustering. After video analysis a message in topic *TOP030_REPORT_REQUESTED* is produced by the KBS, with both incidents and incident participants enumerated, along with additional details, such as severity, timestamps, etc. Subsequently, the MRG generates a report with a title and a description, as the summarization and verbalization of the enumerated entities. These values are returned to the KBS through the topic *TOP040_TEXT_RERORT_GENERATED* and eventually are integrated to the PSAP narrative, enriching the initial report with further information.

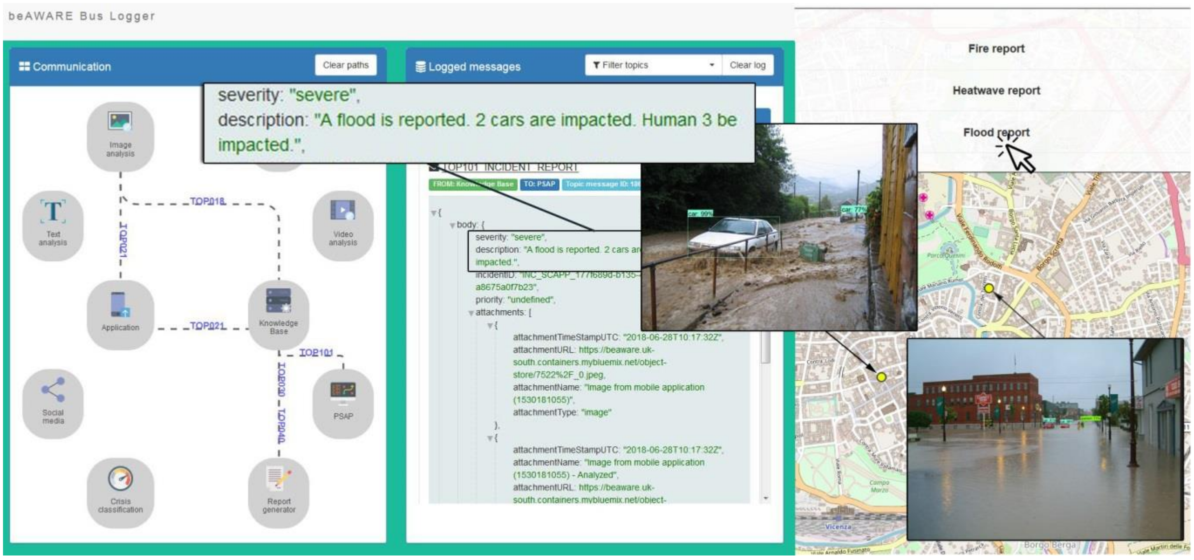


Figure 8: A second flood incident is reported through the mobile application.

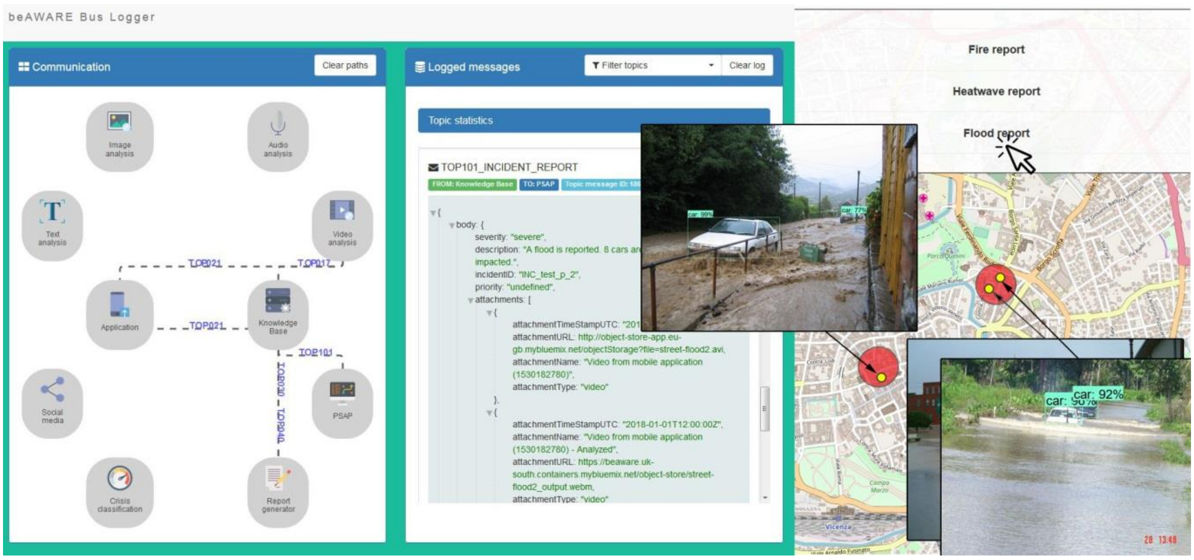


Figure 9: Proximity-based clustering is performed on the new ingested incident report.

➢ **PHASE 3 – Social Media**

During the third phase of the UC, the beAWARE Social Media Analysis module crawls social media content exploiting Twitter's Streaming API and after it classifies the posts as relevant or irrelevant; it forwards them to the Knowledge Base Service (KBS) to populate corresponding incident reports and to the Multilingual Text Analyzer (MTA) for further analysis. Then, the KBS assembles all the information from the tweets and the tweet analysis and produces corresponding messages to notify the PSAP.

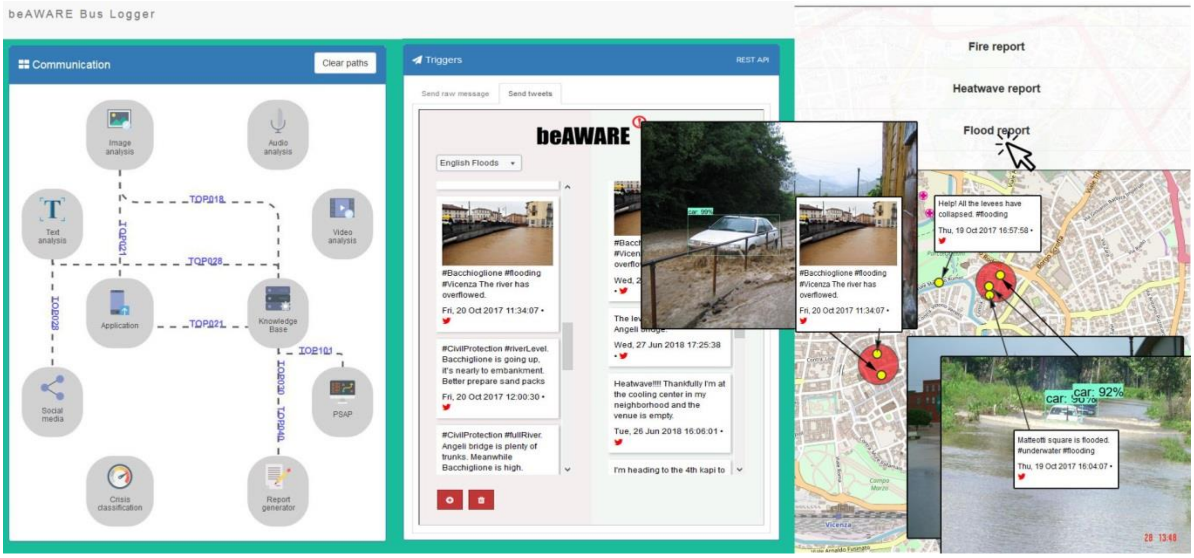


Figure 10: New ingested tweets are converted into incident reports and clustered together with existing incident reports

➢ **PHASE 4: Public Alert**

In the final phase of the storyline, a public alert is sent from the PSAP notifying people on their mobile devices. The alert shows the affected region together with a message by the authorities

3.2 Fire Use Case

This use case refers to a simulated forest fire event close to Valencia. Again, the pilot operational scenario is based on a real event that occurred in August 2015 on the territory of La Devesa de l´Albufera (Valencia). The storyline for this scenario is composed of four phases.

➢ **PHASE 1. Pre-emergency phase**

The first phase is the pre-emergency phase where the meteorological sensors forecast poses for an extreme risk of fire. Crisis Classification component is triggered to collect weather data (EFFIS) of fire weather forecasts for a period of 10 days. Seven predefined points have been chosen to enclose the region of interest (Parc Natural de l'Albufera). System estimates the fire danger in each of these points and the overall crisis level per day and creates metric reports (TOPIC104_METRIC_REPORT). These reports are sent to the PSAP where the positions of the measurements are visualized in a dedicated map layer.

➢ **PHASE 2**: **ASR mobile application**.

The second phase is initiated when the PSAP receives a fire warning, from a citizen sending an audio message from his mobile application. A second incident is reported by another audio message. The Media Hub subscribes to the messages published by the mobile application and triggers the ASR module. ASR component selects the corresponding language model, invokes the transcriber, creates the transcription and sends the transcription text to MTA module. As previously described, the analysis results are saved in the KB. The KBS produces one message per incident to notify the PSAP. Finally, the MRG is activated to generate a narrative summary based on the reported incidents.

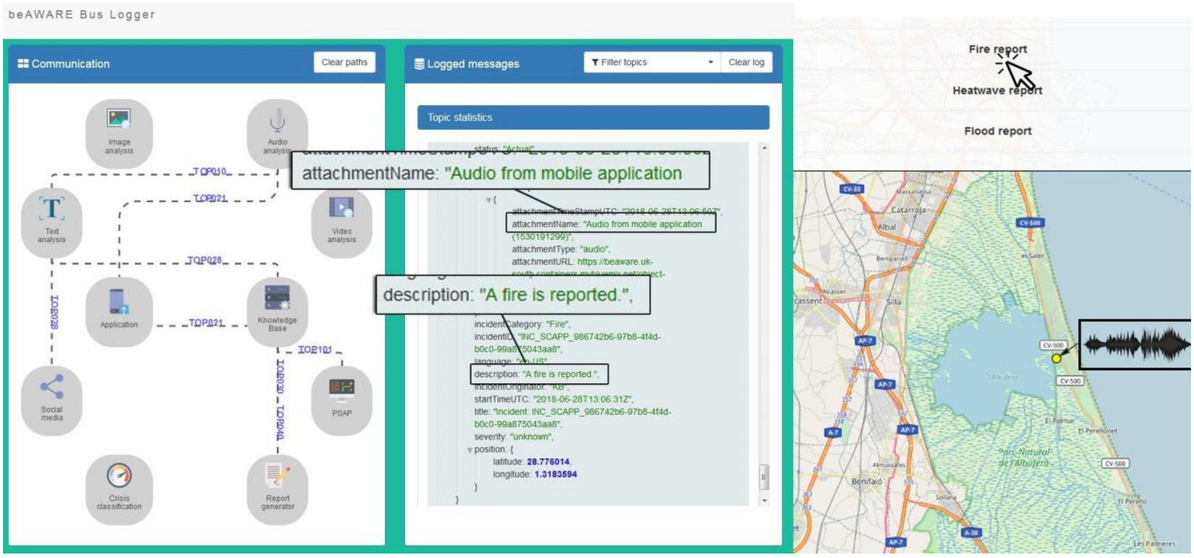


Figure 11: Audio analysis, along with the corresponding information flow. KBS forwards a message to the PSAP followed by a summary based on the reported incident.

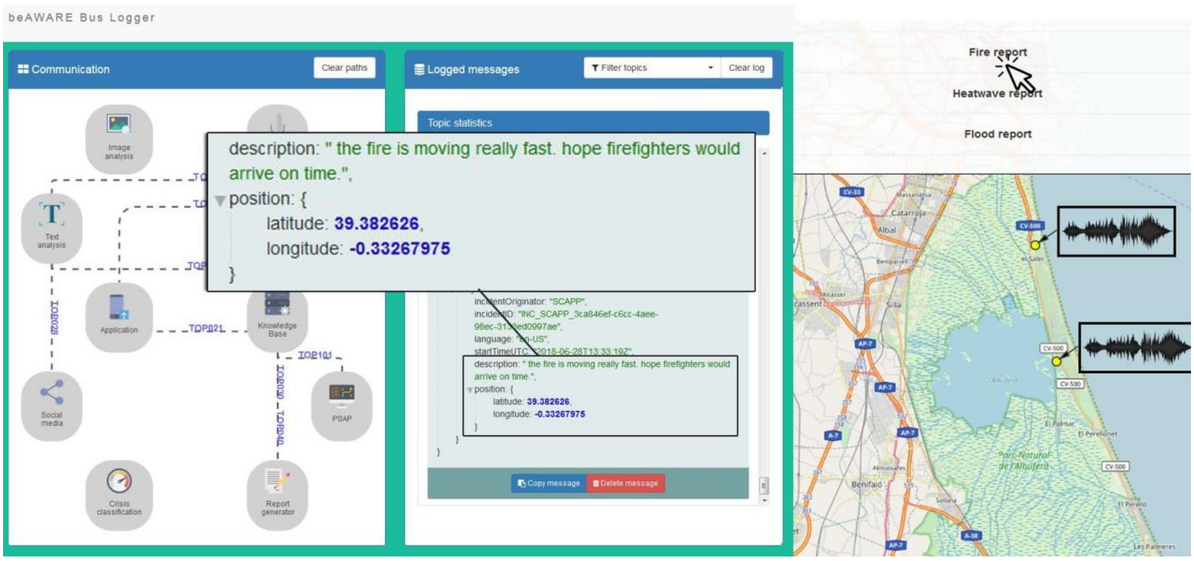


Figure 12: A second fire incident is reported with an audio message via the mobile application.

Subsequently, a third incident is reported, confirming the fire emergency, via a text message coming from the mobile device of a first responder. Following the same flow through the MTA-KBS-MRG the reasoning mechanism of beAWARE discovers interlinkages between new and stored incidents. Proximity-based clustering is then done and the updated information is forwarded to the PSAP.

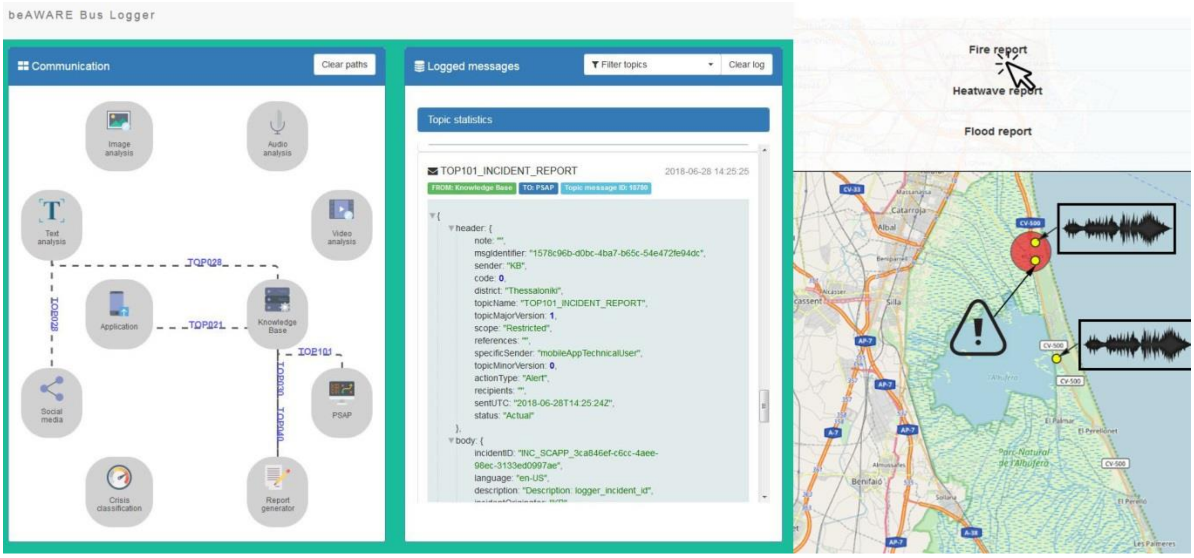


Figure 13: New iincident confirms the fire emergency, via a text message coming from the mobile device of a first responder

➢ **PHASE 3**: **Public alert.**

This next phase is related to the generation of a public alert by the PSAP operator with a notification about the specific location of the event. The generated alert is sent to the MSB for distribution to relevant subscribers.

➢ **PHASE 4**: **Image Analysis & Clustering.**

Finally, a civilian uploads an Image from his mobile application. The KBS monitors the analysis results and associates the detected entities with the analysed media items and the original incident reports that had been previously created in the KB. The results are sent as a descriptive narrative, followed by Title and Description, for the short, intuitive presentation of the incidents to the user interface.

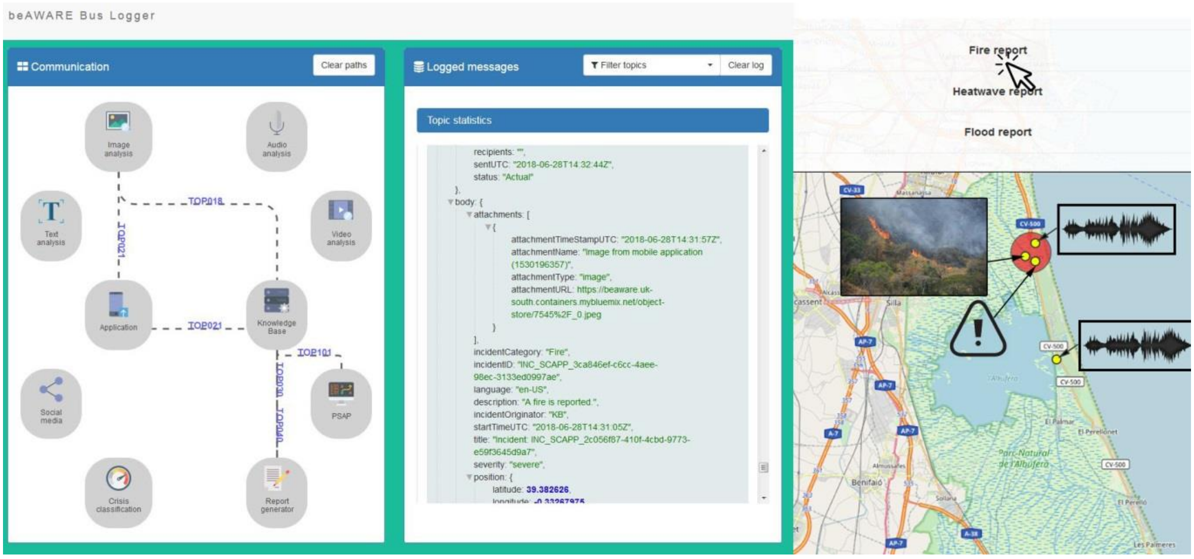


Figure 14: A new image updates the first incident reported on phase 2.

3.3 Heatwave Use Case (Management of Places of Relief)

P1 pilot for the heatwave case scenario focuses on the management of the places of relief. The goal of P1 for the heatwave is to look at the ability to manage places of relief in the event of a heatwave scenario.

The storyline for the simulated event that describes fire UC is divided in four phases. The separation in phases allows in having a clear distinction of the steps to be performed.

- **PHASE 1: Pre-emergency phase**
  In the heatwave use case, the Early Warning System operates in similar way, as before. Crisis Classification component is triggered to collect forecasting weather data (Air Temperature and Humidity) from six (6) pre-defined spots in the region of Thessaloniki. For demonstration purposes, prefabricated data are used to simulate a heatwave emergency. Crisis Classification system calculates the Discomfort Index (DI) to evaluate the discomfort sensation and health danger in the population. Then, it generates and forwards messages to the PSAP notifying the authorities about the progression of the heatwave emergency.

- **PHASE 2: Places of Relief**
  At the second phase it is signalled the insertion of Places of Relief by a message in topic *TOP111_SYSTEM_INITIALIZATION*. This message triggers the KBS to search the KB for registered places of relief (e.g. public facilities) within a certain radius (e.g. 50km). For each such place, a new incident report is sent to the PSAP and incidents are shown on the map. As part of this demo, two reflective incident reports are generated for two designated places of relief respectively in the Thessaloniki district.

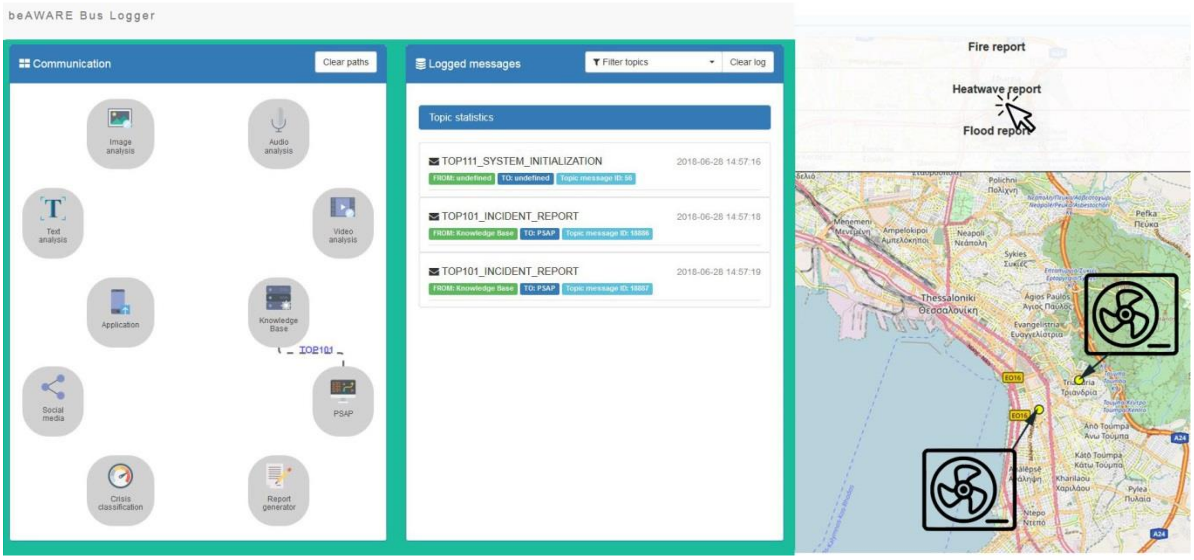


Figure 15: Insertion of Places of Relief

- **PHASE 3: Public Alert**

At the 3rd phase, a public alert will be sent from the PSAP notifying that relief places are open. As part of this demo, public alerts are generated for the whole city as a general notice.

- **PHASE 4: Social Media**

As it has been described, for the Flood scenario during social media demonstration phase, the beAWARE Social Media Analysis module crawls social media content classifying the relative to the UC scenario Twitter posts. Relevant tweets are forwarded to the KBS to populate corresponding incident reports and to the MTA for further analysis. The KBS assembles information from tweets and tweet analysis and notifies the PSAP. On this showcase is highlighted the update on the availability status of the relief places. This happens right after the analysis components discover relevant information within the ingested information.

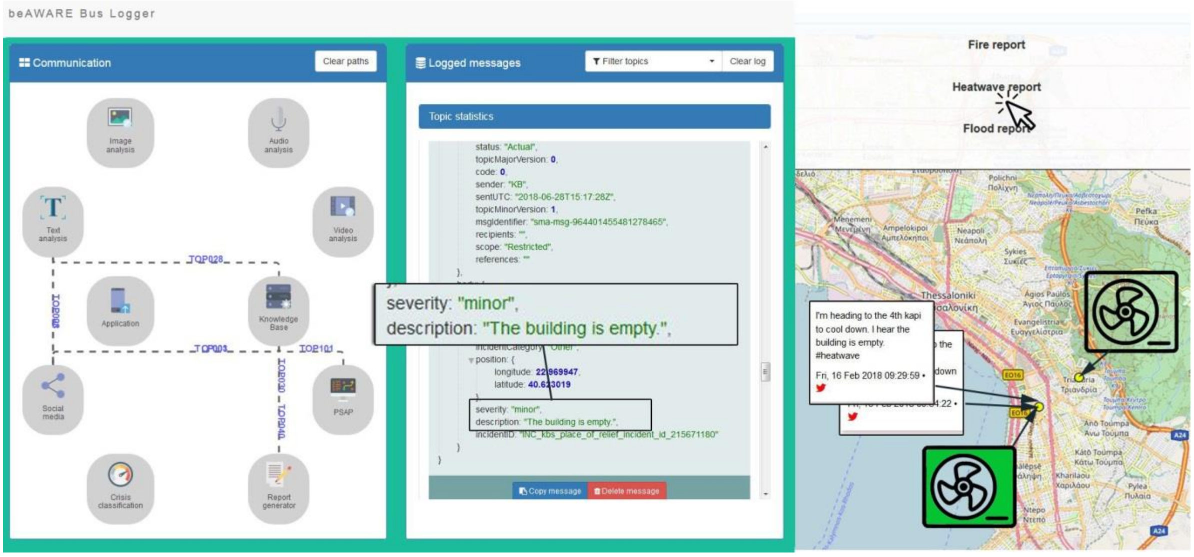


Figure 16: The availability of relief places is updated once the analysis components discover relevant information within the various attachments

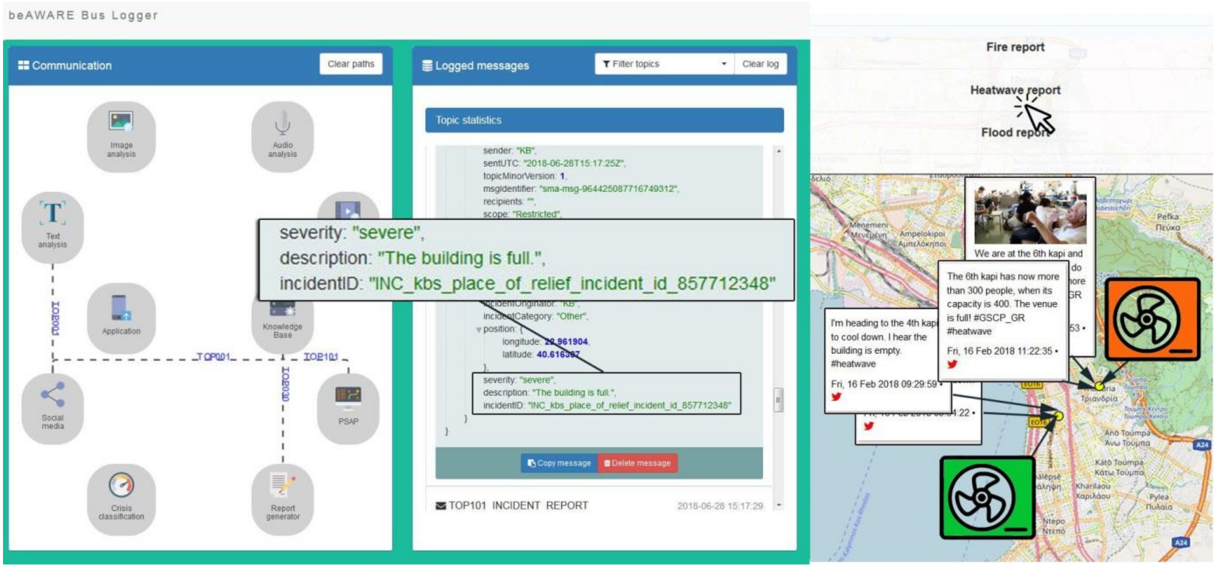


Figure 17: "The venue is full #heatwave". Relevant tweets are forwarded to the KBS to populate corresponding incident reports. Consequently, the PSAP is notified that one of the places of relief is full.

- **PHASE 5: Public Alert**

Finally, in the last phase a public alert is declared from the authorities warning citizens and first responders not use the first place of relief, directing people to an alternative/secondary place. At the end of the day a message will be send to all citizens that the phenomenon will deteriorate in the next days.

4 Technical Evaluation of the system

Following the guidelines of the beAWARE technical evaluation strategy, technical meetings were regularly organized and several test cases were performed so as to validate and verify 1st version's achievements. The evaluation focused on covering as many Key Factors as possible with respect to the technical work.

4.1.1 Testing and Evaluation Methodology

Prototyping serves for software development, regression verification, and in-system validation. In other words for testing if each component involved in creating and delivering the system works well with other components, and if the entire scale of operations can run smoothly.

In order to conclude about the technical evaluation, it is noteworthy to refer to the extended efforts of testing that we committed upon, through teleconferences with other partners, so as to solve intra-communication problems with other components as well as internal functionality misbehaviours.

At project integration level, we used weekly plenary WebEx meetings to provide updates on the current status of the different parts of the project in order to accomplish with specific deadlines following a SCRUM testing methodology.

In the following is described this testing methodology which was tailored to the integration of the 1st version of beAWARE platform. Our testing framework was designed with reference to the requirements that are defined in D2.3 "Pilot User Cases for the 1st prototype". As previously mentioned beAWARE is a use-case-driven project where three different use cases based on real crisis scenarios were defined by the end-users to validate the under-deployment platform. These use cases are consisted by a sequence of steps that describe the interactions between the users and the system.

Within beAWARE testing framework meetings were organized and facilitated by the scientific manager assisted by the technical manager of the project. Every week a different sequence of the storyline from the Use-Cases was set by the scientific manager to be tested, creating a backlog – a list of tasks to perform during the meeting. These tasks were forwarded before meetings via email to all participants and were populated on beAWARE's DokuWiki under the page "Technical Telcos – Minutes and action points". By this manner and assigning roles to the participant "actors" the testing procedure was well tuned and all members had a clear idea about their responsibilities.

In this integration of the 1st prototype of the platform all testing scenarios were consisted of two main phases. The phase before the crisis also named pre-emergency phase and the phase during the crisis event. The main "acting roles" in the sequence of testing scenarios were those of the citizens, the first-responders and the authorities. These roles are incarnated in the context of the **External facing layer** of the platform.

On one hand by the mobile application which is the contact point for the citizens and the first-responders to interact with the platform by sending reports including audio, text messages, audio, images etc. On the other hand, by the PSAP which is the interface tool to obtain situational awareness and a common operational picture before and during an emergency.

The aforementioned tools were serving as the principal criterion for the stakeholders to monitor and to evaluate the process. Furthermore, an additional tool serving for internal purposes as a message logger was used for the graphical representation and monitoring of the messages exchanged between the different components.

The main idea of integration testing was to combine gradually and test the interface between the key components and eventually to expand in order to test all the integrated components of the platform fulfilling a complete flow that describes a Pilot Use Case scenario.

Thus, the scientific manager, orchestrating the whole process and with respect to the predefined testing plan, was requesting for actions from technical partners such as an image or a video to be uploaded indicating an event, an audio message to be sent reporting an incident, a pre-emergency phase to be triggered and so on.

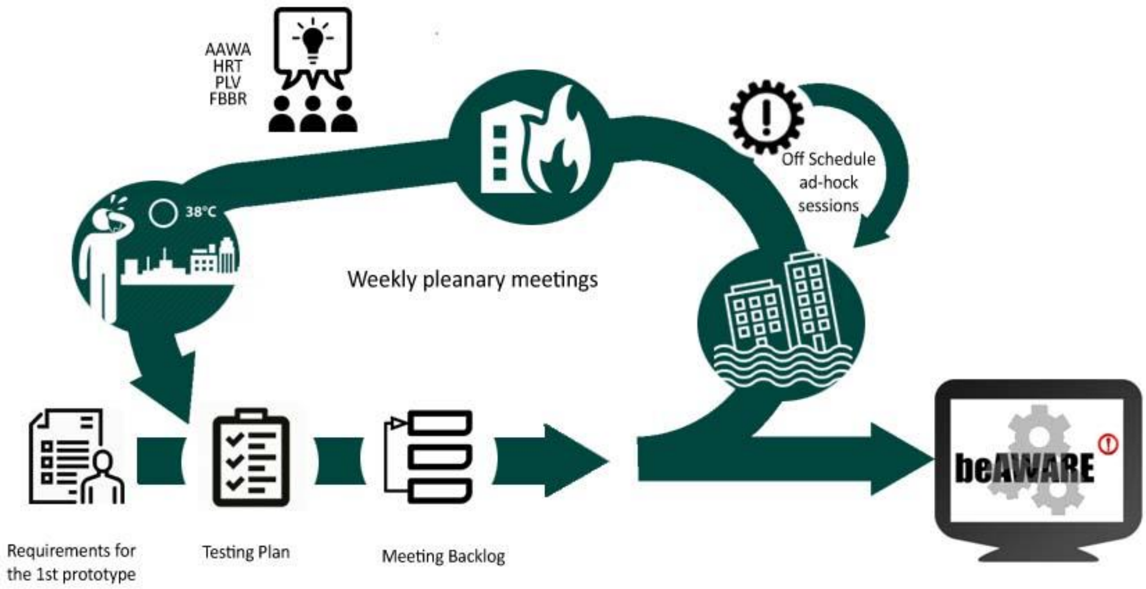


Figure 18: Testing and Evaluation Methodology

This approach emphasizes the value of collaboration and the rapid feedback that iterative usability evaluation can provide, its effectiveness typically relies more on the involvement of End Users - usability experts.

To summarize, WebEx plenary meetings were serving as weekly reviews for testing the platform, during which the teams demonstrate the new functionality to the end-users who provide feedback that could influence the next meeting. Technical overview combined to the feedback from the stakeholders was resulting in changes to the delivered functionality, but also in revising or adding items to the product backlog. Defects, change requests, new tasks were accumulated after session to feed the Issue Management tool that is described in the following subsection.

Various ad-hock sessions were also scheduled jointly with the regular meetings. The main goal of those sessions was to fill the software gaps between components

For intertask work a more informal communication protocol has been used via direct mailing or Skype group chat/calls

4.1.2 Issues Management

There is no doubt that issue tracker, is one of the key factors for evaluating the software process development. It provides a platform to support software development and maintenance activities such as issue reporting, tracking and resolution. Issue tracking (such as a bug report or a feature enhancement request) is a social and collaborative process in which an issue tracker serves as a communication hub and channel between the developers and QA (Quality Assurance) team.

For the issues found during development and technical tests of the beAWARE platform, the consortium partners are using GitLab embedded issues management approach to store issues, create milestones and distribute the technical work needed to overcome these issues

GitLab issue tracker serves the following purposes:

- Safe and reliable method for the team to raise issues.
- Track and assign responsibility to specific people for each issue.
- Analyze and prioritize issues easily.
- Record issue resolution for future reference and project learning.
- Monitor overall project health and status.

For the better organization of the issues, specific labels have been created and are used in order to prioritize and to map the issues with the platform components, as depicted in Figure 19

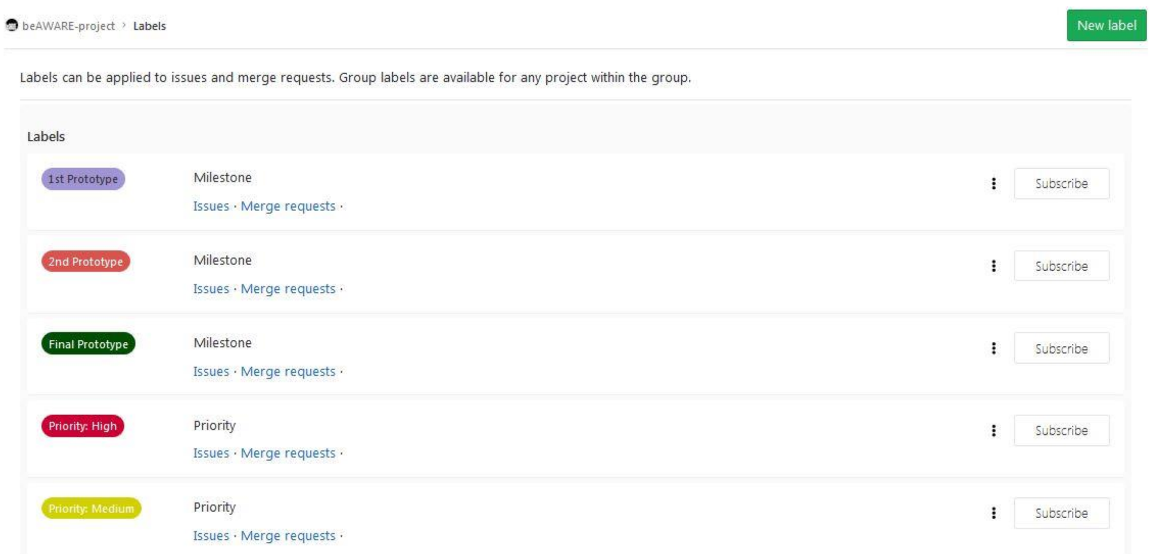


Figure 19: Issues Management for beAWARE components - Labels

Finally, in order to impose the time plan for addressing the reported issues, specific Milestones are used with reference to the technical weekly meetings The management of issues, labels and milestones is open to all technical partners and is done at repository level, thus changes are made easily when needed. The view of issues however can be aggregated at the root level beAWARE-project in GitLab, therefore it is easy to have an overview of the issues of whole framework, as shown in Figure 20.

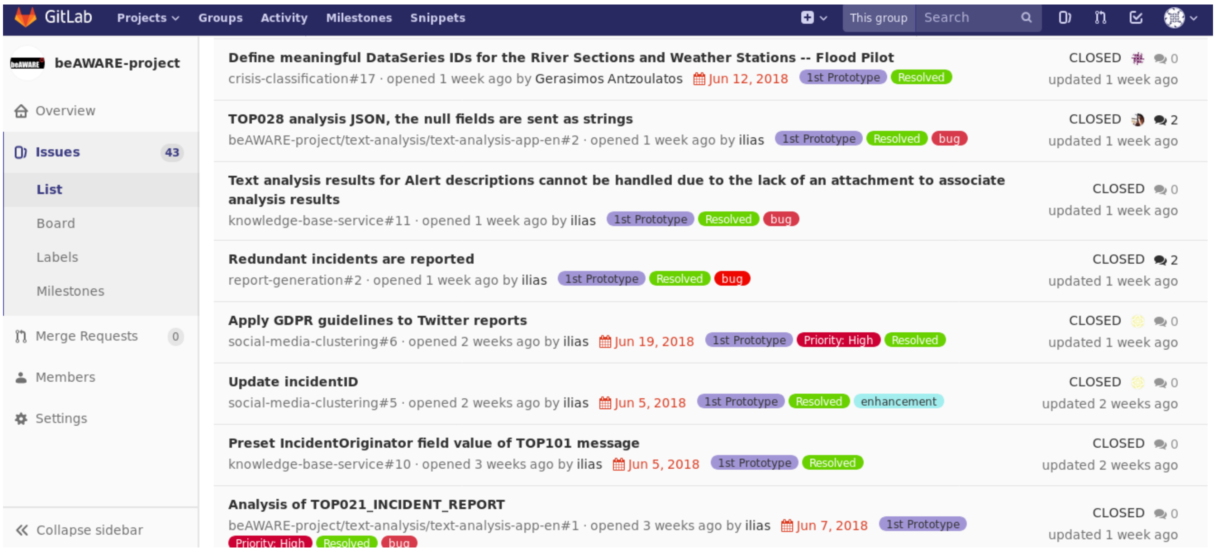


Figure 20: Aggregated issues from all components

4.1.3 Evaluation of Key Factors

For the evaluation of the quality of the 1st version of the platform we selected to present some quality metrics of the modules which are considered as the key factors of the implementation.

Knowledge Base Service

The interaction between the WebGenesis knowledge base component and the Knowledge Base Service (KBS) is based on the execution of complex and elaborate queries from the latter to the first. The processing times of these queries have been highly optimized by adjusting the size of dedicated memory in WebGenesis. Indicatively, the insertion of data from the video analysis module - which is expected to produce the greatest volume of data - had been initially observed to cost approximately 30 seconds. However, after scaling up the memory by a factor of 3, the very same query execution was noticed to last around 3-4 seconds.

Audio Analysis

The ASR module is currently uses 200m cpu and 2048Mi memory. CMU Sphinx[3] is used as the base of ASR and specifically, Sphinx4, which is a pure Java speech recognition library. It provides a quick and easy API to convert the speech recordings into text with the help of CMU Sphinx acoustic models. In order to use CMU Sphinx4 in the project, we had to add the Sphinx4 libraries (namely sphinx4-core and sphinx4-data) to the dependencies of the project. Sphinx4 is available as a maven package in the Sonatype OSS repository, which is also included in the project's repositories.

[3] https://cmusphinx.github.io/

Visual Analysis

The visual analysis components have for the most part behaved as expected. We have showcased the relative work in proper manner though participation in conferences and submission of papers. Currently the visual analysis components are well inside the expected level of fulfilment of requirements for the 1st prototype.

Crisis Classification

Below is presented the performance evaluation of the Crisis-Classification component regarding the required computational time to perform the algorithmic steps for each pilot.

In general, the sequence of the algorithmic procedure is the following: (i) Data Acquisition, (ii) Danger Estimation, Crisis Level calculation, (iii) Messages generation. The relevant quantitative metrics regarding the performance of Crisis Classification can be seen in Table 1.

Table 1Performance of Crisis Classification per use case

Flood Early Warning component					
Time in sec.	**Step 1. Data Acquisition**	**Step 2. Calc. WL – Top.104**	**Step 3. Calc. PFLCL & Top.104**	**Total Time**	
Avg	1.349	21.0678	0.0086	**22.4254**	
Std	0.1656	2.3540	0.0062	**2.3104**	
Flood Real-Time Monitoring & Risk Assessment component					
Time in sec.	**Step 1. Data Acquisition**	**Steps 2 Calc. WL – Top.104**	**Step 3. Calc. OFLCL & Top.104**	**Total Time**	
Avg	1.5426	0.7522	0.0018	**2.2966**	
Std	0.0761	0.0234	0.0011	**0.0916**	
Fire Early Warning component					
Time in sec.	**Step 1. Data Acquisition**	**Step 2. Estimate Fire Danger**	**Step 3. Estimate PFRCL**	**Step 4. Create & forward Topics**	**Total Time**
Avg	24.3396	0.0566	0.0752	0.0312	**24.5026**
Std	2.3440	0.0193	0.0305	0.0170	**2.3312**
Heatwave Early Warning component					
Time in sec.	**Step 1. Data Acquisition**	**Step 2. Estimate Fire Danger**	**Step 3. Estimate PHWCL**	**Step 4. Create & forward Topics**	**Total Time**
Avg	0.0003	2.9345	0.6283	0.0003	**3.5635**
Std	0.0001	0.1389	0.0446	0.0004	**0.1195**

Communication Bus

There are many scalability dimensions in the communication bus and mechanisms used which affect favorably the overall performance and throughput of the system. Currently the deployed system can comfortably accommodate the anticipated load of the beAWARE pilots, and has the capacity to support a higher load, given the current installation and deployment. In addition, there are various scalability factors, affecting performance that can be applied when the system load gets considerably larger.

- **Number of servers/brokers**
  For scalability and fault tolerance the communication bus can run with a number of servers acting as cooperating message brokers; cooperating for providing continuous service. Running multiple brokers means that for each partition there shall be a single broker acting as the designated leader, and a list of brokers acting as replicas. Currently beAWARE's communication bus is deployed over 5 brokers. The number of brokers can be scaled up based on need, but for the foreseeable future there is no expectation that the platform would require more brokers to be deployed. Replication factor for beAWARE's topic is 3, thus we ensure that sent messages are available in at least 3 brokers, such that the platform can continue normal operations even in the unlikely event of two brokers being unavailable simultaneously.
- **Number of topics**
  The entire messages space sent and received by the communication bus is divided into topics. Each topic forms a separate unit to which messages can be sent and through which messages can be consumed. Message publishers designate every message they send to a single topic, and message consumers declare a set of topics which are of interest to it. In such a manner the entire spectrum can be divided between different processes distributed over different nodes and have the overall load to be distributed between different clients and different broker entities. Currently in the communication bus there are 32 topics declared; most of them are used operationally and some are used only for testing purposes.
- **Number of partitions**
  The partition is the unit of total order within the communication bus. Every topic is divided into 1 or more partitions; messages order is guaranteed within a partition. Clients consume messages from the partition head while messages are added to the tail. The partition is also the unit of division between different brokers. Thus, each partition is owned by a specific broker / server. The number of partitions of a topic can be scaled up and down based on need. Messages are kept in memory, thus dividing a topic to multiple partitions enables handling large topics regardless of the amount of memory in a single server.
- **Consumer groups**
  Consumer groups enhance the scalability of the messaging system, by declaring a group of cooperating consumers, and having the system ensure that each message will reach one member of each consumer group. Within a group each member is assigned a fair share of partitions to receive messages from. The combined features of partitions and consumer groups contribute to the overall system scalability and load balancing.

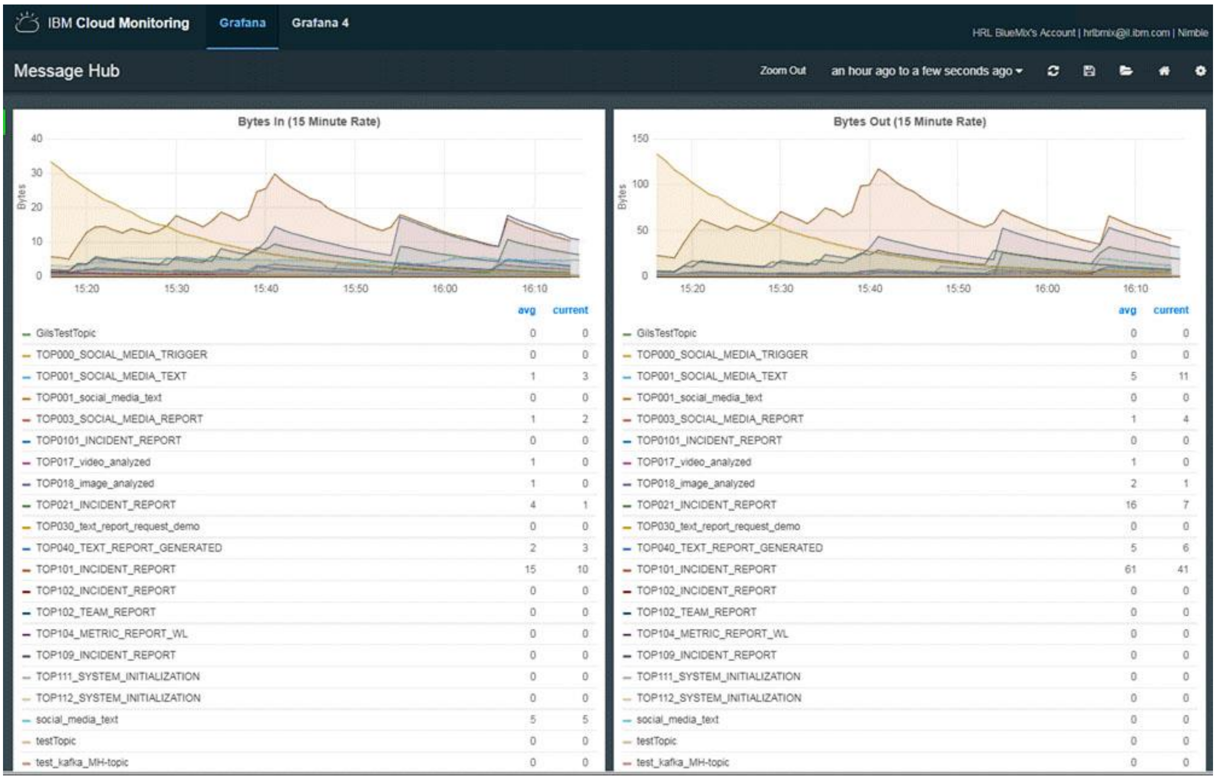


Figure 21: Data Traffic through the Communication bus during a typical technical testing.

5 Conclusions and Future Plans

This deliverable is a demonstrator of the integration of the first prototype of the beAWARE platform that was realized during MS3. This document presents the development status of each component, the integration approach the methodology for evaluation and validation of the platform. The deliverable also presents the operational use case scenarios that were implemented in order to validate the functionality of the platform and the different modules.

According to the MS3, the first prototype of the beAWARE system includes the completion of the first SW development cycle of the project. It includes the 1st version of beAWARE platform integrating: (ii) basic module of text analysis (iii) semantic representation and reasoning module; (iv) emergency report generation modules; (v) data source management framework prototype.

The demonstrator of this deliverable presented that the 1st prototype of the system included more functionalities of the system. During the integration procedure, problems and issues solved during continues evaluation of the system based always on the needs of the use cases and the end users.

The first prototype of the system will be tested in a pilot use case and the evaluations will help to develop the second prototype of the system with enhanced capabilities and more features.

6 Appendix 1: System Functionalities - first prototype / final prototype

beAWARE components	First Prototype (Review)	Final Prototype
KB	✓ KB fully implemented ✓ 1st version of the ontology available	• No additional features planned • Ontology changes as needed
SensorThing server	✓ SensorThings-Server fully implemented and deployed. ✓ Basic visualization available in WebGenesis	• Event detection • Threshold detection • Other events need to be discussed and aligned with crisis classification
Crisis Classification	✓ Connect with SensorThing-Server to receive metereological, hydrological data and forecasting data from AMICO system ✓ Implement the Early Warning System. Develop forecasting methods to evaluate the level of severity in each user case (flood, fire and heatwave) ✓ Forward the outcome of the analysis to PSAP (Topic 104 Metric Report)	• Receive data from heterogeneous sources (e.g. photos, text messages, IoT, sensors data, social media) • Analyze the data so as to forecast an upcoming crisis event (full functionality of the Early Warning System) • Risk assessment services & real-time monitoring system to identify crisis event's severity level
Image analysis	✓ Detection of flood event ✓ Detection of fire event ✓ Traffic level classification ✓ Object detection (People, Vehicles)	• Localization of flood event • Localization of fire event • Object risk assessment based on localization • Drone photos analysis • Optimization
Video analysis	✓ Detection of flood event ✓ Detection of fire event ✓ Traffic level classification ✓ Object detection and tracking through sequence of frames (People, Vehicles)	• Spatiotemporal localization of flood event • Spatiotemporal • localization of fire event • Object risk assessment based on localization • Drone footage analysis • Optimization
ASR	✓ All four language models up and running	• Fine tune language models (in order to improve recognition accuracy) • Automatic language identification
Social Media	✓ Crawling tweets for all predefined pilots and languages ✓ Visual and text classification (in order to estimate if a tweet is relevant or not) ✓ A dummy twitter report is sent to KB	• Improve text classification techniques • Produce Twitter reports that contain • Spatiotemporal clusters of tweets
Text analysis	✓ Extraction of of key events/incidents, objects & locations for the four languages	• extended coverage for the three domains for the four languages

beAWARE components	First Prototype (Review)	Final Prototype
	✓ Manually curated lexicons & grammars; coverage tailored to first prototype UCs	• Semantic abstraction over individual languages' particularities • Improve robustness and semantic coherency of extracted representations
Report generator	✓ First version of linguistic generation for the four languages ✓ Basic content selection ✓ Coverage tailored to first prototype UCs	• Improve domain coverage for the four languages • Enhance lexical variability & syntactic structures complexity • Temporally-aware planning and report generation
Mobile application	✓ Sent report with attached media files	• Receive warning from authority • User/Role Management • Sent current position • Receive/Handle Task from Authority Multilingual support
PSAP	✓ Displays metrics on the map. ✓ Displays teams on the map. ✓ Displays incidents on the map. ✓ Sends public alerts from a fixed list of texts.	• 2nd version of the Dashboard • Metrics Visualization • Operational Management • Incident Management • Optimization