beAWARE

Enhancing decision support and management services in extreme weather climate events

700475

D7.8

Final beAWARE System

Dissemination level:	Public
Contractual date of delivery:	Month 34, 30 October 2019
Actual date of delivery:	Month 35, 30 November 2019
Workpackage:	WP 7: System development, integration and evaluation
Task:	T7.3 - Overall Technical Testing of beAWARE platform
Type:	Demonstrator
Approval Status:	Working Version
Version:	V0.6
Number of pages:	69
Filename:	D7.8_Final_beAWARE_system_2019-11-30_v0.6

Abstract

This document describes the implementation of each component and the integration stage of the last and final version of the beAWARE platform. It provides an overview of the integration status of the different components the hosting infrastructure and the improvements that were performed since the second version of the platform, described in D7.5.

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

Co-funded by the European Union

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

History

Version	Date	Reason	Revised by
V0.1	26/09/2019	Document initiation and assignments distribution	CERTH
V0.2	07/10/2019	First version of content and elicitation for contributions	IBM
V0.3	23/10/2019	Incorporate contributions from IOSB	IBM
V0.4	29/10/2019	Updating PSAP information	MSIL
V0.5	15/11/2019	Incorporate contributions from CERTH	IBM
V0.6	28/11/2019	Final version for submission	CERTH

Author list

Organisation	Name	Contact Information
CERTH	Ilias Koulalis	iliask@iti.gr
IBM	Benny Mandler	mandler@il.ibm.com
IOSB	Philipp Hertweck	philipp.hertweck@iosb.fraunhofer.de
MSIL	Itay Koren	itay.koren@motorolasolutions.com
CERTH	Stelios Andreadis	andreadisst@iti.gr
CERTH	Krestenitis Marios	mikrestenitis@iti.gr
CERTH	George Orfanidis	g.orfanidis@iti.gr
CERTH	Anastasios Karakostas	akarakos@iti.gr

Executive Summary

The D7.8 of the beAWARE platform provides a technical overview of the development and integration of the final system, explains the improvements carried out with respect to the previous two prototypes.

The objectives of this report are:

- The presentation of the final capabilities of the beAWARE platform
- The demonstration of the system based on a specific and complete Use Case

The deliverable summarises the final status of all the services of the system.

Abbreviations and Acronyms

API	Application Programming Interface
ASR	Automatic Speech Recognition
CI	Continuous Integration
DA	Drones Analysis
DTr	dynamic texture recognition
DTstL	Dynamic Texture spatio-temporal localization
EFAS	European Flood Awareness System
FIFO	First In First Out
GPU	Graphics Processing Unit
GUI	Graphical User Interface
JSON	JavaScript Object Notation
KB	Knowledge Base
KBR	Knowledge Base Repository
KBS	Knowledge Base Service
K8s	Kubernetes
MS	Milestone
M2M	Machine-to-machine
MTA	Multilingual Text Analyser
MRG	Multilingual Report Generator
ObjD	Object detection
PSAP	Public-safety answering point
P2	Prototype 2
REST	Representational State Transfer
SMA	Social Media Analysis
VRS	Visual River Sensing
WP	Work Package

Table of Contents

List of Figures

List of Tables

1 Introduction

The main goal of this document is to provide an overview of the final version of the beAWARE platform. The presented platform state builds upon prior WP7 activities and deliverables, starting from the architecture document (D7.2), followed by specification of the state and integration of the evolving platform (D7.3 & D7.5), while taking into consideration the evaluation of prior platform versions (D7.4 &D7.6). Thus, this document was devised and written with the requirements document (D2.1 - Use cases and initial user requirements) and the architecture documents (D7.2 - System requirements and architecture) in mind.

The document is structured in 7 sections: Section 2 provides a high-level overview of the system architecture. Section 3 provides details on the individual components comprising the system. Section 4 details the integration status and the interplay between the various components. Section 5 provides details on the physical cloud based deployment of the system. Section 6 provides links providing more information including videos of the final pilot. Finally, Section 7 summarizes the conclusions.

Overall, during the development process, many challenges like integration between different technologies, component interconnection, shared usage for some functionalities, etc. were tackled. Many technical issues regarding dependencies between services and 3rd party libraries were addressed, in order to obtain a stable and coherent system.

The early adoption of a microservices based architecture, with a clear manner of interaction among components based on a communication bus, all deployed and orchestrated via a Kubernetes cluster, made the daunting task of producing a coherent platform out of many individual components, designed and developed in parallel, across may partners in many countries, much easier.

1.1 Final Integration Overview

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

During the first year of the project an accelerated task of requirements gathering and architectural specification was carried out. The process took off by gathering requirements, from end users and from the technical partners. These requirements in turn were transformed into system level requirements. These requirements fed the architecture which determined the different components and the interaction among components required to abide by the requirements. The results of this work were described in D7.2.

In parallel, during the first year of the project the foundation for collaboration among the partners and components were laid out. First, via the establishment of an easy to use CI / CD toolchain, and second by putting in place the infrastructure on which components will be deployed, and common platform services which will be used by the different components for their own needs (such as storage), and for interaction with additional components.

In addition, during the first year of the project, an initial operational version of the prototype was established, enabling a fast and easy integration framework including dummy services and services with limited functionality. This process laid the foundation for the individual advances in each component (from dummy to full fledged capabilities), and in turn established the modes of interaction across components and services.

After the establishment of the initial prototype, the development and integration of components proceeded in coordinating waves, each one adding more functionality for established components. That process enabled gradually to support more elaborated requirements which were demonstrated through the three pilots.

The development and integration advanced according to plan anchored at the demonstrator pilots. The second year of the project was led by the establishment of the second prototype, driven by the flood scenario use case; driving requirements to be fulfilled by individual components as well as the system as a whole. The third year was dominated by the final system prototype, demonstrated in a fire scenario. Once again that led to the introduction of a new set of capabilities from individual components and from the system as a whole via the communication and collaboration between different components.

This deliverable explains the status of the system in terms of repositories, services, processes and workflows of the Final System.

The most important efforts were focused on providing the final capabilities of the different components required to fulfil the use case requirements (for example the recognition of people in wheelchairs and animals in images and videos, the automatic detection of the geographical coordinates of locations mentioned in audio and text messages, the two levels validator for the crowdsourcing information, the knowledge base interface for displaying risk maps and summarising the analysed data etc.).

1.2 Integration approach

The system development, as already described in Deliverable 7.2, follows an iterative approach, from an initial dummy prototype to the final version, and follows the following cycle:

1. Integration Prototype: Dummy end-to-end architecture; most service dummies in place, operating on test/mock data. This was presented at the end of the first year of the project as a part of MS2.

2. First Prototype: Using real services from WP3-WP6; First Prototype milestone (MS3), presented at M18.
3. Second Prototype: Using real services from WP3-WP6; Second Prototype milestone (MS4), presented at M24.
4. Final System: Complete beAWARE platform; Final System milestone (MS5), expected by the end of the project (M36), reported in this deliverable.

This approach promotes continuous iterations of development and testing throughout the software development cycle of the project. The objective is 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.

The integration and advances of components were demonstrated and discussed in weekly calls throughout the lifetime of the project. Key flows tested were driven by real use case scenarios.

According to the expected timeline, the last SW development cycle of the project. stands for the successful delivery of final system.

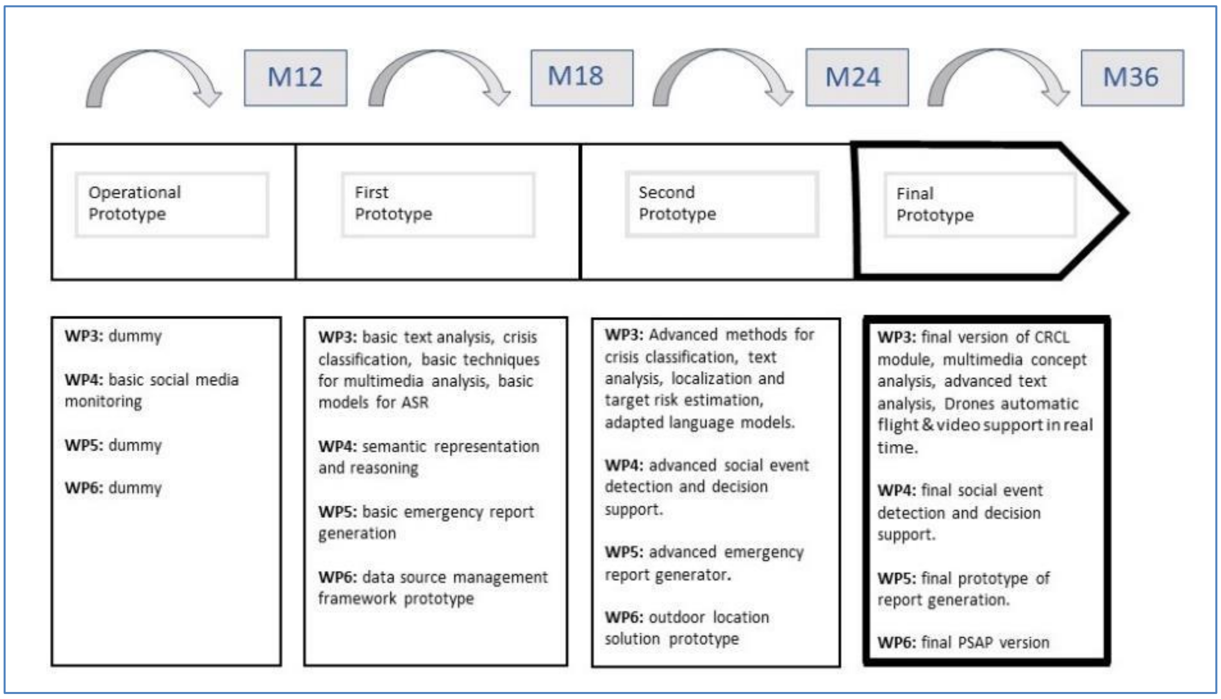

M12
M18
M24
M36
Operational Prototype
First Prototype
Second Prototype
Final Prototype
WP3: dummy
WP4: basic social media monitoring
WP5: dummy
WP6: dummy
WP3: basic text analysis, crisis classification, basic techniques for multimedia analysis, basic models for ASR
WP4: semantic representation and reasoning
WP5: basic emergency report generation
WP6: data source management framework prototype
WP3: Advanced methods for crisis classification, text analysis, localization and target risk estimation, adapted language models.
WP4: advanced social event detection and decision support.
WP5: advanced emergency report generator.
WP6: outdoor location solution prototype
WP3: final version of CRCL module, multimedia concept analysis, advanced text analysis, Drones automatic flight & video support in real time.
WP4: final social event detection and decision support.
WP5: final prototype of report generation.
WP6: final PSAP version

Figure 1: beAWARE walking skeleton

2 Final System Architecture

2.1 Global view

The global architecture of the beAWARE platform has been discussed at length in D7.1, D7.2, D7.3, D7.4 and D7.5.

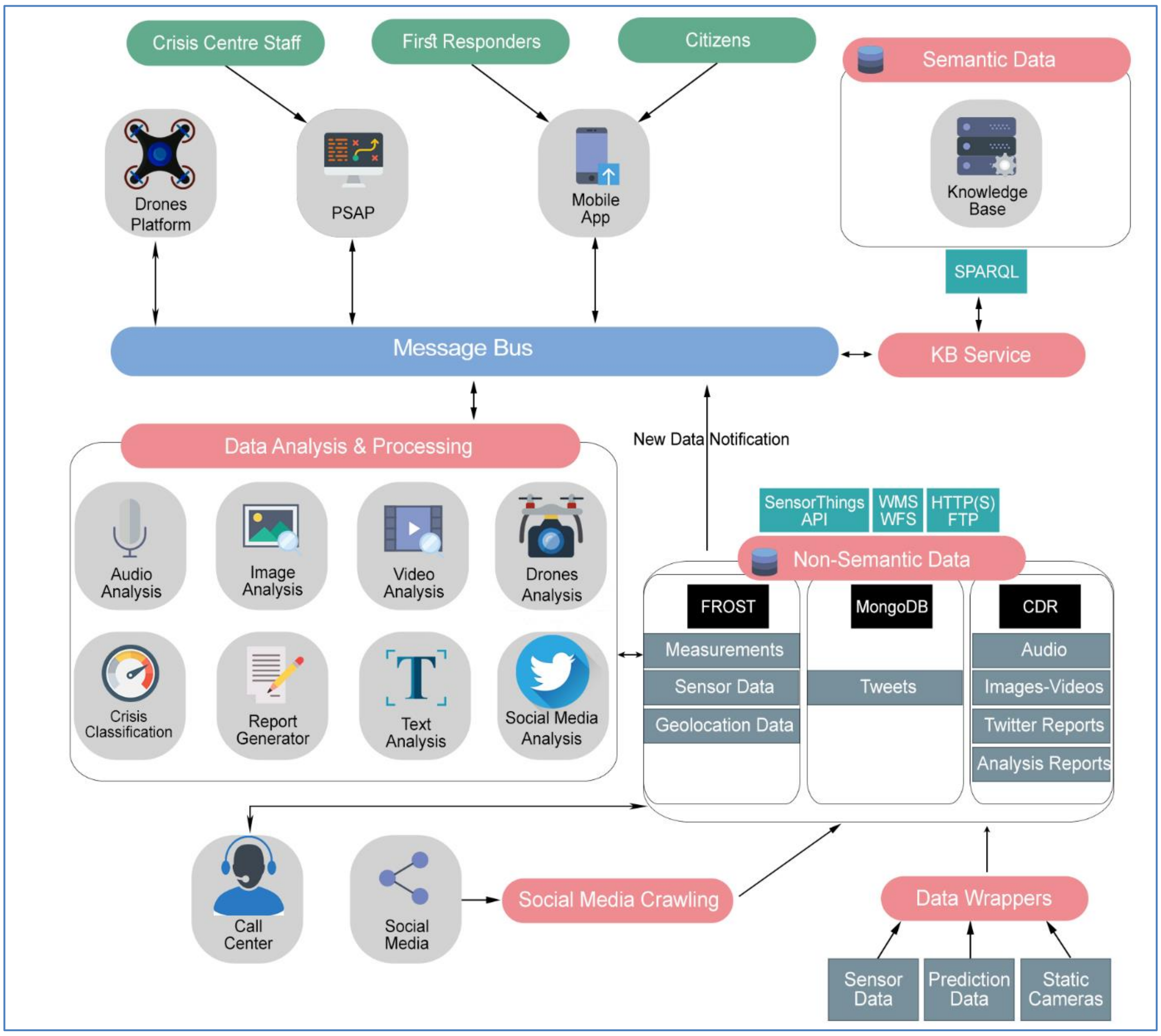


Figure 2: Architectural high-level view

The beAWARE architecture (see Figure 2) encompasses 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.

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) , providing the operational picture to the authorities.

2.2 Status of the previous Prototype (Second Prototype)

In D7.6, the status of the Second Prototype was explained. A quick summary of the elements that were part of it is provided below:

- The system was hosted on a powerful cloud infrastructure. The cloud infrastructure is composed of a highly available and scalable Kubernetes cluster on which the system components are deployed and orchestrated. In addition, system level functionalities, such as storage and messaging are provided by IBM cloud services, ensuring that the required performance indicators are met, such as throughput and latency.
- All the main services were in an advanced state of implementation.
- The different services were integrated into the platform, i.e. interacting with each other to perform complex and aggregated tasks, such as ingesting an image to the platform, have it analysed, asses its importance and relevance, produce text describing the findings, and presenting it in the map and incidents manager of the PSAP.
- The three Use Cases applications were aligned with the interaction of the available services. Weekly status and integration tests were focused on use case provided scenarios. The final version was prepared with the Valencia fire scenario in mind.

3 Objectives and Status of the Final System

The Final System (FS) represents complete integration of all services and specific components. The architecture is roughly made up of the following layers:

1. Business layer, containing the components that perform the actual platform-specific capabilities, such as the analysis modules.
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 for their own needs and for collaborating with other components.
3. External facing layer, including the end-user's applications and PSAP (Public-safety answering point) modules, interacting with people and entities outside the platform (end-users of the platform). This includes the Ingestion layer, containing mechanisms and channels through which data is brought into the platform.

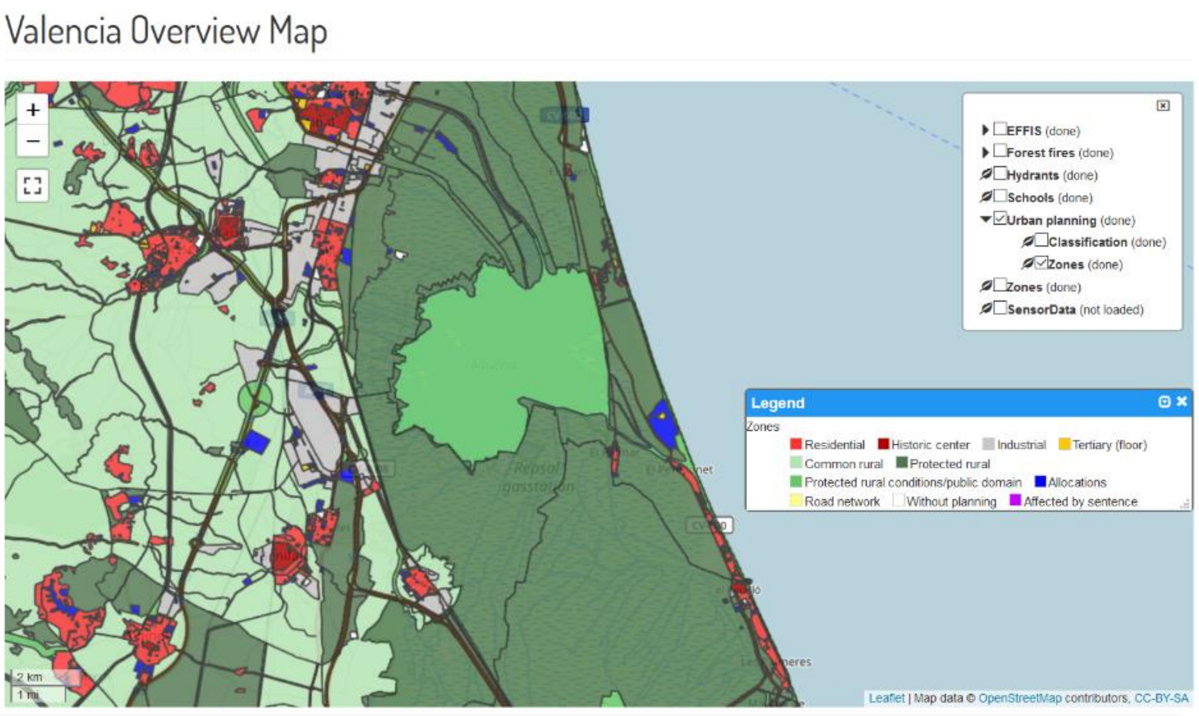
3.1 Business Services

3.1.1 Knowledge Base (KB)

The Knowledge Base is used to store the semantic data of the beAWARE platform. The development of the semantic representation, the beAWARE ontology, was already finished for the second prototype. For the final system, a few, more specific concepts (e.g. the concept of a Wheelchair user) was included to be able to represent all the aspects, addressed by the beAWARE platform.

The knowledge base also contains an interface for displaying risk maps. For the final system new map layers have been integrated. Those layers include:

- **EFFIS:** The European Forest Fire Information System (EFFIS) provides forecasts for specific forest fire related metrics like the Fire Weather Index (FWI), which are available on the risk maps.
- **Historical fires:** The historical fires (between 1993 and 2015) in the area of the pilot
- **Hydrants** in the pilot area
- **Schools** in the pilot area
- **Urban Planning:** Information on how the area is used (see Figure 3)



-

Figure 3 Risk map of the pilot region

The knowledge base was extended by an input form, which allows to enter information into the system manually. This feature can be used, e.g. when citizens or first responders report incidents not directly to the beAWARE platform via the mobile app. For example, if first responders inform the authorities via the radio, this can be inserted into the platform by using the provided forms. This input modality not only ensures that the data is available inside the semantic model but also triggers the internal data flow, to notify the other components about the newly available data.

To further analyse the semantic content, especially the incidents and how they are represented and clustered, an incident table and map are included in the knowledge base. This is explained in more detail in D4.3. In addition, the changes in the analysis dashboard are explained in this WP4 deliverable.

3.1.2 **Knowledge Base Service (KBS)**

The Knowledge Base Service (KBS) is a middleware service that handles storage, processing and retrieval of system data from the KB. Besides monitoring the message bus and storing/exchanging information, the KBS incorporates a semantic reasoning mechanism for semantically enriching the beAWARE KB with incoming information, and for inferring underlying knowledge and discovering inter-linkages between incidents during a crisis.

For the final version of the platform some additional work has been carried out for updating the previous version of the service and making new functions available. The most notable additions are listed below:

- Additions related to Improvements of Analysis Components: New additions to the KB have been made to adapt it to improvements of the analysis components (Text Analysis, Social Media Analysis, Video, Image and Drones analysis). Among others this includes new subclasses such as the *WheelchairUser, Cat, Dog etc*. Further, new datatype properties have been added to the *IncidentReport* concept. The *isSpam* datatype property for example to be used to store the spam results from the social media analysis. More details can be found in the deliverable D4.3.
- The two-layer validation mechanism: A second validation layer (or Validator, VAL) is implemented within the Knowledge Base Service (KBS) working alongside with the Crisis Classification module (CRCL) to detect erroneous instances. The validation results are processed by the KBS and propagated to the PSAP when necessary.

3.1.3 **Social Media Monitoring**

The Social Media Monitoring task is achieved with two individual components: the Social Media Analysis (SMA) module and the Social Media Clustering (SMC) module.

The first module is responsible for collecting social media content that relates to the examined use cases. Twitter's Streaming API is used to find tweets that contain preselected keywords about floods, fires, and heatwave incidents in four different languages (i.e., English, Italian, Greek, and Spanish). To ensure the quality of the collected information and protect the system from malicious acts, a three-step filtering process is performed. The first step is a verification service that aims to detect tweets that share fake news and filter them out. The second step is checking tweets for irrelevant emoticons, e.g. messages with smileys or hearts most probably do not contain information that is important for a crisis. Finally, a textual and a visual classifier try to estimate whether the posts are relevant to the use cases. All details of a collected tweet are stored in a MongoDB database, along with the outcomes of the analyses. Every tweet that passes the three-step filtering is forwarded to the Knowledge Base Service (KBS) to populate a corresponding incident report and to the Multilingual Text Analyser (MTA) for further analysis.

Regarding the second module, SMC consumes from MTA the analysed texts that originate from SMA's collection and groups them into clusters. Spatial clustering is based on the locations detected by MTA and is performed on tweets consumed within one minute. SMC creates a summary for each estimated cluster, called Twitter Report, and forwards them to the KBS, where they are handled as incidents.

3.1.4 **Crisis Classification Module**

The main objective of the Crisis Classification module is to process the available forecasts from prediction models (weather, hydrological etc.) and data obtained from sensors, as well as other heterogeneous sources (multimedia and text analysis outcomes), to generate early warnings and real time alerts.

The Crisis Classification module consists of two components. The Early Warning component that provides alerts during the pre-emergency phase and the Real-Time Monitoring and Risk Assessment component that is activated during the emergency phase and is responsible for monitoring the evolution of the crisis.

In the 1st prototype of the project, several functionalities were already developed and integrated, including a flood, fire and heatwave Early Warning modules, which are able to timely notify stakeholders of the upcoming natural extreme event, as well as to estimate its level of crisis (severity level). Furthermore, during the crisis, functionalities able to track the evolution of the hazardous event have already been deployed and assessed the risk level of the crisis. These modalities rely on the forecasting data and/or observations obtained from sensors.

During the 2nd prototype further enhancements were accomplished:

- Enrich the risk assessment process in the pre-crisis phase. The *Early Warning* component can estimate the crisis level in a local scale (identifying small areas of interest) and global scale for the whole Region of Interest.
- Enrich the risk assessment process in the crisis phase. The Real-Time Monitoring and Risk Assessment component is able to estimate the risk level of the ongoing crisis event by employing a multi-level assessment process. Obtaining real-time sensing observations, the Sensor Fusion Module fuses them providing the Observed Crisis Level. Thereafter, the Decision Fusion Module consolidates the information acquired from the Sensor Fusion module (Observed Crisis Level) as well as from the outcome (Crisis Severity Level) of the analysis of additional beAWARE modalities, in order to provide a total risk assessment of the crisis event. It is worth to note, that the innovative aspect of this module is the engagement of citizens and first responders with the risk assessment process by the utilisation of the mobile app and inserting into the system useful and valuable data from the field. The obtained data is analysed and taken under consideration in assessing the crisis risk in a local level.

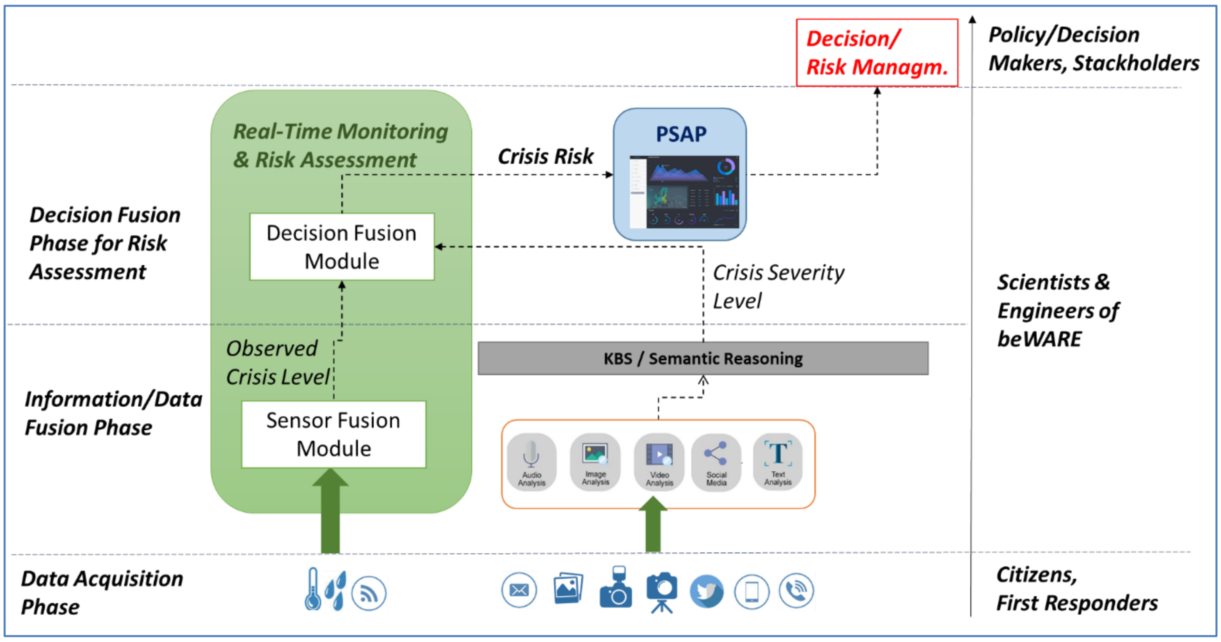


Figure 4: High-level architecture of Real-Time Monitoring and Risk Assessment component

- Integrate functionalities to enhance the interoperability with other numerical weather forecast systems, such as the HIRLAM model, hydrological forecast models, such as the AMICO model.
- Integrate functionalities to enhance Crisis Classification Module's interoperability with the European Flood Awareness System (EFAS), by employing impact/risk flood maps, such as the "EFAS rapid impact assessment" which provides a risk assessment of an extreme flood event.

Our efforts in the final version of the Crisis Classification module focused on the improvement of the readiness of the system to face a forest fire extreme event. Towards this direction, an enhanced risk assessment algorithm was deployed by taking into consideration the outcomes of the multimedia analysis (image/videos, text). Specifically, the following functionalities were developed and integrated into the existing Crisis Classification module:

- In the pre-Emergency phase, the *Early Warning* component interconnects with the EFFIS (European Forest Fire Information System), so as the estimations of the Fire Weather Index based on the Canadian Rating System be acquired, which indicates the fire danger of a forest.
- In the emergency phase, the *Real-Time Monitoring and Risk Assessment* component established the interconnections with the State Meteorological Agency (AEMET) open data API and DarkSky API to obtain "near" real-time weather observations in specific locations at the region of interest (Valencia). The weather observations for temperature, humidity, precipitation, wind speed, and direction were combined with short-term weather forecasts and delivered to beAWARE dashboard for illustration.

- A risk assessment algorithm has been developed to estimate the severity level of a cluster of incidents in an ongoing fire extreme event. Moreover, the Real-Time Monitoring and Risk Assessment component has utilised, dynamically, information with multimedia content, which was received from the citizens and first responders in the field. The results of the image/video analysis were fused in order to assess and classify the severity/crisis level of the evolving forest fire event.
- Finally, an interaction between CRCL component and the validator component has been established in order to forward, every time that it will be needed, the suitable and reliable real-time weather observations for a specific location of interest.

3.1.5 **Visual Analysis Module**

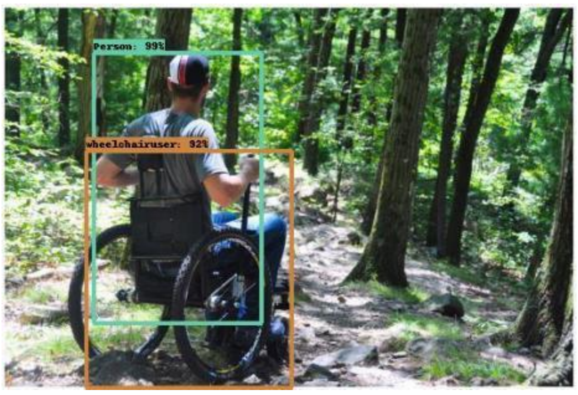
The Visual analysis module's main objective in the beAWARE project is concept extraction from visual content (images/videos), and it is supported by two separate components, namely IMAGE ANALYSIS and VIDEO ANALYSIS. In the 1st prototype and 2nd prototype of the project, several modules were already integrated, including a fire and flood detection system, as well as people and vehicle detection that may undergo danger during flood or fire emergencies.

For the final version further development has been carried out towards updating the previous versions of the modules and making new functions available as well. Together with the previous version's integrated modules an array of cutting-edge computer vision and machine learning techniques has been deployed:

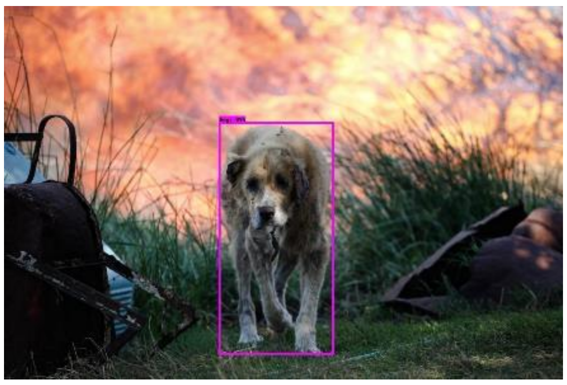
- (1st prototype) Image classification, so as to determine which images/video frames contain an emergent event or not (i.e. a fire of flood event).
- (1st prototype) Object Detection, so as to find people and vehicles that exist in the images/videos.
- (1st prototype) Object Tracking, so as to track targets throughout sequential video frames.
- (2nd prototype) Face Detection, so as to accurately count persons in crowded indoor shelters or places of relief.
- (2nd prototype) Dynamic texture localization so as to localize fire or flood dynamic textures in videos.
- (final prototype) Internal validator mechanism.
- (final prototype) Image classification, so as to detect smoke inside images and videos.
- (final prototype) Object detection and tracking, so as to find animals in danger, specifically cats and dogs.
- (final prototype) Object detection and tracking, so as to find wheelchair users that may be located close to an impacted area

All of the above techniques have been successfully integrated in the beAWARE final platform and were tested in three pilot cases.

From the system's point of view the Visual Analysis module 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 in 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, the internal validator mechanism chooses either to reject the content or to generate an analysed version of the media file which is then uploaded to the OBJECT STORAGE. Similarly, a JSON item is also uploaded and forwarded to the system which contains all the results in a suitable format that is readable by the KB service.



```
{
 "image": {
  "timestamp": "2019-11-21T16:11Z",
  "height": 359,
  "name": "71929%2F2915c06f-0e2b-458b-90c0-aaf300c52483",
  "width": 535,
  "crisis_type": "other",
  "crisis_level": "unknown",
  "target": [
   {
    "left": 85,
    "height": 250,
    "confidence": 0.998,
    "type": "Person",
    "width": 157,
    "risk": 0.0,
    "top": 46
   },
   {
    "left": 78,
    "height": 219,
    "confidence": 0.928,
    "type": "wheelchairuser",
    "width": 192,
    "risk": 0.0,
    "top": 137
   }
  ]
 }
}
```



```
{
 "image": {
  "timestamp": "2019-11-21T16:06:54Z",
  "height": 683,
  "name": "71868%2F3238c531-376b-4dd1-a8ad-40a9b4e731db",
  "width": 1024,
  "crisis_type": "fire",
  "crisis_level": "extreme",
  "target": [
   {
    "left": 394,
    "height": 421,
    "confidence": 0.993,
    "type": "Dog",
    "width": 209,
    "risk": 0.5,
    "top": 214
   }
  ]
 }
}
```

Figure 5: Example of the Visual Analysis results

Figure 5 shows examples of the analysed images (left) and the contents of the JSON files that are produced for each image (right).

3.1.6 Visual River Sensing

At the second prototype, Visual River Sensing (VRS) module was added, in order to perform visual analysis on footage from static surveillance cameras installed by the river. The aim of the module is to estimate the water level and generate alerts, in case of threshold exceeding. VRS streams video directly from the IP of the camera and creates a short video chunk in order to be processed. In case a rise in water level is detected, the video chunk is also forwarded to Object Detection and Tracking (ObD) module for traffic estimation, in order to obtain a better overview of the flood event. The water level estimation algorithm uses edge detection in order to detect a marker installed on the bank of the river. At the final prototype, the algorithm has been modified in order to improve estimation accuracy. The final algorithm uses average estimated values from multiple frames of a video chunk, in order to make it more robust, instead of using single frames. The resolution of the streamed video was also increased in order to improve accuracy.

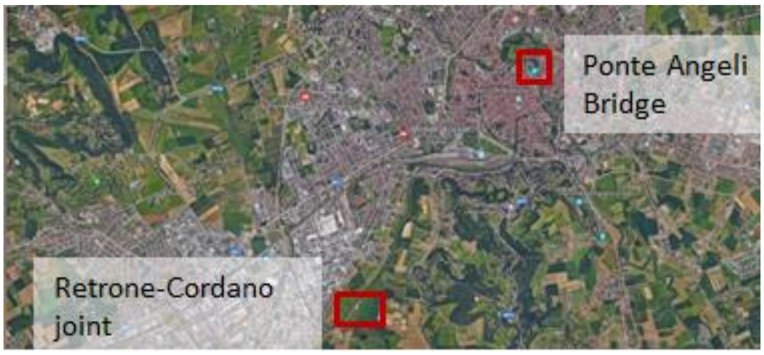


Figure 6: Location of the two surveillance cameras in Vicenza.

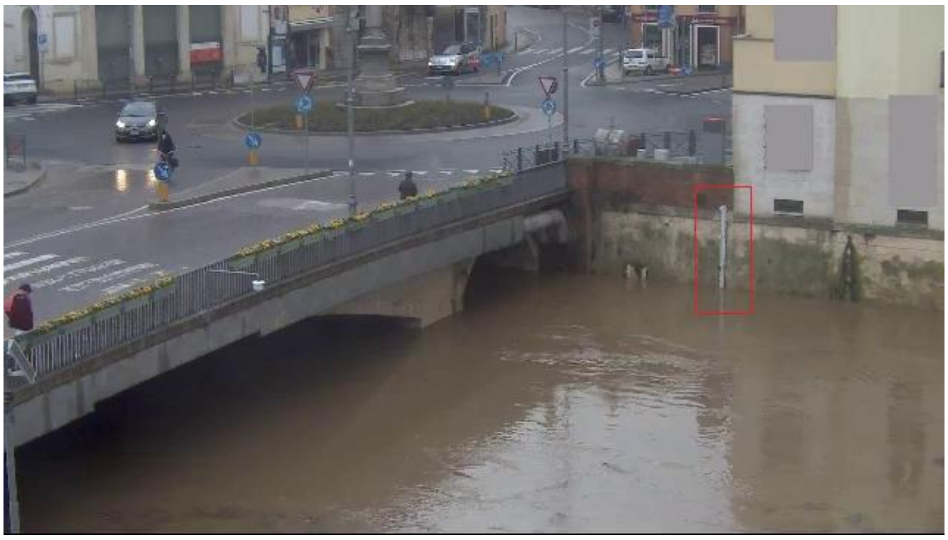


Figure 7: Captured frame for the static camera in Bacchiglione river (Angeli Bridge), with the water marker marked with a red box.

3.1.7 **Automatic speech recognition**

Automatic Speech Recognition (ASR) module is based on an open-source CMU platform and uses open-source acoustic and language models and dictionaries. At the first prototype, ASR was able to analyze audio messages from the mobile application in order to extract speech transcriptions in four different languages. The focus during the first prototype was on the improvement of the performance of the Greek model, by adapting the acoustic model on speech recordings provided by HRT and by tailoring the Greek dictionary. At the second prototype, the Italian model was likewise adapted, using case specific Italian audio recordings. Additionally, audio analysis was extended in order to include emergency phone calls too, apart from audio messages. For this reason, a call center solution was integrated to beAWARE, able to handle calls, record them and forward audio recordings to ASR. At the final prototype, the focus was on the Spanish language, by expanding the Spanish language model and dictionary in order to include more location names. The recognition was also enhanced by improving audio quality during audio conversion at the input of the module and by integrating more robust noise removal algorithms.

3.1.8 **Drones Platform**

The drones platform is intended as a service to connect providers of drones, drones services, and customers, to easily configure, launch, and monitor drone related activities. The essence of the drones platform capabilities is the combination of route planning and autonomous dynamic piloting, with the provisioning of data collected by the drone making it available to corresponding analysis components, all deployed in a cognitive cloud based platform.

Drones can carry different types of equipment collecting data (such as cameras, sensors), providing corresponding analysis tools data from a different angle (aerial data), and may reach locations which are difficult for humans. Thus, autonomous flight capabilities combined with cognitive edge analytics lead to novel ways of bringing lifesaving impact. Building such a generic platform for drone operations needs to overcome various technical and scientific challenges. Managing, provisioning, storing and analyzing high volumes of data and dynamically changing the route based on insights extracted from this data is a critical success factor.

Drones are being used for commercial activities these days, mostly requiring a pilot per drone to be operating it. On board equipment stores information locally which needs to be downloaded and transmitted upon landing. Some autonomous flying capabilities do exist, but the route is pre-programmed before the drone takeoff.

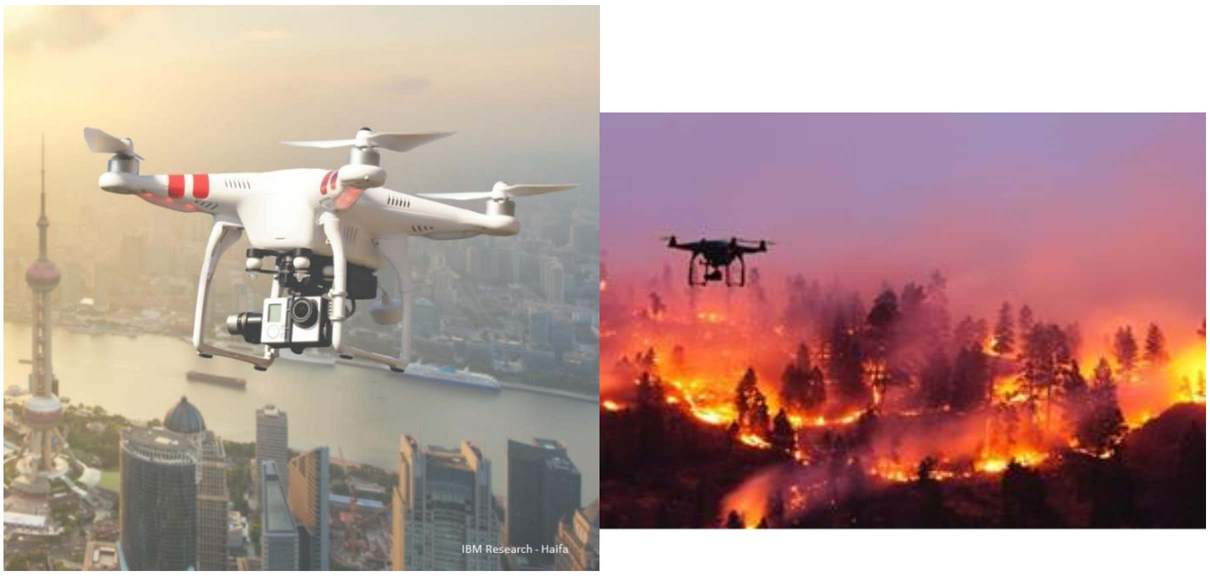
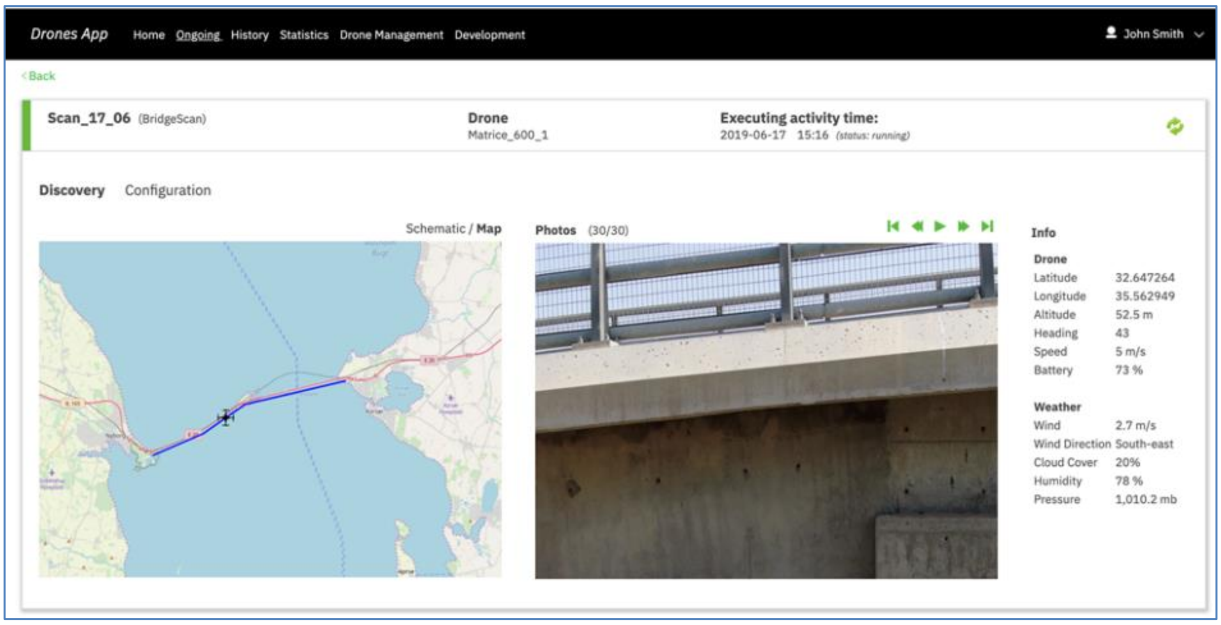


Figure 8: Drones in action

The developed drones platform provides the following main capabilities:

- Autonomous piloting – programmatic piloting without the need of a drones pilot. Generic commands are created to support agnostic control of drones; translated to a specific SDK based on the drone currently in action. Drone is controlled programmatically, thus not requiring a pilot per drone, and relieves related people's attention span to deal with the information and analysis coming from the drone, rather than with the piloting itself.
- Route planning – calculate flight parameters for optimal coverage of the area to be scanned. Taking into consideration the overall area, maximum flight time, height, and desired overlap area between consecutive scan paths (to ensure complete coverage of the area to be scanned).
- Dynamic route planning – The complete route is not necessarily fully loaded to the drone upon takeoff, but is rather created one step at a time, in real-time, while the drone is in the air. This enables incorporating analysis results in determining the continuation of the route. This capability makes the drones platform extremely useful as a general inspection can be performed, and once a specific incident of interest is spotted, the drone can be directed to take a closer look without the need to start a different mission, or to send a different drone.
- Control of on-board equipment – equipment such as cameras and possibly additional sensors can be controlled during the flight.
- Real-time transmission of acquired data – data produced and accumulated by on board equipment can be transmitted in real-time, for example to the analysis module and to the drones control dashboard.
  As can be seen in Figure 9, the dashboard provides a real-time view of the mission at

hand, providing information including planned the planned route, current location, imagery, and drone state.



Scan_17_06 (BridgeScan)

Drone
Matrice_600_1

Executing activity time:
2019-06-17 15:16 *(status: running)*

Info

Drone	
Latitude	32.647264
Longitude	35.562949
Altitude	52.5 m
Heading	43
Speed	5 m/s
Battery	73 %
Weather	
Wind	2.7 m/s
Wind Direction	South-east
Cloud Cover	20%
Humidity	78 %
Pressure	1,010.2 mb

Figure 9: Drones platform command and control dashboard

The drones platform consists of 3 components:

1. Drones server – On which the different service types are deployed, and specifically service instances are run. At run-time the server creates generic execution blocks for the drone based on the service, its specific configuration, and information gathered during the drone flight. It provides a drone vendor/model independent and agnostic SDK for developers to create new service types.
2. Drones edge device – Receives instructions from the server component and interacts with the drone via a specific SDK to fulfill the service. It received the captured data from the drone, and transmits it over in real-time to the server. In addition it enables the configuration and control of a specific service instance.
3. Platform dashboard – provides a real-time view of the service as its being carried out, including the current location of the drone, the past and following locations to be visited and the path taken. In addition it displays relevant analysis results and current state of the service.

Within beAWARE the drones platform acts as an additional content provider with a unique point of view, fast, in a safe manner, even in areas which are difficult to reach by humans and ground vehicles.

The drones platform establishes a bi-directional communication path with the beAWARE platform, based on the established collaboration tools in the platform. The drones platform captures media from on board equipment and passes it to beAWARE. At this stage the data consists of pictures and videos along with their respective metadata. The data is first uploaded to the cloud based object store, and once the data has been stored a message is sent over the cloud based message bus to alert all interested component that a new drone based information is available.

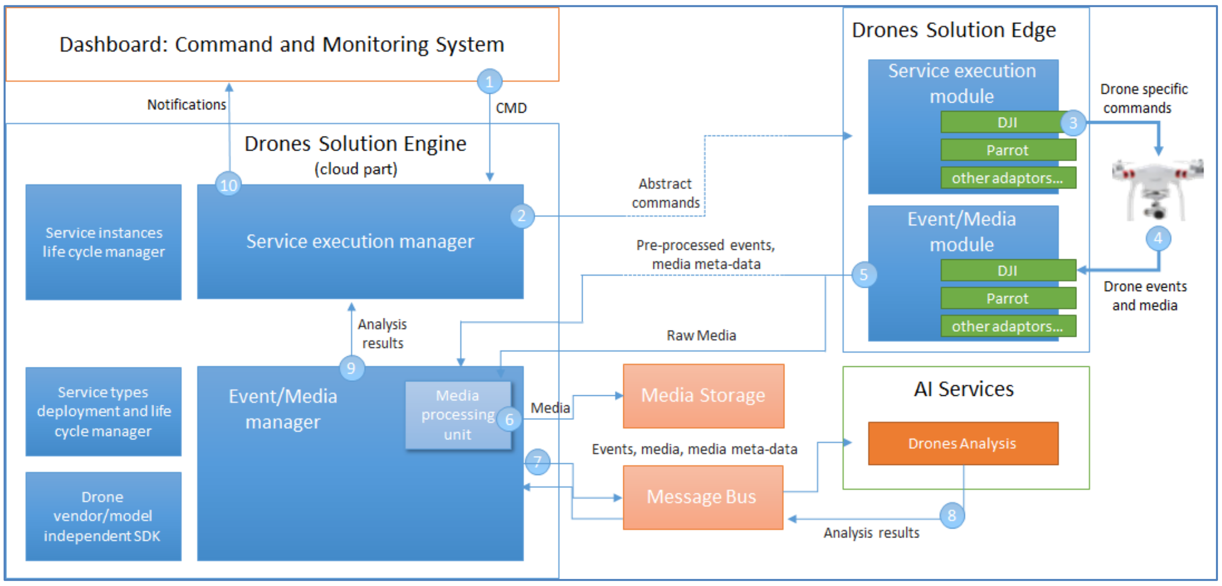


Figure 10: Drones platform architecture

Data and control flow of the drones platform:

1. A service type is selected and a specific service instance is configured and launched using the edge device.
2. The server part executes the service instance and produces the next execution block.
3. The edge device receives the execution block, which is drone agnostic, instantiates it for the specific drone in question, and passes the new commands to the drone via its specific SDK.
4. Drone events and media are transmitted to the edge device, and from there they are passed to the server component.
5. Server component stores media in the object store and published an appropriate message on a specific topic of the message bus
6. Media and analysis results are pushed by the server to the drones dashboard.

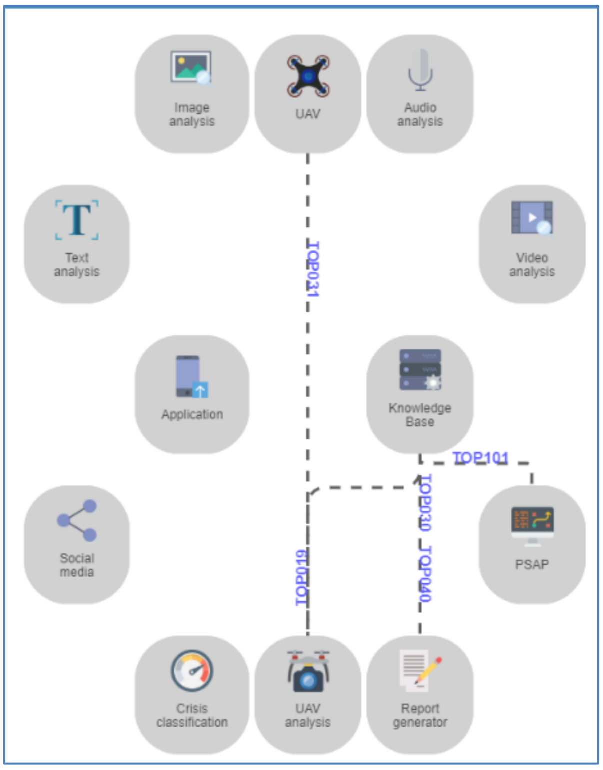


Figure 11: drones data processing path

For communicating with additional platform components, the drones platform uses a topic called TOP031_UAVP_MESSAGE, which includes information about the location, altitude, heading, and gimble pitch of the drone. In addition it includes the path in which the data can be retrieved. **Error! Reference source not found.** shows the path in which drones collected information flows through the different system components. The drones platform publishes a message describing a new piece of data being available. The Drones analysis module subscribes to notifications from this topic, and thus picks up the new data. When the analysis is done the drones analysis module published the results, which are picked up by the knowledge base, which in turn asks for a report to be generated before sending a message to which the PSAP subscribes.

The drones platform in turn also registers as a message consumer to obtain analysis results from the drones analysis module (TOP019_UAV_media_analyzed detailed in Table 1). That, along with the dynamic routing feature, enables the drone platform to set in real-time a new route to be followed by the drone, based on the analysis results. The message from the drones analysis module contains a link to the original media file, a link to the analyzed media file, and a link to the analysis results.

The drone can pick up this information. For example, in one of the scenarios the drone scanned an area which included a “person in danger”, analysis on the drone imagery was performed while the drone completed scanning the area. At that time the drones platform picked up the message from the drones analysis module, which identified the location of a person-in-danger, and a new execution block was created for the drone to fly to the identified location to take (and broadcast) a closer look, before returning to its landing place.

Table 1: Media analyzed message format

Field	Type	Description	Mandatory (Y/N)
media_original	String	Link to the original media file	Y
media_analyzed	String	Link to the analyzed media file	Y
media_analysis	String	Link to the JSON file that was produced from the analysis	Y
media_timestamp	Datetime	The timestamp of the media creation	Y
location		The location of the media creation (UAV), defined with latitude and longitude	Y
latitude	Decimal	The geographic latitude of the media	Y
longitude	Decimal	The geographic longitude of the media	Y
incidentID	String	The incidentID as received from the UAVP in topic 031	Y

The drones platform is connected to the beAWARE platform, according to the generic beAWARE architecture. The main means of interaction between the drones and the beAWARE platform are via the message bus and the object storage. The drones platform shall operate just like an additional data source, thus new data shall be stored in the object storage by the drones platform, and a corresponding notification shall be sent over the message bus. On the reverse path, the drones platform shall be subscribed to incoming messages via the message bus, for example to learn through it the results of data analysis, including potentially changes to the original route, such as flying to a specific location which was deemed important by the analysis or the PSAP components.

3.1.9 Drones Analysis module

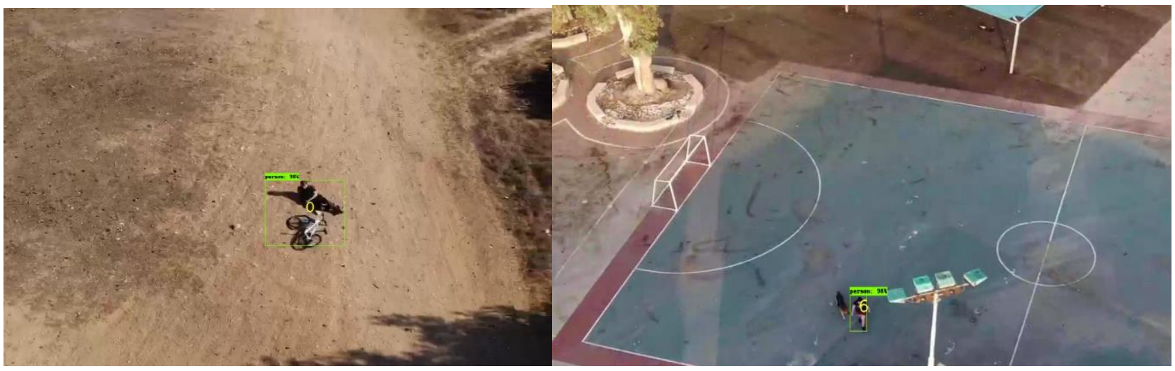


Figure 12: Examples of object detection algorithm[1].

At the second prototype, a new analysis module was added in order to support drone activities inside beAWARE. The module is called Drones Analysis (DA) and is based on deep-learning visual analysis techniques. It receives drone footage, performs visual analysis in order to detect people and track vehicles in danger and detect crises, such as flood, smoke and fire. Analysis results are communicated to PSAP and Drones Platform, along with analysed footage. Whereas, at the second prototype, the module was able to handle only sequences of images, at the final prototype DA has been extended in order to handle both image and video sequences. Some additional post processing analysis steps were added, in order to improve object detection and tracking and reject some random false positive detections. The classification model that is used, in order to detect if a frame contains a crisis incident has been updated, in order to include fire and smoke categories, apart from flood, that was supported at the 2nd prototype. Analysis tasks have been grouped in three categories. The first is 'Object Detection', during which DA detects and tracks people and vehicles. The second task

[1] In Figure 12 : Left image: detection of an injured biker ('Object Detection' task). Right image: detection of a person during evacuation task ('Evacuation' task).

is ‘Crisis Detection’, during which it detects flood, fire and smoke. Finally, during ‘Evacuation’ task, that was added at the final prototype in order to assist authorities at evacuation missions, DA detects if people are still present in an area under evacuation and creates relevant reports about the status of the mission. The distinguished tasks can be performed separately or in combination, according to an input variable.

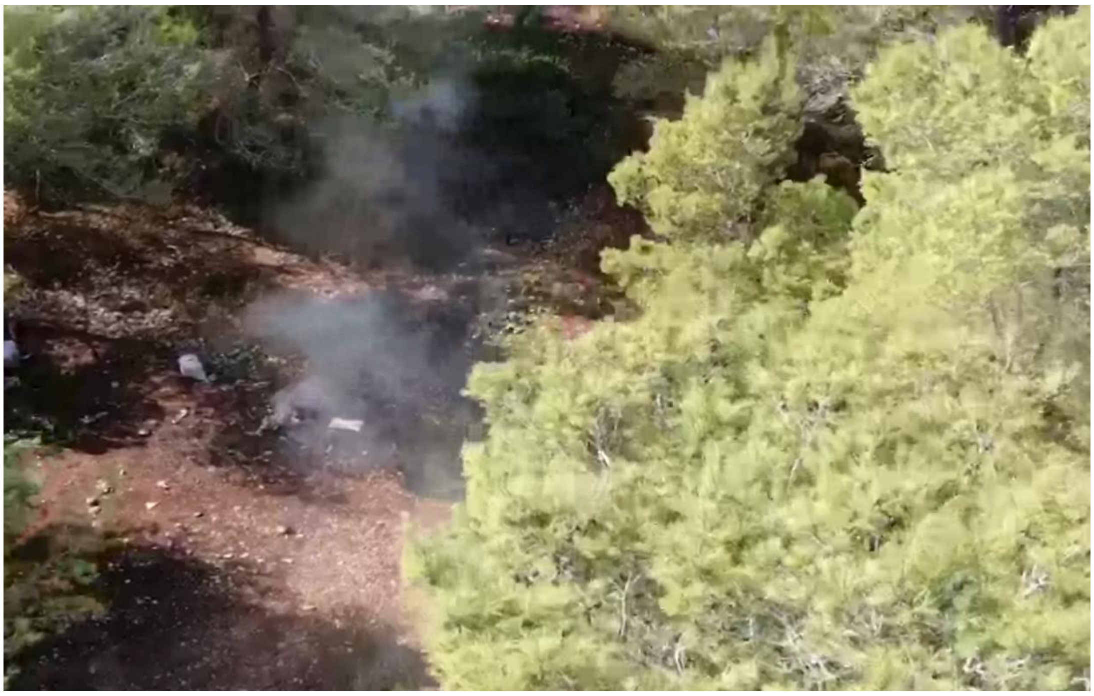


Figure 13: Example of an image classified as ‘smoke’ (‘Crisis Detection’ task).

3.1.10 **Multilingual Text Analyzer (MTA)**

The purpose of this component is to extract concepts and conceptual relations from natural language text obtained from social media or from the transcription of audio messages, in any of the project languages –English, Greek, Italian and Spanish. As with other analysis modules, MTA produces an ontology-ready representation where concepts and relations extracted are mapped to classes and properties in the project ontology.

The first implementation of this module was limited to the texts used in the 1st pilot. While it already included NLP tools with multilingual support, i.e. deep syntactic dependency parsing, the extraction of contents and their mapping to the ontology was done in an ad-hoc manner. Since then, a mixture of off-the-shelf IE tools and methods developed fully or partially for beAWARE have been incorporated into the MTA module. The version of MTA for the 2nd prototype already included several IE tools, namely NER, statistical term detection, geolocation and disambiguation. At this point the textual analysis was capable of identifying heat-related incidents in Greek and English texts, and flood-related incidents in Italian and English texts. In addition to incidents, MTA also detected various types of vulnerable objects -

persons, vehicles, etc.- impacted by the incident, and geolocated places associated with the event.

In the 3rd and final version we managed to expand the coverage of the module to go beyond the detect concepts and locations found in several well-known knowledge bases and geographical databases, beyond pre-scripted lists of concepts and locations.

The final version of the MTA module, the architecture of which is shown in Figure 14, has the following improvements with respect to the previous version:

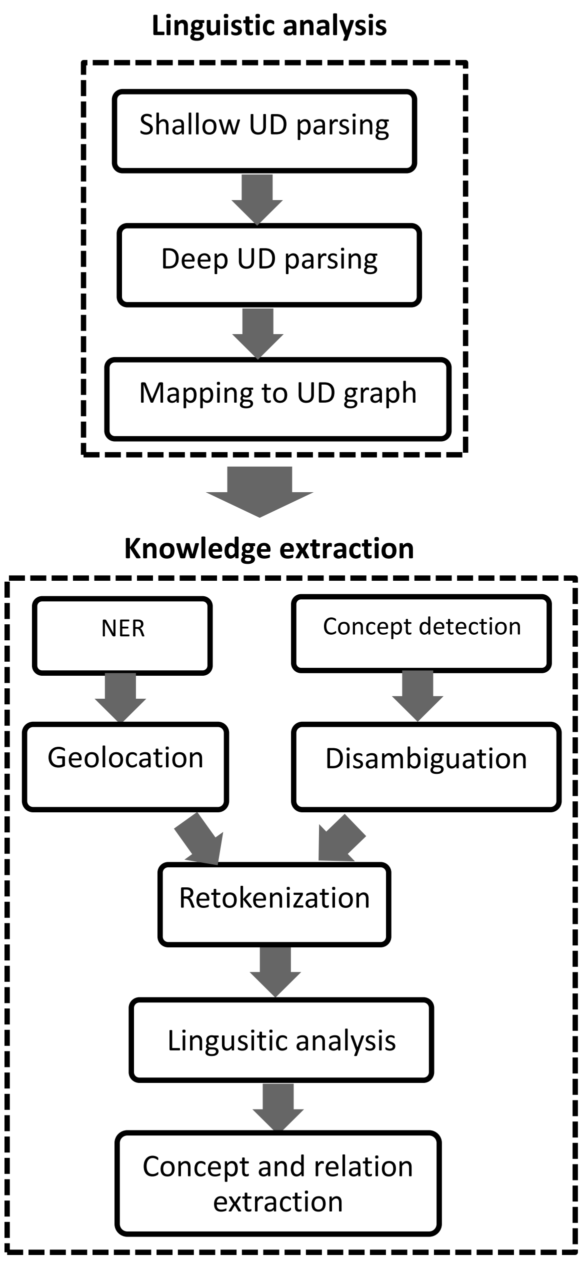


Figure 14: architecture of the final version of the MTA

1. Extended TA to detect fire-related incidents from Spanish and English texts.
2. Linguistic analysis is now fully based on Universal Dependencies (UD) for all languages, simplifying the architecture of the module.
3. Added mapping from deep UD trees to shallow semantic graphs to facilitate mapping to the project ontologies.
4. Improved the performance of the disambiguation and geolocation components.
5. Added a language-independent mapping from disambiguated BabelNet meanings to classes in the project ontology, the mapping being obtained using semi-automatic methods.
6. Designed and implemented a flexible language-independent relation extraction strategy based on the shallow semantic structures obtained from linguistic analysis and the results of the IE tools (disambiguated meanings and detected location names).
7. Taking advantage of improvements in the NLP and IE components, extended relation extraction to detect important states associated with events. These states indicate whether an incident is factual or hypothetical. e.g. if a text is mentioning an actual fire or just a risk of fire, and whether an incident is growing in magnitude, decreasing or stopped to be.
8. Reformulated the knowledge representation produced by the MTA to include both ontological constructs and references to BabelNet and Open Street Maps (OSM), as shown in Figure 15.

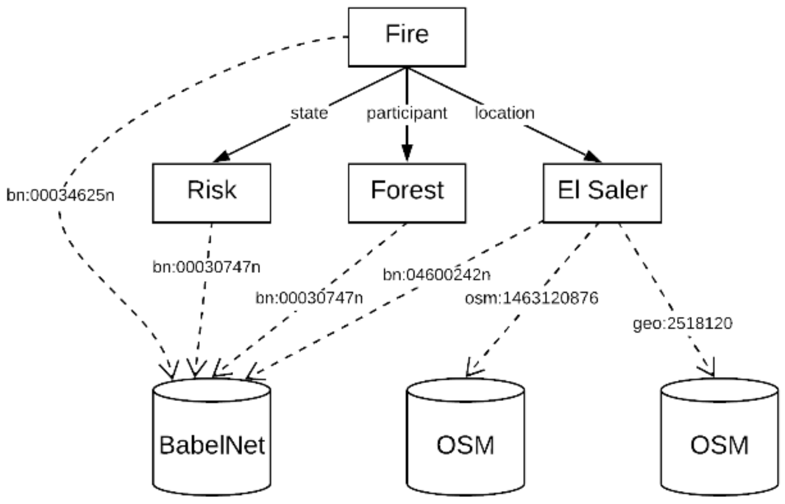


Figure 15: diagram of the analysis resulting from sentence "Danger of forest fire in El Saler"

3.1.11 **Multilingual Report Generator (MRG)**

Two types of reports were foreseen in beAWARE, short reports providing situational updates during ongoing crisis scenarios and wrap-up summaries providing an overview of a crisis. The 1st prototype of the MRG only supported the production of situational updates for heat-related incidents and their impacted objects and locations in Greek and English. A simple incident selection strategy was followed to avoid redundancies and repetitions across multiple reports. Surface realization used a mixture of dictionaries, rule-based and statistical tools to map the information received from the KBS to grammatical and fluent natural language.

For the 2nd prototype MRG was extended to produce situational updates in Italian and English about flood-related incidents. In a move that mirrored changes in the MTA, the linguistic generation component was updated to start from UD. Since this constitutes a common starting representation for all languages, it facilitates the mapping from ontological representations of incidents received from the KBS.

In the last stage of the project efforts have been focused on producing wrap-up summaries, in addition to extend MRG to produce situational update reports in Spanish and English about fire-related incidents. As described in detail in D5.3, a new content selection strategy and linguistic aggregation methods have been designed to select, groups and order incidents detected by the system during an emergency scenario based on their time, location, type of event and type of impacted objects. This results in a chronologically-structured summary where similar incidents are communicated in the same sentence and using syntactical and lexical devices to reduce redundancy and repetition.

In addition, MRG has also been extended to generate new information about states produced by the MTA, thus providing more useful incident reports, e.g. "Risk of fire is reported", "Strong winds in El Saler ", etc.

3.2 Internal Services

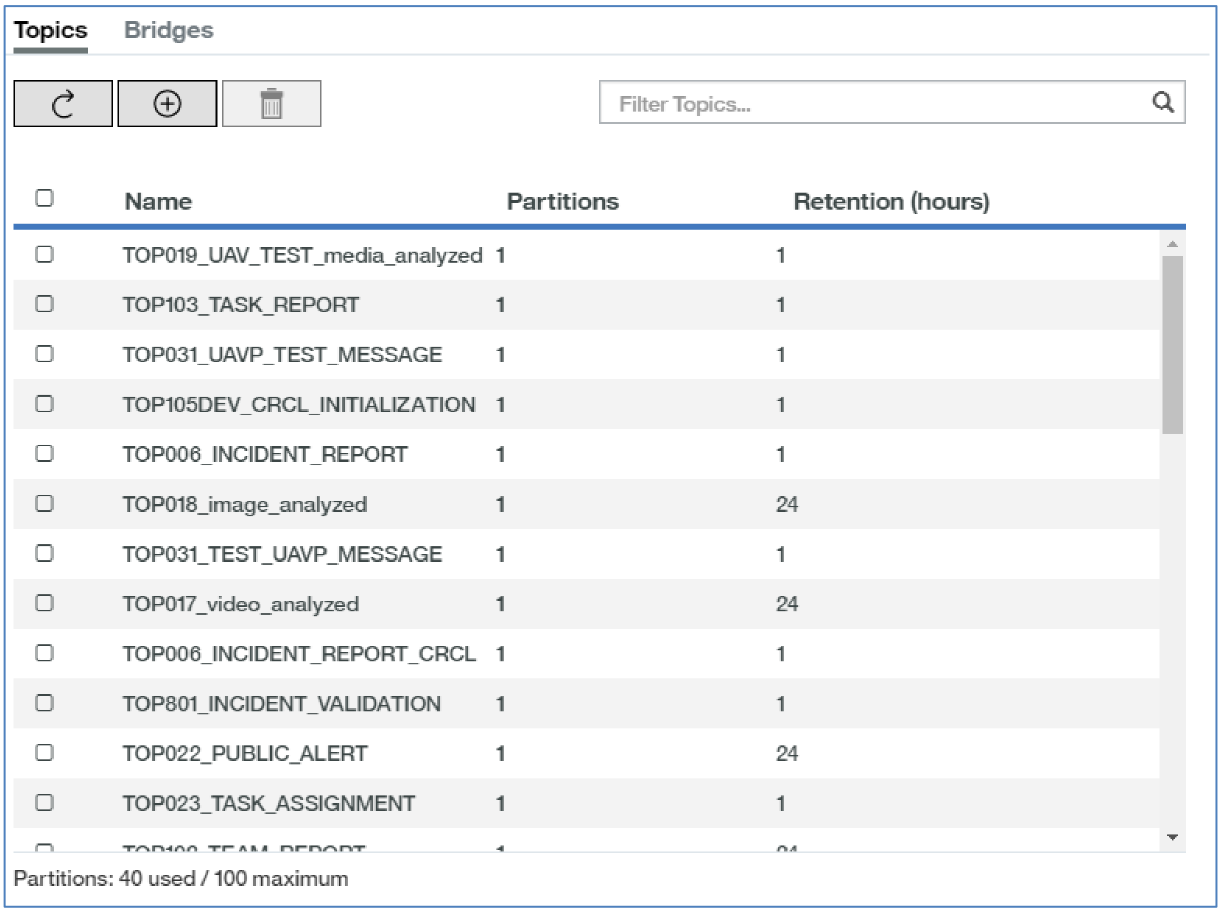
This layer handles services that are used internally for data storage and communication between components. The services within this layer have been updated to accommodate the new requirements and the new modules integrated into the system.

3.2.1 Communication Bus

The main purpose of the communication / message bus component is to provide generic communication capabilities among different beAWARE components and participants. It is used to send messages and notification among components and to share information among various entities. The dominant paradigm is the publish/ subscribe pattern leading to event-based communication among collaborating partners by registering interest in particular events. Using this paradigm producers and consumers do not have to be aware of each other and need only to agree on the topic via which they are going to communicate, and the message format of the agreed upon topic.

This component enables the distributed collaboration among different micro-services, with a minimal amount of synchronization required. The dominant flow is for a micro-service having a new piece of data (such as a new image has been uploaded) to publish the new event. All the micro-services interested in that specific kind of events shall be informed of the occurrence through the message bus and will be able to act accordingly (for example access the image via a provided link and perform the corresponding analysis).

Figure 16 depicts a portion of the cloud based dashboard of the message bus used by the platform, specifically depicting a sub-set of the topics being used by the platform.



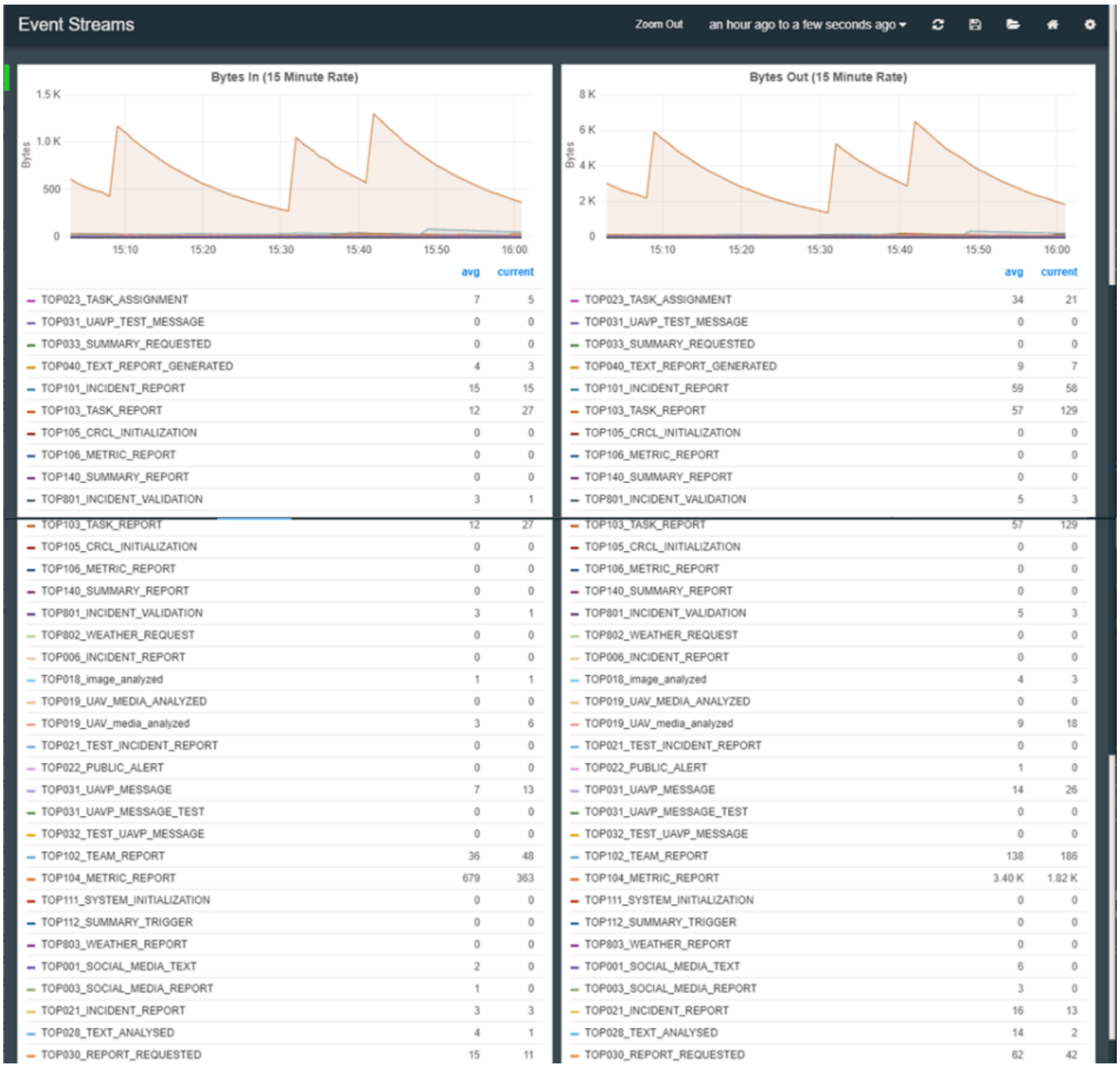
☐	Name	Partitions	Retention (hours)
☐	TOP019_UAV_TEST_media_analyzed	1	1
☐	TOP103_TASK_REPORT	1	1
☐	TOP031_UAVP_TEST_MESSAGE	1	1
☐	TOP105DEV_CRCL_INITIALIZATION	1	1
☐	TOP006_INCIDENT_REPORT	1	1
☐	TOP018_image_analyzed	1	24
☐	TOP031_TEST_UAVP_MESSAGE	1	1
☐	TOP017_video_analyzed	1	24
☐	TOP006_INCIDENT_REPORT_CRCL	1	1
☐	TOP801_INCIDENT_VALIDATION	1	1
☐	TOP022_PUBLIC_ALERT	1	24
☐	TOP023_TASK_ASSIGNMENT	1	1

Partitions: 40 used / 100 maximum

Figure 16: Communication bus dashboard

In Figure 17 we can see a portion of the message bus activity during one of the testing sessions performed, simulating the fire scenario.

Finally, Figure 18 depicts the messages common header, which is adopted by all platform topics. In addition each topic adds its specific payload.



Bytes In (15 Minute Rate)

	avg	current
TOP023_TASK_ASSIGNMENT	7	5
TOP031_UAVP_TEST_MESSAGE	0	0
TOP033_SUMMARY_REQUESTED	0	0
TOP040_TEXT_REPORT_GENERATED	4	3
TOP101_INCIDENT_REPORT	15	15
TOP103_TASK_REPORT	12	27
TOP105_CRCL_INITIALIZATION	0	0
TOP106_METRIC_REPORT	0	0
TOP140_SUMMARY_REPORT	0	0
TOP801_INCIDENT_VALIDATION	3	1
TOP103_TASK_REPORT	12	27
TOP105_CRCL_INITIALIZATION	0	0
TOP106_METRIC_REPORT	0	0
TOP140_SUMMARY_REPORT	0	0
TOP801_INCIDENT_VALIDATION	3	1
TOP802_WEATHER_REQUEST	0	0
TOP006_INCIDENT_REPORT	0	0
TOP018_image_analyzed	1	1
TOP019_UAV_MEDIA_ANALYZED	0	0
TOP019_UAV_media_analyzed	3	6
TOP021_TEST_INCIDENT_REPORT	0	0
TOP022_PUBLIC_ALERT	0	0
TOP031_UAVP_MESSAGE	7	13
TOP031_UAVP_MESSAGE_TEST	0	0
TOP032_TEST_UAVP_MESSAGE	0	0
TOP102_TEAM_REPORT	36	48
TOP104_METRIC_REPORT	679	363
TOP111_SYSTEM_INITIALIZATION	0	0
TOP112_SUMMARY_TRIGGER	0	0
TOP803_WEATHER_REPORT	0	0
TOP001_SOCIAL_MEDIA_TEXT	2	0
TOP003_SOCIAL_MEDIA_REPORT	1	0
TOP021_INCIDENT_REPORT	3	3
TOP028_TEXT_ANALYSED	4	1
TOP030_REPORT_REQUESTED	15	11

Bytes Out (15 Minute Rate)

	avg	current
TOP023_TASK_ASSIGNMENT	34	21
TOP031_UAVP_TEST_MESSAGE	0	0
TOP033_SUMMARY_REQUESTED	0	0
TOP040_TEXT_REPORT_GENERATED	9	7
TOP101_INCIDENT_REPORT	59	58
TOP103_TASK_REPORT	57	129
TOP105_CRCL_INITIALIZATION	0	0
TOP106_METRIC_REPORT	0	0
TOP140_SUMMARY_REPORT	0	0
TOP801_INCIDENT_VALIDATION	5	3
TOP103_TASK_REPORT	57	129
TOP105_CRCL_INITIALIZATION	0	0
TOP106_METRIC_REPORT	0	0
TOP140_SUMMARY_REPORT	0	0
TOP801_INCIDENT_VALIDATION	5	3
TOP802_WEATHER_REQUEST	0	0
TOP006_INCIDENT_REPORT	0	0
TOP018_image_analyzed	4	3
TOP019_UAV_MEDIA_ANALYZED	0	0
TOP019_UAV_media_analyzed	9	18
TOP021_TEST_INCIDENT_REPORT	0	0
TOP022_PUBLIC_ALERT	1	0
TOP031_UAVP_MESSAGE	14	26
TOP031_UAVP_MESSAGE_TEST	0	0
TOP032_TEST_UAVP_MESSAGE	0	0
TOP102_TEAM_REPORT	138	186
TOP104_METRIC_REPORT	3.40 K	1.82 K
TOP111_SYSTEM_INITIALIZATION	0	0
TOP112_SUMMARY_TRIGGER	0	0
TOP803_WEATHER_REPORT	0	0
TOP001_SOCIAL_MEDIA_TEXT	6	0
TOP003_SOCIAL_MEDIA_REPORT	3	0
TOP021_INCIDENT_REPORT	16	13
TOP028_TEXT_ANALYSED	14	2
TOP030_REPORT_REQUESTED	62	42

Figure 17: Message Bus activity

Attribute Name	Attribute Description	CAP Equivalent	Attribute Type
topicName	name of topic of	n/a	string
topicMajorVersion	topic major version	n/a	uint8
topicMinorVersion	topic minor version	n/a	uint8

sender	originating system	sender	string
msgIdentifier	unique message identifier	identifier	string
sentUTC	UTC time message was sent	sent	datetime
status	The code denoting the appropriate handling of the alert message	status	string
actionType	type of action to apply to alert	msgType	string
specificSender	Originating specific	source	string
scope	intended distribution of the alert	scope	string
district	specific district in which the message is relevant	restriction	string
recipients	specific recipients' name, id, ip address, etc. within the subscriber to whom the message is intended	addresses	string
code	identifier of the reality to which the message applies	code	unit32
note	textual notification for various purposes	note	string
references	optional identifier of previous messages to which the current message refers, or URI in which the content of the current message is stored (multiple values allowed)	references	string

Figure 18: beAWARE messages common header

3.2.2 **FROST-Server**

The FROST-Server, formally called SensorThingsAPI-Server is responsible to store the time series data inside the beAWARE platform. The implementation of the storage component itself was already finished for the second prototype. The number of used data sources was increased for the final system. To support the Pilot, forecasted weather data is continuously imported from the FMI weather predictions for the pilot region in Valencia.

The previous version already included a map to visualize and analyse the time series data, stored in the FROST-Server. For the final system, the analysis functionality was extended. For example, as shown in Figure 19, it is possible to visualize the data of different places in one graph. This offers the possibility to easily compare measurements of different places.**Error! R eference source not found.**

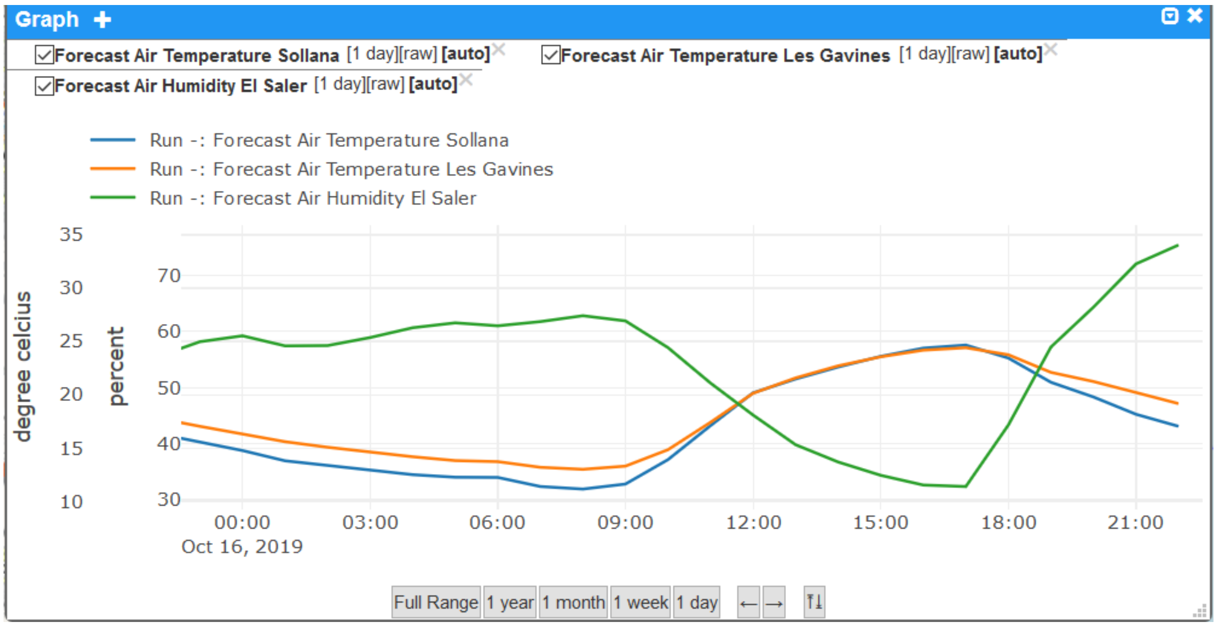
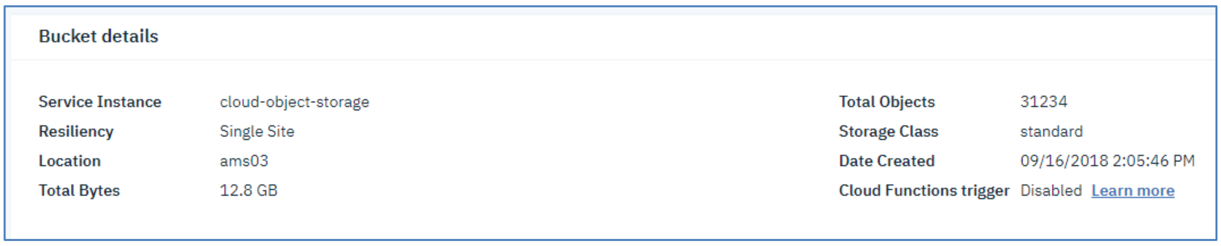


Figure 19 Visulizing the data of different places in one graph

3.2.3 Object Storage

Cloud based object storage serves as the central data repository of the platform, storing heterogeneous structured and unstructured content, in the form of images, videos, text file, etc.

This component serves for storing and making available large amounts of data to interested components. The flow typically consists of a component storing a data item in the object store, followed by sending a message incorporating the link to the specific media in the object store.



Bucket details

Service Instance	cloud-object-storage	Total Objects	31234
Resiliency	Single Site	Storage Class	standard
Location	ams03	Date Created	09/16/2018 2:05:46 PM
Total Bytes	12.8 GB	Cloud Functions trigger	Disabled Learn more

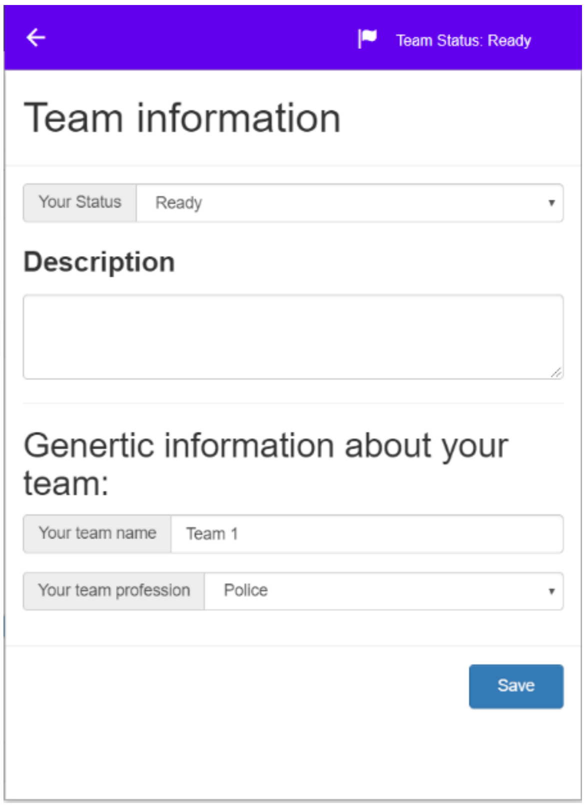
Figure 20: Object Store configuration

3.3 External Facing Layer

This layer handles the interaction of the platform with external entities, both as input providers and output recipients.

3.3.1 Mobile Application

The mobile application is the main interaction point for citizens and first responders with the beAWARE platform. Its functionality and the improvements since the second prototype are explained in detail in deliverable "D7.7 User applications". This includes, for example the updated user interface and the improvements in the task management functionality. Not reported until now is the possibility for a team to specify its name and profession (see Figure 21). The teams' profession is not only transmitted to the authorities. Public alerts can now be limited to a specific first responder role, so it's only visible for them.



Team information

Your Status: Ready

Description

Genertic information about your team:

Your team name: Team 1

Your team profession: Police

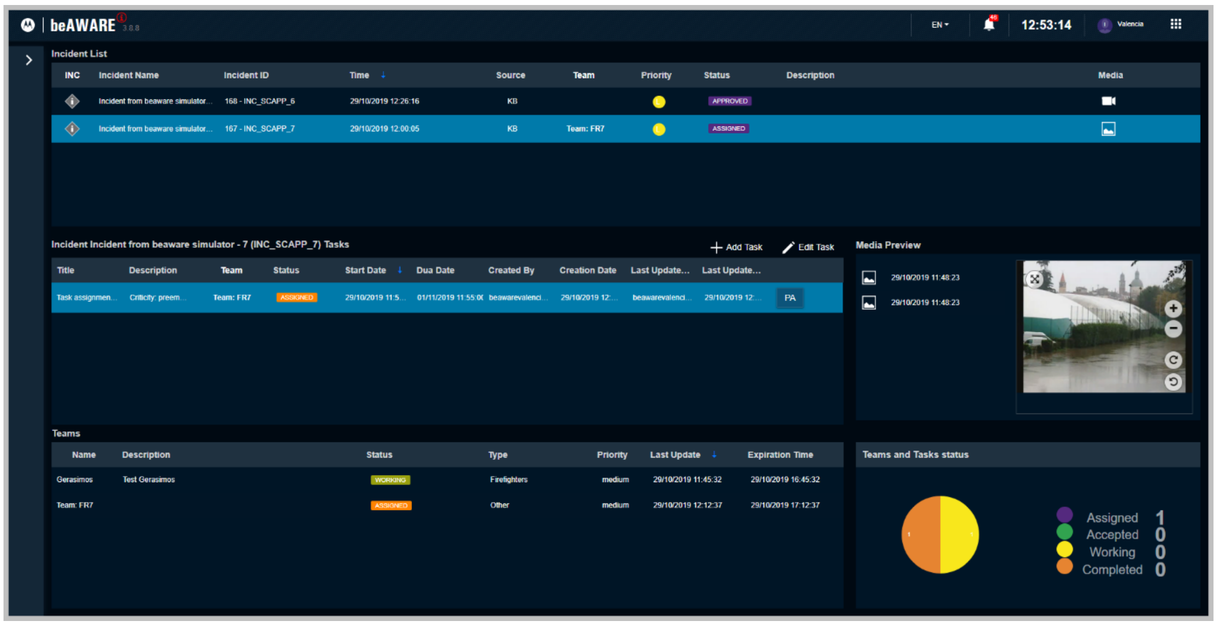
Figure 21 Selecting the team name and profession

3.3.2 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.

In the second version of PSAP, we improved the map visualization and incident clustering mechanism including Raw and analyzed media like images, videos, and audio, stacking multiple media files from different incident updates, thus creating a unified situational picture together with enhanced Incident manager module to show all the necessary information to the user.

Incident information like status, severity, priory, description, type and more, can be edited via PSAP, by simply selecting the relevant incident, edit information below and click 'Save Changes'.



Incident List

INC	Incident Name	Incident ID	Time	Source	Team	Priority	Status	Description	Media
	Incident from beaware simulator...	168 - INC_SCAPP_6	29/10/2019 12:26:16	KB		L	APPROVED		
	Incident from beaware simulator...	167 - INC_SCAPP_7	29/10/2019 12:00:05	KB	Team: FR7	L	ASSIGNED		

Incident Incident from beaware simulator - 7 (INC_SCAPP_7) Tasks

Title	Description	Team	Status	Start Date	Dua Date	Created By	Creation Date	Last Update...	Last Update...	
Task assignmen...	Criticity: preem...	Team: FR7	ASSIGNED	29/10/2019 11:5...	01/11/2019 11:55:0(	beawarevalenci...	29/10/2019 12:...	beawarevalenci...	29/10/2019 12:...	PA

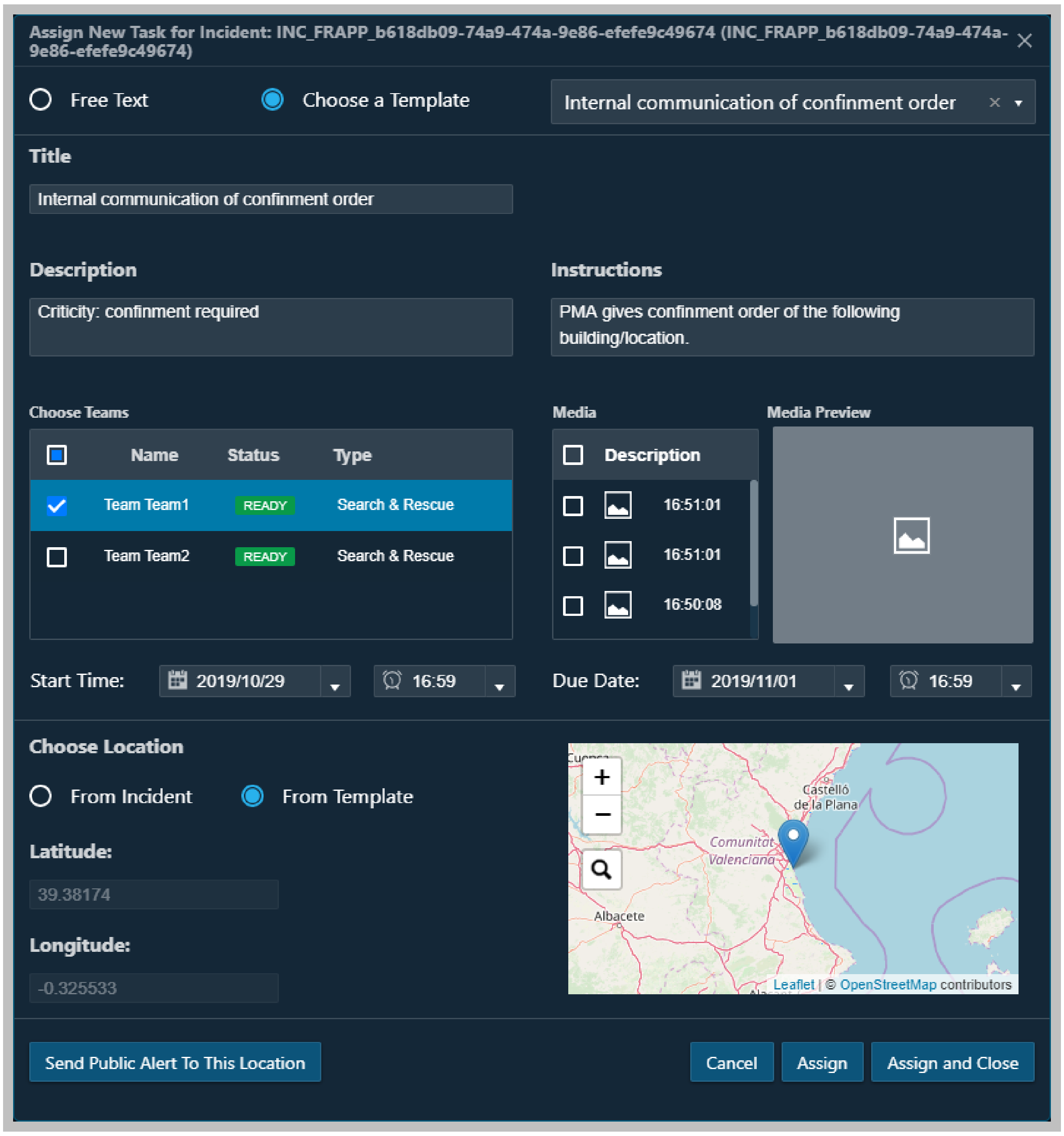
Teams

Name	Description	Status	Type	Priority	Last Update	Expiration Time
Gerasimos	Test Gerasimos	WORKING	Firefighters	medium	29/10/2019 11:45:32	29/10/2019 16:45:32
Team: FR7		ASSIGNED	Other	medium	29/10/2019 12:12:37	29/10/2019 17:12:37

Figure 22: Incident Manager

Figure 22 show the possibility to view all incidents, available teams and assigned tasks. The operator has the ability to select an Incident from the list and assign tasks and teams.

In addition, the first version of the Operations Manager module has been developed, for incident management, giving the ability to send tasks to available teams containing relevant information like descriptions, instructions, media, location, etc and tracking the tasks and teams statuses.



Assign New Task for Incident: INC_FRAPP_b618db09-74a9-474a-9e86-efefe9c49674 (INC_FRAPP_b618db09-74a9-474a-9e86-efefe9c49674)

Free Text [✓ Choose a Template] Internal communication of confinement order

Title

Internal communication of confinement order

Description

Criticity: confinment required

Instructions

PMA gives confinment order of the following building/location.

Choose Teams

	Name	Status	Type
✓	Team Team1	READY	Search & Rescue
	Team Team2	READY	Search & Rescue

Media

	Description
	16:51:01
	16:51:01
	16:50:08

Media Preview

Start Time: 2019/10/29 16:59 Due Date: 2019/11/01 16:59

Choose Location

From Incident [✓ From Template]

Latitude:

39.38174

Longitude:

-0.325533

Send Public Alert To This Location

Cancel Assign Assign and Close

Figure 23: Task assignment feature.

Figure 23 show the form for assigning tasks to teams and sending public alert to the task location area.

4 Components and Integration status

Table 2: Social Media Analysis Tool status

Service name	Social Media Analysis	Social Media Clustering
CI cluster	beAWARE-project/social-media-analysis-live	beAWARE-project/social-media-clustering-live
Functional Description	Version 1.1, working version	Version 1.0, working version
Deployment status	Deployed	Deployed
Integration status	Integrated	Integrated
Integration issues / dependencies	All dependencies are included in pom.xml. Environment variables: SECRET_MH_API_KEY, SECRET_MH_BROKERS, SECRET_MONGO_URI, TWITTER_API_CONSUMER_KEY, TWITTER_API_CONSUMER_SECRET, TWITTER_API_SECRET, TWITTER_API_TOKEN	All dependencies are included in pom.xml. Environment variables: SECRET_MH_API_KEY, SECRET_MH_BROKERS, SECRET_MONGO_URI

Table 3: Media Hub Tool status

Service name	Central Hub to Assign Media Analysis
CI cluster	beAWARE-project/ Media Hub
Functional Description	Version 1.1, working version
Deployment status	Deployed

Integration status	Integrated
Integration issues / dependencies	All dependencies are included in pom.xml. Environment variables: SECRET_MH_API_KEY, SECRET_MH_BROKERS For a complete workflow, it requires CI clusters ASR, image-analysis, video-analysis, and drone-analysis.

Table 4: Crisis Classification status

Service name	Crisis Classification
CI cluster	beaware-project/ crisis-classification
Functional Description	Version 2.0, working version
Development status	Implementation of first prototype. This includes the updated versions for the *Early Warning* and *Real-Time Monitoring and Risk Assessment* components for the flood pilot. Implement interoperability functionalities with EFAS impact/risk map.
Deployment status	Deployed
Integration status	Integrated
Integration issues / dependencies	No issues regarding the integration with other components. Dependencies show up in the Dockerfile

Table 5: Image Analysis Tool status

Service name	Image Analysis
CI cluster	beaware-project/image-analysis
Functional Description	Version 2.0, advanced version of image analysis techniques

Deployment status	Deployed
Integration status	Integrated
Integration issues /dependencies	Communicates with Media Hub service Uses port 7788 Dependencies show up in the Dockerfile

Table 6: Video Analysis Tool status

Service name	Video Analysis
CI cluster	beaware-project/video-analysis
Functional Description	Version 2.0, advanced version of video analysis techniques
Deployment status	Deployed
Integration status	Integrated
Integration issues /dependencies	Communicates with Media Hub service Uses port 7777 Dependencies show up in the Dockerfile

Table 7: Automatic Speech Recognition Tool status

Service name	Automatic Speech Recognition
CI cluster	beaware-project/ASR
Functional Description	Version 3.0, final version
Development status	Adapted versions of Greek, Spanish and Italian ASR models. ASR has been extended to include phone calls. Enhanced noise removal algorithms.
Deployment status	Deployed
Integration status	Integrated
Integration issues / dependencies	No issues regarding integration All dependencies are included in pom.xml project file. It communicates with Media Hub:

Table 8: SCAPP/FRAPP status

Service name	SCAPP/FRAPP (Mobile Application)
CI cluster	Web based version available on the cluster. The APK installation package for Android mobile devices is separately shared with a download link
Functional Description	End user application for mobile devices.
Development status	Implementation of final system. This includes improvements of the user interface. In contrast to the second prototype, the team functionality was improved, to that a team can specify its name and profession. The public alert mechanism was enhanced to restrict alerts to specific first responder roles.
Deployment status	Web version deployed to cluster.
Integration status	Interface with other components specified. Successfully integrated with the other components.
Integration issues /dependencies	No issues regarding the integration with other components. Source code should not be available to public. Therefore, a separate, not publicly available continuous integration process is setup.

Table 9: Knowledge Base status

Service name	KB
CI cluster	Deployed on the cluster.
Functional Description	Server and management APIs for semantic data.
Development status	Knowledge Base fully developed. Latest version of ontology deployed to Knowledge Base. Included risk maps and interfaces for analysing incidents and the semantic content.
Deployment status	Deployed to cluster.
Integration status	Provided interfaces for other components to access/modify semantic data.

Integration issues /dependencies	No issues regarding the integration with other components. To store the data internally the Knowledge Base depends on a MySQL-Server. This is provided as a service by IBM Bluemix. Source code should not be available to public. Therefore, a separate, not publicly available continuous integration process is setup.

Table 10: Drones Analysis Tool status

Service name	DA
CI cluster	Locally deployed on beAWARE server
Functional Description	Version 3.0, final version
Development status	Support for both image and video sequences. Performs object detection and tracking in order to detect people and vehicles (Object Detection task). Performs image classification in order to detect flood, smoke and fire in images (Crisis Detection task). Detects remaining people in areas under evacuation and creates relevant reports (Evacuation task).
Deployment status	Deployed
Integration status	Integrated. DA communicates with Drones Platform through Media Hub, which also informs KBs for possible detected incidents.
Integration issues / dependencies	No issues regarding integration. It requires a GPU in order to accelerate image analysis. For near real-time analysis the frame rate should be relatively low (1-5 fps), whereas, for offline analysis there is no such constraint.

Table 11: Visual River Sensing Tool status

Service name	VRS
CI cluster	Locally deployed on beAWARE server

Functional Description	Version 3.0, final version
Development status	Streams and analyzes videos from static surveillance cameras and estimates water level in order to create alerts. In case of an alert, it forwards the video to ObD module for further analysis.
Deployment status	Deployed
Integration status	Integrated.
Integration issues / dependencies	No issues regarding integration. VRS analysis is based on GPU.

Table 12: FROST-Server status

Service name	SENSAN
CI cluster	Deployed to the cluster.
Functional Description	Server and management APIs for sensor data.
Development status	Server fully implemented. Data integration is done and importing new data is continuously ongoing.
Deployment status	Server, importer and processing scripts are deployed to cluster.
Integration status	Integrated. Data visualization integrated in UI.
Integration issues /dependencies	No issues regarding the integration. All available data sources are integrated.

Table 9: Multilingual Text Analyzer Tool status

Service name	Text Analysis Tool
CI cluster	Text-analysis

Functional Description	Version 1.0, basic version
Deployment status	Deployed
Integration status	Integrated
Integration issues /dependencies	Newly added contents in the output such as the results of geolocation require extensions to TOP028 and TOP030 Kafka bus messages and to the KBS service in order to be correctly handed by components consuming these contents.

Table 10: Multilingual Report Generation Tool status

Service name	Report Generation Tool
CI cluster	Report-generation
Functional Description	Version 1.0, basic version
Deployment status	Deployed
Integration status	Integrated. Reports are integrated in the UI.
Integration issues /dependencies	Production of wrap-up summaries require additional information from the KBS.

Table 13: Drones Platform status

Service name	Drones Platform Tool
CI cluster	Code is not public, held in an internal enterprise repository.
Functional Description	Version 1.0
Deployment status	Deployed

Integration status	Integrated. Main integration point is with the drones analysis component using the platform tools (message bus, and object storage).
Integration issues /dependencies	No current open issues.

Table 11: PSAP status

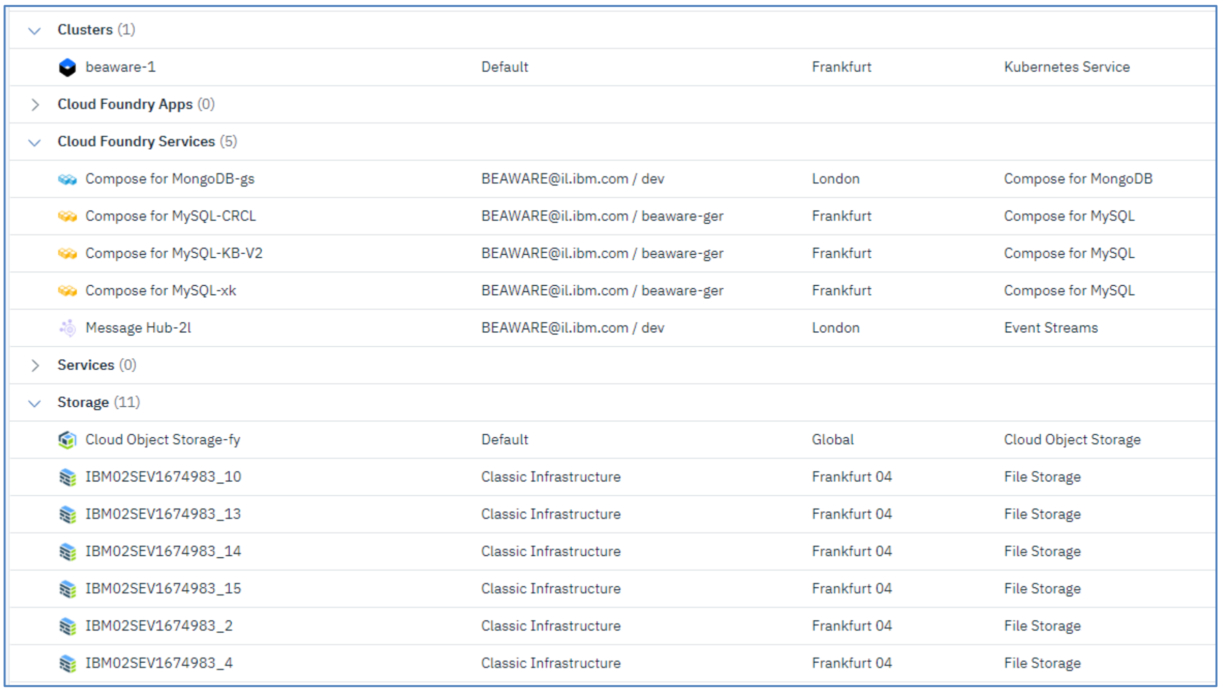
Service name	Drones Platform Tool
CI cluster	Code is not public, held in an internal enterprise repository.
Functional Description	Version 1.0
Deployment status	Deployed
Integration status	Integrated. Main integration points using the platform tools (message bus, and object storage).
Integration issues /dependencies	No current open issues.

5 Platform Deployment

WE have established a complete CI / CD toolchain for the project, from source code to running components. Source code control is handled by GitHub; connection to deployment is handled by Jenkins; deployment and hosting issues are handled by Kubernetes.

Currently there is a specific IBM cloud account which is dedicated for the beAWARE project; all cloud services and deployment are grouped in this account.

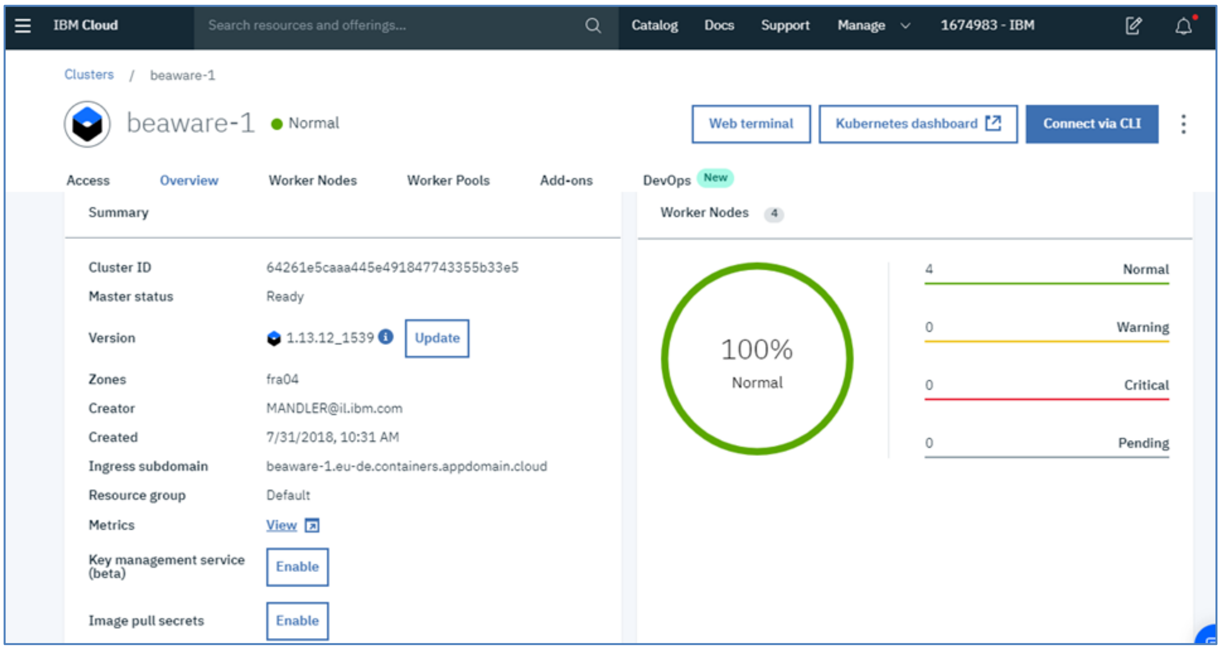
A view of the dashboard of beAWARE's cloud account can be seen in Figure 24. In the dashboard we can see the cloud services which are used by the beAWARE platform: Message Bus and Object Store which are used by most platform components. In addition, we can see an instance of MySQL which is mainly used by the FROST server, and an instance of MongoDB which is mainly used by the social media analysis component. We can further see the Kubernetes cluster on which the platform is deployed.



Clusters (1)			
beaware-1	Default	Frankfurt	Kubernetes Service
Cloud Foundry Apps (0)			
Cloud Foundry Services (5)			
Compose for MongoDB-gs	BEAWARE@il.ibm.com / dev	London	Compose for MongoDB
Compose for MySQL-CRCL	BEAWARE@il.ibm.com / beaware-ger	Frankfurt	Compose for MySQL
Compose for MySQL-KB-V2	BEAWARE@il.ibm.com / beaware-ger	Frankfurt	Compose for MySQL
Compose for MySQL-xk	BEAWARE@il.ibm.com / beaware-ger	Frankfurt	Compose for MySQL
Message Hub-2l	BEAWARE@il.ibm.com / dev	London	Event Streams
Services (0)			
Storage (11)			
Cloud Object Storage-fy	Default	Global	Cloud Object Storage
IBM02SEV1674983_10	Classic Infrastructure	Frankfurt 04	File Storage
IBM02SEV1674983_13	Classic Infrastructure	Frankfurt 04	File Storage
IBM02SEV1674983_14	Classic Infrastructure	Frankfurt 04	File Storage
IBM02SEV1674983_15	Classic Infrastructure	Frankfurt 04	File Storage
IBM02SEV1674983_2	Classic Infrastructure	Frankfurt 04	File Storage
IBM02SEV1674983_4	Classic Infrastructure	Frankfurt 04	File Storage

Figure 24: IBM cloud account dashboard

At the centre of the figure we can see the Kubernetes cluster, which hosts the bulk of the system components. A cluster overview can be seen in Figure 25.The entry point for the platform's Kubernetes cluster is: https://beaware-1.eu-de.containers.appdomain.cloud/.



Summary

Cluster ID	64261e5caaa445e491847743355b33e5
Master status	Ready
Version	1.13.12_1539 Update
Zones	fra04
Creator	MANDLER@il.ibm.com
Created	7/31/2018, 10:31 AM
Ingress subdomain	beaware-1.eu-de.containers.appdomain.cloud
Resource group	Default
Metrics	View
Key management service (beta)	Enable
Image pull secrets	Enable

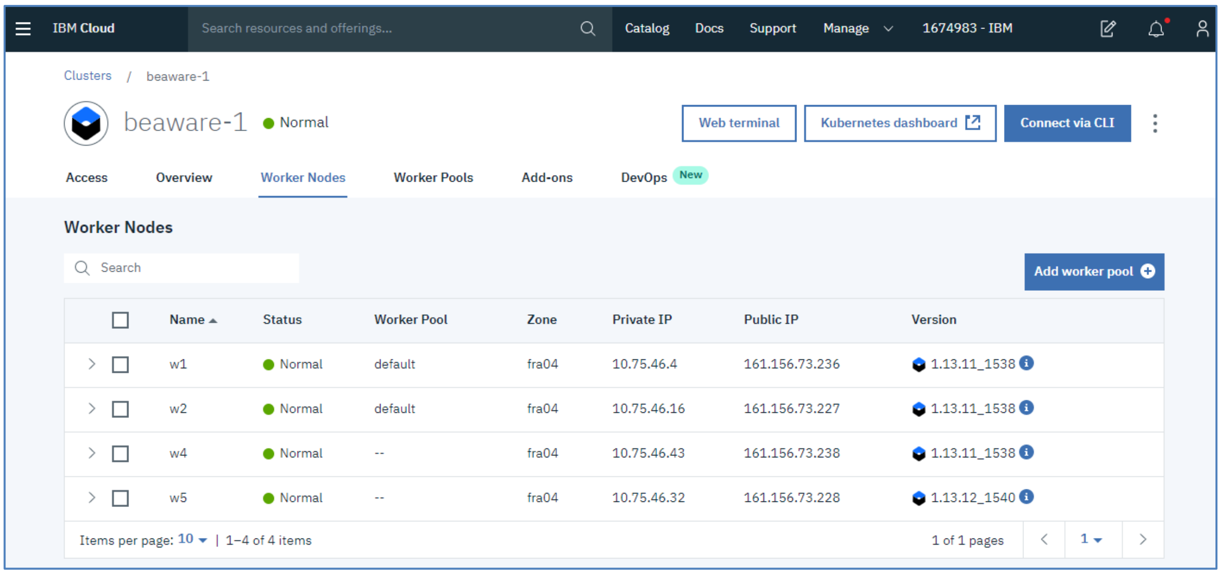
Worker Nodes 4

100% Normal

4	Normal
0	Warning
0	Critical
0	Pending

Figure 25: beAWARE Kubernetes cluster overview

Currently the cluster is composed of four worker nodes as seen in Figure 26, and it may grow based on evolving system needs. All the worker nodes consist of 4 Cores and 16GB RAM.



Name	Status	Worker Pool	Zone	Private IP	Public IP	Version
w1	Normal	default	fra04	10.75.46.4	161.156.73.236	1.13.11_1538
w2	Normal	default	fra04	10.75.46.16	161.156.73.227	1.13.11_1538
w4	Normal	--	fra04	10.75.46.43	161.156.73.238	1.13.11_1538
w5	Normal	--	fra04	10.75.46.32	161.156.73.228	1.13.12_1540

Items per page: 10 | 1–4 of 4 items 1 of 1 pages 1

Figure 26: Kubernetes cluster worker nodes

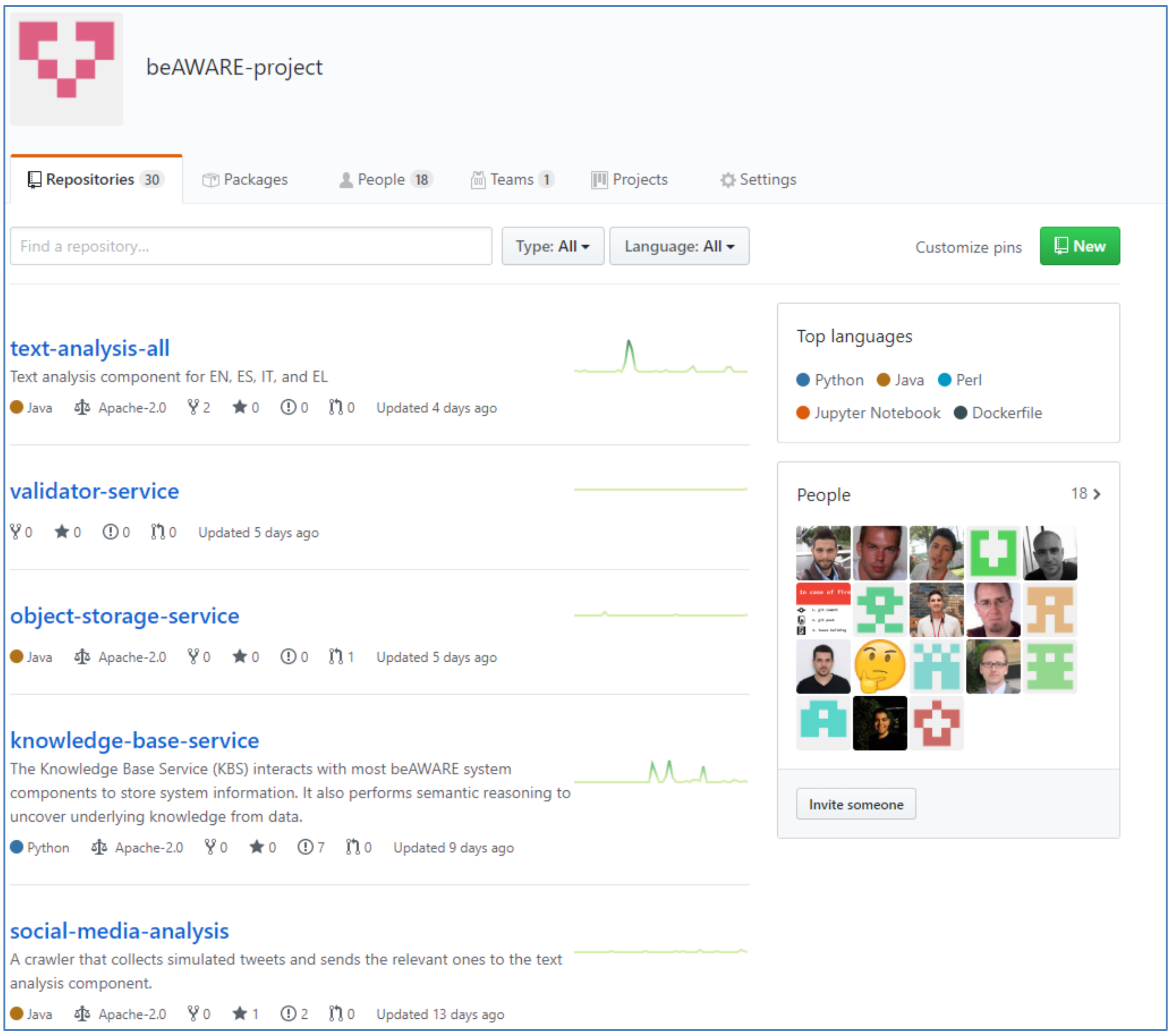
The K8s cluster is divided into 3 namespaces; the one used for beAWARE deployment is the “prod” namespace.

Moreover, there are components that are deployed external to the project K8s cluster, such as PSAP, while their backend may still reside within the project cluster.

The Continuous Integration (CI) environment is comprised of the following components

1. GitHub repository: all components should have a repository under the beAWARE project (https://github.com/beAWARE-project).
2. Docker – a docker image is created for each component.
3. Jenkins: build a new version of a microservice based on code commit, and deploys the new version of the service to the Kubernetes cluster.

4. Kubernetes -IBM container services - managed cluster on which all components are deployed
   - Requires specific Kubernetes configuration for each component


beAWARE-project

text-analysis-all
Text analysis component for EN, ES, IT, and EL
● Java Apache-2.0 2 ★ 0 0 0 Updated 4 days ago

validator-service
0 ★ 0 0 0 Updated 5 days ago

object-storage-service
● Java Apache-2.0 0 ★ 0 0 1 Updated 5 days ago

knowledge-base-service
The Knowledge Base Service (KBS) interacts with most beAWARE system components to store system information. It also performs semantic reasoning to uncover underlying knowledge from data.
● Python Apache-2.0 0 ★ 0 7 0 Updated 9 days ago

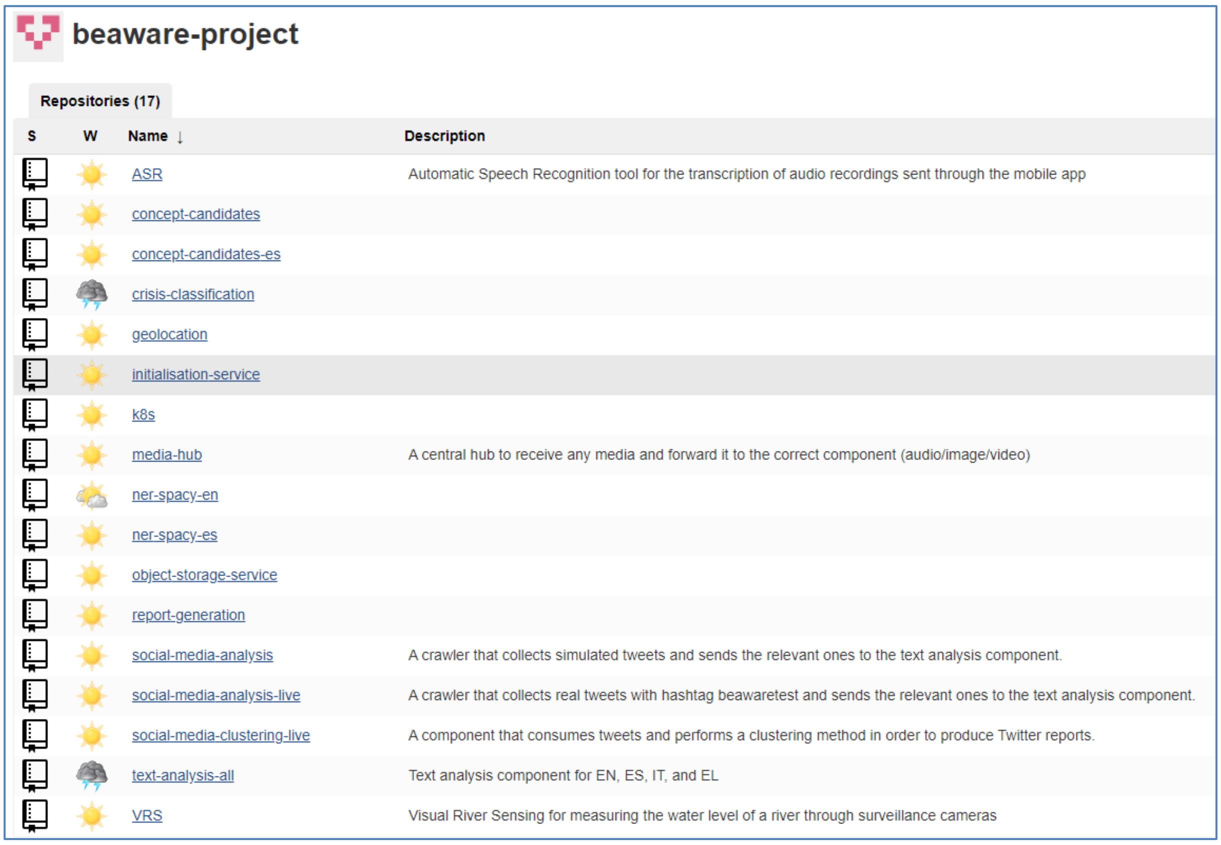
social-media-analysis
A crawler that collects simulated tweets and sends the relevant ones to the text analysis component.
● Java Apache-2.0 0 ★ 1 2 0 Updated 13 days ago

Top languages
● Python ● Java ● Perl ● Jupyter Notebook ● Dockerfile

People 18

Figure 27: Project GitHub repository

The automated workflow kicks in upon a new commit to a project in the GitHub repository. The standard procedure, dictated by a JenkinsFile, is to build the component using the dockerFile, and if no errors reported, to deploy to the Kubernetes cluster, using the specified K8s configuration. This process happens for every repository for which there is an associated JenkinsFile.

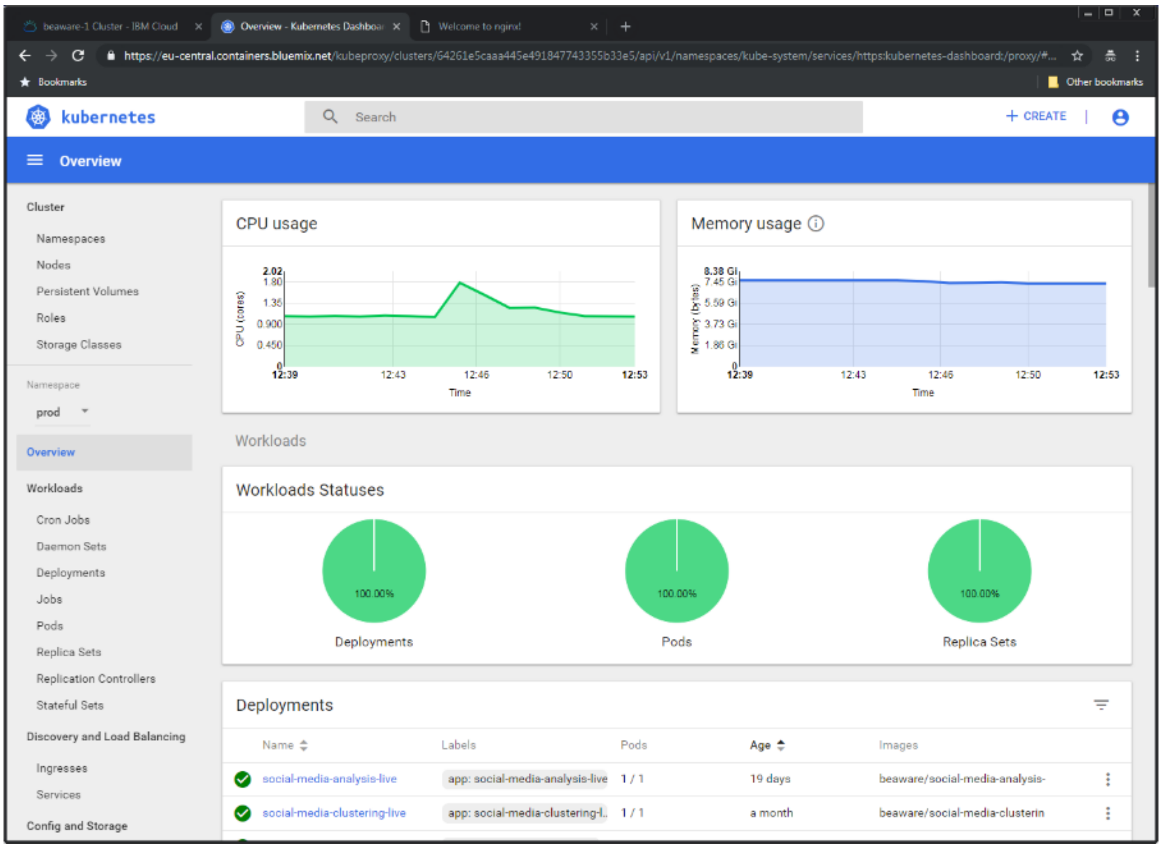


beaware-project

Repositories (17)

S	W	Name ↓	Description
	☀	ASR	Automatic Speech Recognition tool for the transcription of audio recordings sent through the mobile app
	☀	concept-candidates	
	☀	concept-candidates-es	
	⛈	crisis-classification	
	☀	geolocation	
	☀	initialisation-service	
	☀	k8s	
	☀	media-hub	A central hub to receive any media and forward it to the correct component (audio/image/video)
	⛅	ner-spacy-en	
	☀	ner-spacy-es	
	☀	object-storage-service	
	☀	report-generation	
	☀	social-media-analysis	A crawler that collects simulated tweets and sends the relevant ones to the text analysis component.
	☀	social-media-analysis-live	A crawler that collects real tweets with hashtag beawaretest and sends the relevant ones to the text analysis component.
	☀	social-media-clustering-live	A component that consumes tweets and performs a clustering method in order to produce Twitter reports.
	⛈	text-analysis-all	Text analysis component for EN, ES, IT, and EL
	☀	VRS	Visual River Sensing for measuring the water level of a river through surveillance cameras

Figure 28: Jenkins dashboard



Name	Labels	Pods	Age	Images
social-media-analysis-live	app: social-media-analysis-live	1 / 1	19 days	beaware/social-media-analysis-
social-media-clustering-live	app: social-media-clustering-l...	1 / 1	a month	beaware/social-media-clusterin

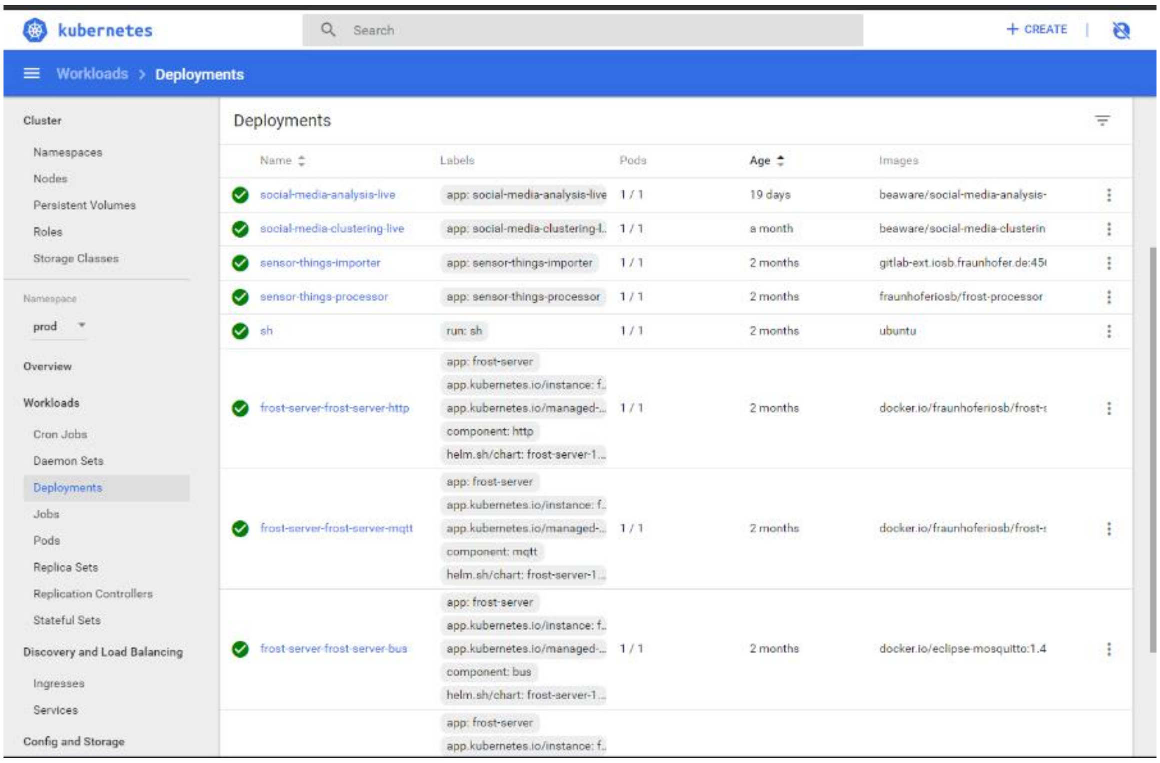
Figure 29: beAWARE Kubernetes cluster

A sub-set of the individual components deployed within the cluster can be seen in Figure 30. beAWARE code on the GitHub repository is organised on a per-component basis. The root of the source tree is located at: https://github.com/beAWARE-project.

The code of the individual components can be found in the following links:

- **Text Analysis module**: Text analysis tools to extract information from tweets or other sources, Maven package (https://github.com/beAWARE-project/text-analysis-all)
- Text analysis on ASR outputs: Maven package (https://github.com/beAWARE-project/text-analysis-asr)
- Automatic Speech Recognition tool: for the transcription of audio recordings sent through the mobile app, Maven package (https://github.com/beAWARE-project/ASR)
- Social Media Analysis tool: A crawler that collects tweets and pushes the relevant ones to the bus, Maven package (https://github.com/beAWARE-project/social-media-analysis)
- Social Media Analysis tool: A crawler that collects tweets and pushes the relevant ones to the bus, Maven package (https://github.com/beAWARE-project/social-media-analysis-live)

- **Social Media Clustering tool**: A component that consumes analyzed tweets and performs a clustering method in order to produce Twitter reports, Maven package. (https://github.com/beAWARE-project/social-media-clustering-live)
- Image Analysis tool: Performs image analysis for the beAWARE project, Python (https://github.com/beAWARE-project/image-analysis)
- Video Analysis tool: Performs video analysis for the beAWARE project, Python (https://github.com/beAWARE-project/video-analysis)
- **Media Hub**: A central hub to receive any media and forward it to the correct component (audio/image/video), Maven package (https://github.com/beAWARE-project/media-hub)
- Ontology: the beAWARE Knowledge Base Ontology (https://github.com/beAWARE-project/ontology)
- Report Generation: Generates new incident reports, Maven package (https://github.com/beAWARE-project/report-generation)
- K8s: BeAWARE Kubernetes configuration (https://github.com/beAWARE-project/k8s)
- Object Storage Service: Applications for storing and retrieving from the data repository, Maven package **(https://github.com/beAWARE-project/object-storage-service)**
- Drones Analysis: Module to analyse drone data, Maven package (https://github.com/beAWARE-project/drone-analysis)
- Crisis Classification: Module to determine current crisis level (https://github.com/beAWARE-project/crisis-classification)
- **Knowledge Base Service:** Module to store system information; also performing semantic reasoning to uncover underlying knowledge from data. (https://github.com/beAWARE-project/knowledge-base-service)



Deployments

Name	Labels	Pods	Age	Images
✓ social-media-analysis-live	app: social-media-analysis-live	1 / 1	19 days	beaware/social-media-analysis-
✓ social-media-clustering-live	app: social-media-clustering-l...	1 / 1	a month	beaware/social-media-clusterin
✓ sensor-things-importer	app: sensor-things-importer	1 / 1	2 months	gitlab-ext.iosb.fraunhofer.de:45
✓ sensor-things-processor	app: sensor-things-processor	1 / 1	2 months	fraunhoferiosb/frost-processor
✓ sh	run: sh	1 / 1	2 months	ubuntu
✓ frost-server-frost-server-http	app: frost-server app.kubernetes.io/instance: f... app.kubernetes.io/managed-... component: http helm.sh/chart: frost-server-1...	1 / 1	2 months	docker.io/fraunhoferiosb/frost-s
✓ frost-server-frost-server-mqtt	app: frost-server app.kubernetes.io/instance: f... app.kubernetes.io/managed-... component: mqtt helm.sh/chart: frost-server-1...	1 / 1	2 months	docker.io/fraunhoferiosb/frost-s
✓ frost-server-frost-server-bus	app: frost-server app.kubernetes.io/instance: f... app.kubernetes.io/managed-... component: bus helm.sh/chart: frost-server-1...	1 / 1	2 months	docker.io/eclipse-mosquitto:1.4
	app: frost-server app.kubernetes.io/instance: f...			

Figure 30: Cluster deployments

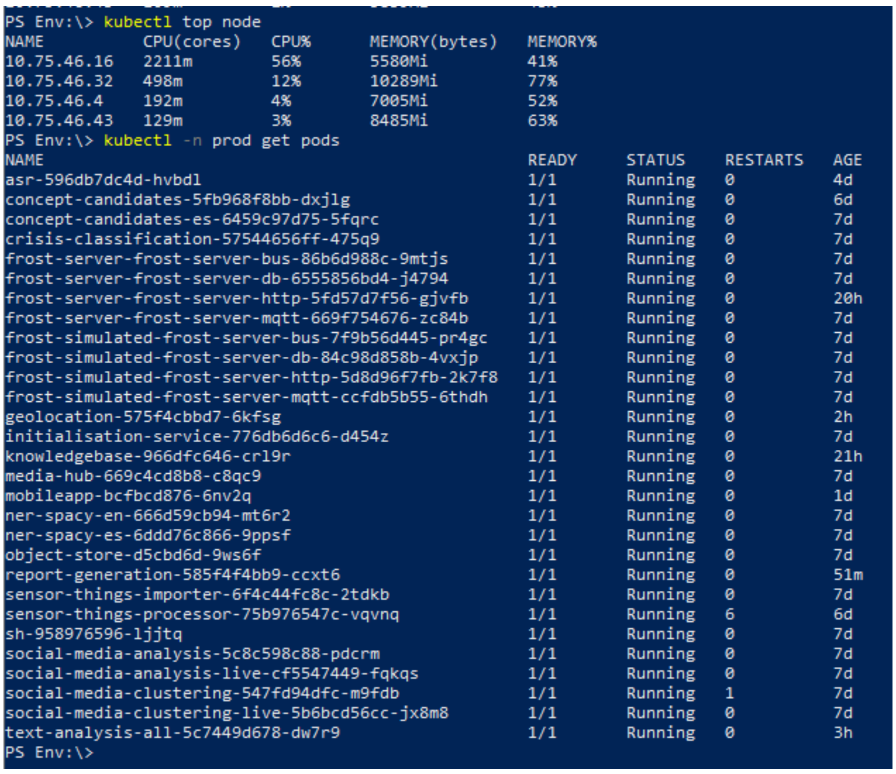

```
PS Env:\> kubectl top node
NAME           CPU(cores)   CPU%      MEMORY(bytes)   MEMORY%
10.75.46.16    2211m        56%       5580Mi          41%
10.75.46.32    498m         12%       10289Mi         77%
10.75.46.4     192m         4%        7005Mi          52%
10.75.46.43    129m         3%        8485Mi          63%
PS Env:\> kubectl -n prod get pods
NAME                                                READY     STATUS    RESTARTS   AGE
asr-596db7dc4d-hvbdl                                1/1       Running   0          4d
concept-candidates-5fb968f8bb-dxjlg                 1/1       Running   0          6d
concept-candidates-es-6459c97d75-5fqrc              1/1       Running   0          7d
crisis-classification-57544656ff-475q9              1/1       Running   0          7d
frost-server-frost-server-bus-86b6d988c-9mtjs       1/1       Running   0          7d
frost-server-frost-server-db-6555856bd4-j4794       1/1       Running   0          7d
frost-server-frost-server-http-5fd57d7f56-gjvfb     1/1       Running   0          20h
frost-server-frost-server-mqtt-669f754676-zc84b     1/1       Running   0          7d
frost-simulated-frost-server-bus-7f9b56d445-pr4gc   1/1       Running   0          7d
frost-simulated-frost-server-db-84c98d858b-4vxjp    1/1       Running   0          7d
frost-simulated-frost-server-http-5d8d96f7fb-2k7f8  1/1       Running   0          7d
frost-simulated-frost-server-mqtt-ccfdb5b55-6thdh   1/1       Running   0          7d
geolocation-575f4cbbd7-6kfsg                        1/1       Running   0          2h
initialisation-service-776db6d6c6-d454z             1/1       Running   0          7d
knowledgebase-966dfc646-cr19r                       1/1       Running   0          21h
media-hub-669c4cd8b8-c8qc9                          1/1       Running   0          7d
mobileapp-bcfbcd876-6nv2q                           1/1       Running   0          1d
ner-spacy-en-666d59cb94-mt6r2                       1/1       Running   0          7d
ner-spacy-es-6ddd76c866-9ppsf                       1/1       Running   0          7d
object-store-d5cbd6d-9ws6f                          1/1       Running   0          7d
report-generation-585f4f4bb9-ccxt6                  1/1       Running   0          51m
sensor-things-importer-6f4c44fc8c-2tdkb             1/1       Running   0          7d
sensor-things-processor-75b976547c-vqvnq            1/1       Running   6          6d
sh-958976596-ljjtq                                  1/1       Running   0          7d
social-media-analysis-5c8c598c88-pdcrm              1/1       Running   0          7d
social-media-analysis-live-cf5547449-fqkqs          1/1       Running   0          7d
social-media-clustering-547fd94dfc-m9fdb            1/1       Running   1          7d
social-media-clustering-live-5b6bcd56cc-jx8m8       1/1       Running   0          7d
text-analysis-all-5c7449d678-dw7r9                  1/1       Running   0          3h
PS Env:\>
```

Figure 31: resources consumption and pods

6 Demonstrator URLS and information

In order to demonstrate the final version of the beAWARE platform, a video has been created, capturing the whole functionality of the platform and it is available in the following link:

https://beaware-project.eu/resources/videos/beaware-3rdprototypevideo/

Briefly, the demonstration has 4 phases:

- **Phase 1**-Pre-emergency phase. The objective of this phase is the early provision of information on emerging hazards and the quick distribution of the first alerts to the public. In this phase, the pre-emergence level 3 is activated.
- **Phase 2**- Emergency phase.
    - The first part of the emergency phase serves as the starting point for setting the scene. At this point, the first reports are received that smoke has been spotted in the district.
    - The second part shows that the situation is worsening but is not critical yet. The weather forecasting shows that the fire front is moving to the sea. No people or properties are under immediate threat yet. Nevertheless, teams are deployed to the area to address any potential danger and the traffic is regulated across the danger zone.
- **Phase3**-The third phase includes a change in the direction of the wind. School's evacuation is ordered to ensure public safety.
- **Phase4**- Fade out.

The rest of this section contains a tutorial with some descriptive screenshots.

6.1 Pre-Emergency Phase

During the pre-emergency phase, the Crisis Classification module acquires forecasting data to classify the crisis level and provide early warnings to the system. The forecast shows that in the next 48 hours, a pre-emergency level may be exceeded, and the risk for the overall area will be high. In this phase and according to the scenario, the authorities issue a general alert informing the public and specific instruction alert for the first responders. Public alert functionality of the PSAP platform has been enriched with a mechanism for easily selecting content, location, and recipients for the alert. In the final version of the system, public alerts can be limited and broadcasted to specific user groups and get visible only for them.

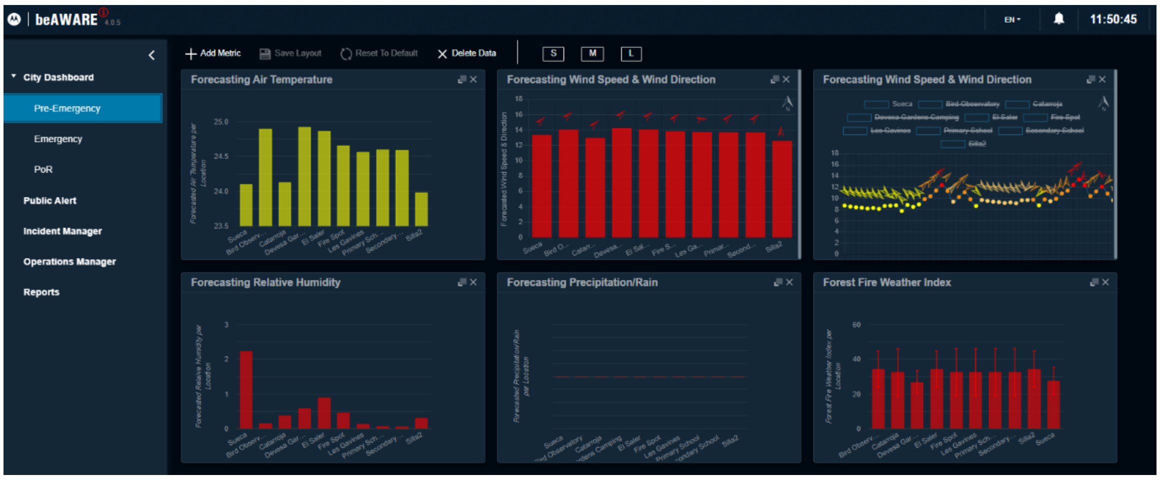
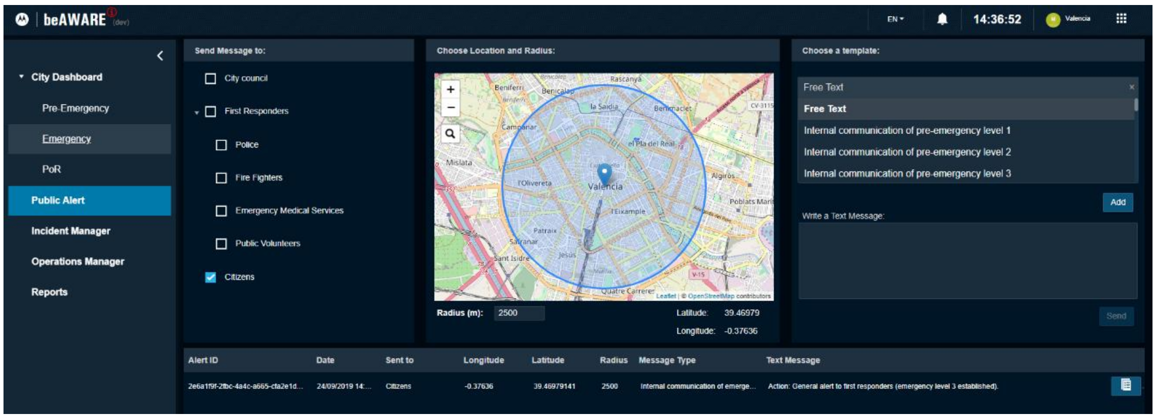

beAWARE
EN
11:50:45
City Dashboard
Pre-Emergency
Emergency
PoR
Public Alert
Incident Manager
Operations Manager
Reports
Add Metric
Save Layout
Reset To Default
Delete Data
S
M
L
Forecasting Air Temperature
Forecasting Wind Speed & Wind Direction
Forecasting Wind Speed & Wind Direction
Forecasting Relative Humidity
Forecasting Precipitation/Rain
Forest Fire Weather Index

Figure 32: Pre Emergency Dashboard

GPS registration of the first responders is an important functionality of the platform that is provided by the mobile application for tracking users' location continuously. In the final version of the Mobile app, FRs groups can additionally declare their profession, status, other textual information, etc. (see D7.7). This information is propagated to the authorities through the PSAP to help them manage better their resources.



Send Message to: Citizens [✓]

Radius (m): 2500
Latitude: 39.46979
Longitude: -0.37636

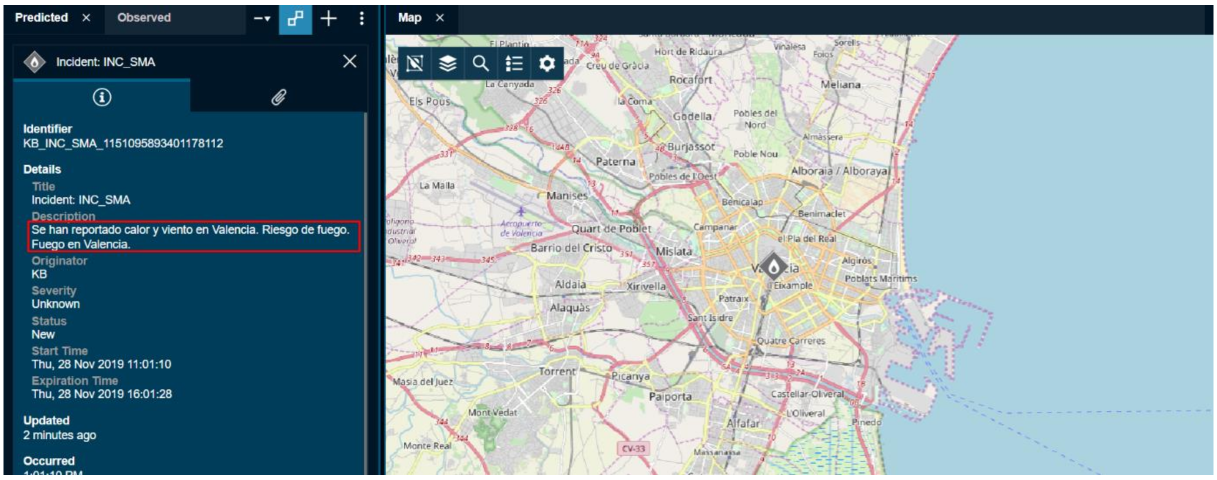
Choose a template:
- Free Text
- Internal communication of pre-emergency level 1
- Internal communication of pre-emergency level 2
- Internal communication of pre-emergency level 3

Alert ID	Date	Sent to	Longitude	Latitude	Radius	Message Type	Text Message
2e6a1f9f-2fbc-4a4c-a665-cfa2e1d...	24/09/2019 14:...	Citizens	-0.37636	39.46979141	2500	Internal communication of emerge...	Action: General alert to first responders (emergency level 3 established).

Figure 33: Example of the puplic Alert mechanism.

One of the tool that is used for the crisis management planning is the KB interface. Through the synthetisation of internal knowledge (Risk Maps e.g., EFFIS) and external knowledge (the location of all known hydrants, schools, points of interest from WikiData), valuable knowledge is visualised on the map. An enhanced GIS-based multi-hazard mapping tool provides a composite picture of natural, historical hazards in varying magnitude, frequency, and areas of effect, critical elements relevant to the damaging impacts of a critical event such as a wildfire, to the authorities.

In the phase of pre-emergency, some initial incidents start to visualise on the map coming from crowdsourcing information from the Twitter users. To ensure the quality of the automatically collected information from the Twitter and protect the system from malicious acts, a three-step filtering process is followed. The content of each tweet is textually and visually analysed. The analysed information is used to categorise the tweets per incident type relevancy and location. Subsequently, the tweets are spatially clustered and sent to the MRG component to generate linguistic situational reports to summarise the information included. The twitter crawler is continuously running looking for relevant crowdsourcing information and reports are automatically generated from this module in all the phases of the emergency.



Incident: INC_SMA

Identifier
KB_INC_SMA_1151095893401178112

Details
Title
Incident: INC_SMA
Description
Se han reportado calor y viento en Valencia. Riesgo de fuego. Fuego en Valencia.
Originator
KB
Severity
Unknown
Status
New
Start Time
Thu, 28 Nov 2019 11:01:10
Expiration Time
Thu, 28 Nov 2019 16:01:28

Updated
2 minutes ago

Occurred

Figure 34: Example of a situational linguistic report of a crowdsourcing incident.

When an information initially ingests to the system and before the analysis components process the collected data, the incident is visualised on the map by the general information icon Its location is spotted on the map while only some initial details are provided. Gradually, and as new analysed information is added, the descriptive details in the property box become richer and the icon of the incident changes to reflect the updated status. In this sense the Incident type and the severity can be easily visualised.

At the end of this phase professional responders are requested through their mobile applications to undertake specific tasks in order to survey the forestall area and guidelines are sent from the PSAP to the teams for applying risk mitigation measures.

6.2 Phase 2 - Emergency phase

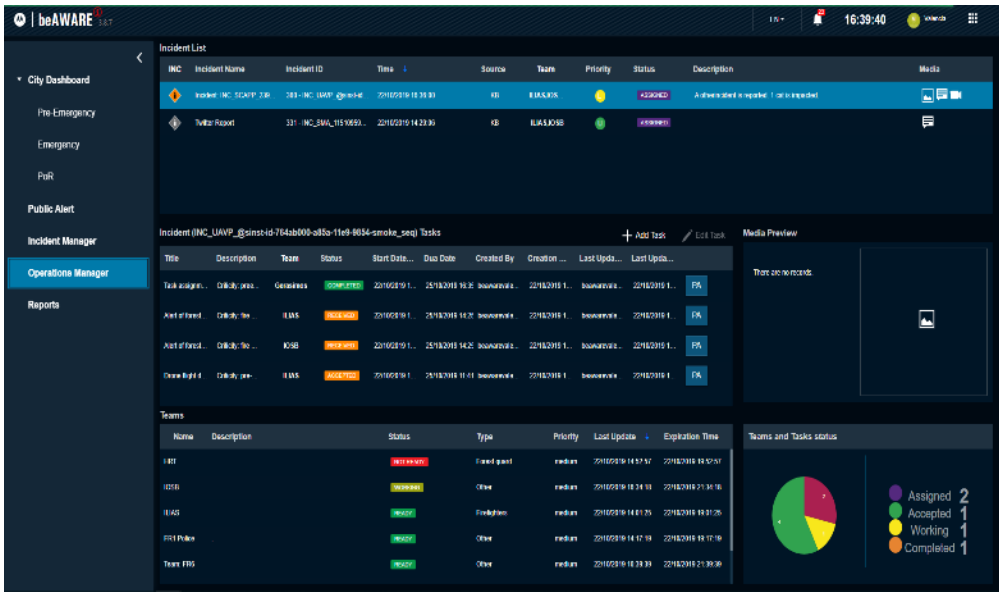
The next phase is broken down into two parts.

In the first part, the introduction of the emergency situation is caused by the detection of smoke in the area.

In the final version of beAWARE system the Image and Video analysis component have been updated with the addition of smoke recognition capabilities. Therefore, we have chosen for the fire scenario, the first indication in the system to arrive in the form of visual content coming from the drones' platform. The drones' platform is connected to the beAWARE platform, according to the generic beAWARE architecture and the main operations that perform are the route planning and the autonomous dynamic piloting, with making the data collected from aerial imagery available to the Drone Analysis (DA) component.

At the final version of the platform, the DA has been extended to handle both image and video sequences coming from the drones. Furthermore, the classification model that is used by the analysis tool, has been updated to include fire and smoke categories, apart from flood that was supported in the previous prototype. Figure 13 shows an example of a frame in a drones' video sequence classified as 'smoke'.

Another module that is demonstrated in this phase is the Automatic Speech Recognition component. The ASR component is used in combination with Multilingual Text Analyser (MTA) in order to automatically extract information from emergency calls and audio messages. ASR receives audio messages, either through the Mobile App or as emergency calls to a dedicated call center and provide transcriptions, which are forwarded to MTA for semantic extraction and geographic information retrieval. In the final version of the platform, MTA is capable to detect concepts and locations found in knowledge and geographical databases, beyond pre-scripted lists that were used to facilitate the execution of the second pilot.



Incident List

INC	Incident Name	Incident ID	Time	Source	Team	Priority	Status	Description	Media
	Incident: INC_SCAPP_20...	380 - INC_UAVP_@sinst-id...	22/10/2019 18:36:30	KB	ILIAS,IOS...		ASSIGNED	A other incident is reported. 1 cat is impacted	
	Twitter Report	331 - INC_SMA_1151065...	22/10/2019 14:23:06	KB	ILIAS,IOSB		ASSIGNED		

Incident (INC_UAVP_@sinst-id-764ab000-a85a-11e9-8054-smoke_seq) Tasks

Title	Description	Team	Status	Start Date...	Due Date	Created By	Creation ...	Last Upda...	Last Upda...	
Task assignm...	Criticity: pres...	Geosimes	COMPLETED	22/10/2019 1...	25/10/2019 18:35	beawareadm...	22/10/2019 1...	beawareadm...	22/10/2019 1...	FA
Alert of forest...	Criticity: fire ...	ILIAS	RECEIVED	22/10/2019 1...	25/10/2019 14:25	beawareadm...	22/10/2019 1...	beawareadm...	22/10/2019 1...	FA
Alert of forest...	Criticity: fire ...	IOSB	RECEIVED	22/10/2019 1...	25/10/2019 14:25	beawareadm...	22/10/2019 1...	beawareadm...	22/10/2019 1...	FA
Drone flight d...	Criticity: pre-...	ILIAS	ACCEPTED	22/10/2019 1...	25/10/2019 11:41	beawareadm...	22/10/2019 1...	beawareadm...	22/10/2019 1...	FA

Teams

Name	Description	Status	Type	Priority	Last Update	Expiration Time
HRT		NOT READY	Forest guard	medium	22/10/2019 14:57:57	22/10/2019 19:57:57
IOSB		WORKING	Other	medium	22/10/2019 18:34:18	22/10/2019 21:34:18
ILIAS		READY	Firefighters	medium	22/10/2019 14:01:25	22/10/2019 19:01:25
FR1 Police		READY	Other	medium	22/10/2019 14:17:19	22/10/2019 19:17:19
Team FR6		READY	Other	medium	22/10/2019 18:39:39	22/10/2019 21:39:39

Figure 35 – Operations Manager

The final versions of the analysis modules can detect incidents in the multiple types of inputs (audio, text, images, video). The outcome of the analysis contributes to the detection of emergencies that are finally visualised on the PSAP. Every new incident that arrives at the PSAP can be handled easily by the Operations Manager (See Figure 35) The Operations Manager is the tool for assigning tasks to the teams to manage incidents and monitor their progress. In the final version of the system, the Operations Manager is capable of propagating additional information to the FRs via their mobile apps, such as descriptions, instructions, media, location. Moreover, several Operations Manager tabs can be opened in different browsers in different PCs so that the head manager of each team group to assign tasks efficiently to its members according to their profession.

In the Second part of the emergency phase the emergency situation is worsening.

beAWARE system collects and combines data from incidents reported by rescuers in the field or citizens that are in danger. As more incidents reach to the system, they are getting clustered to the previous one enriching the enclosed information, extending the linguistic reports, increasing the severity of the clusters that are displayed on the PSAP. This severity is chromatically illustrated through the relevant icons that are displayed on the Map and can be easily interpreted by the users.

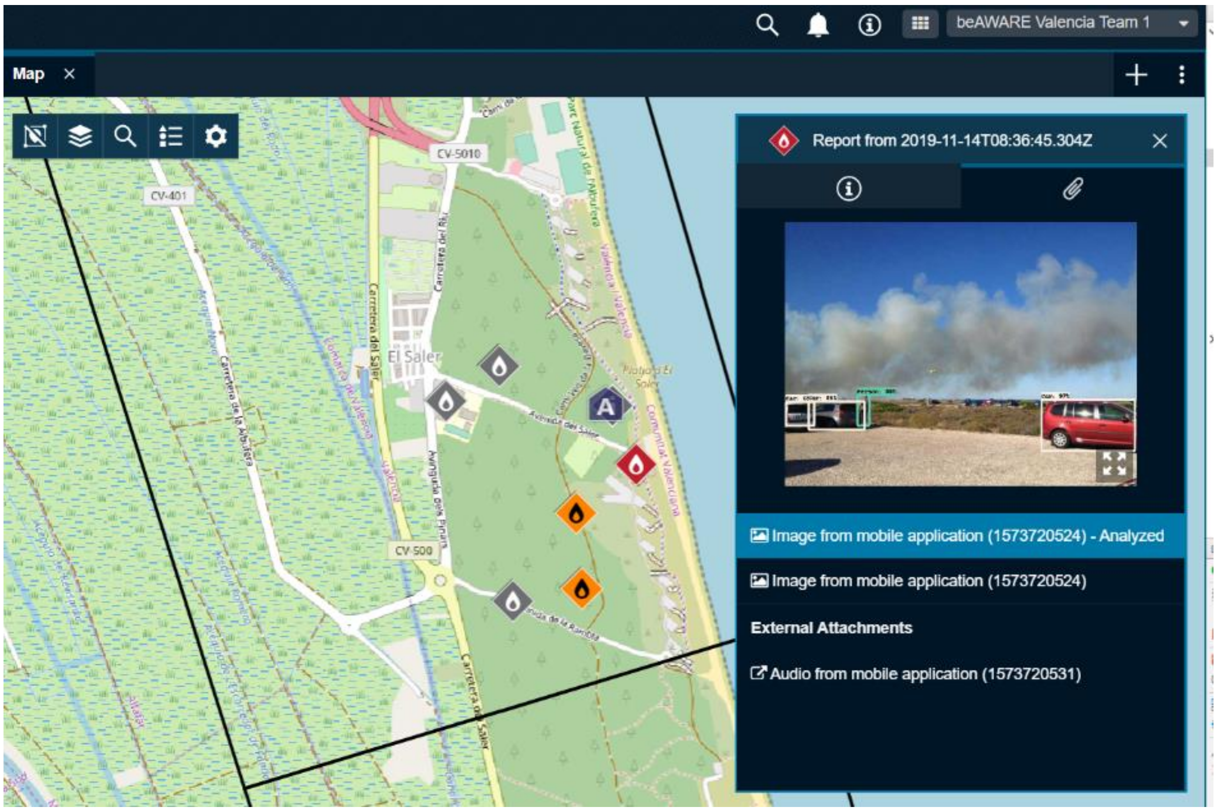

beAWARE Valencia Team 1
Map
Report from 2019-11-14T08:36:45.304Z
Image from mobile application (1573720524) - Analyzed
Image from mobile application (1573720524)
External Attachments
Audio from mobile application (1573720531)
El Saler
CV-5010
CV-401
CV-500

Figure 36: Example of a cluster of Incidents.

The preventive measures are assigned semi-automatically to the rescue teams through the task manager and the task assignment forms. The task assignment form is an extension to the incident view, which allows the operations manager to assign an incident to one or more response teams. Figure 23 show the form for assigning tasks to teams and sending public alert to the task location area. The alerts are received as a geo-fence virtual perimeter for a real-world geographic area (a radius around a point location) and trigger notifications to the mobile app of the users when crossing the physical area that is defined by the geo-fences.

One of the recommendations we received after the second pilot was to consider running the beAWARE system offline and apply the option to integrate data manually. To this end, the knowledge base has been expanded with an input "offline" form, which allows the information to be entered manually into the system. This feature can be used, e.g. when citizens or first responders report incidents not directly to the beAWARE platform via the mobile app. For example, if the first correspondents inform the authorities via radio, this can be introduced on the platform using the "offline" forms. This insertion mode not only ensures that the data is available within the semantic model, but also activates the internal data stream to alert other data about the recently available data (Figure 37).

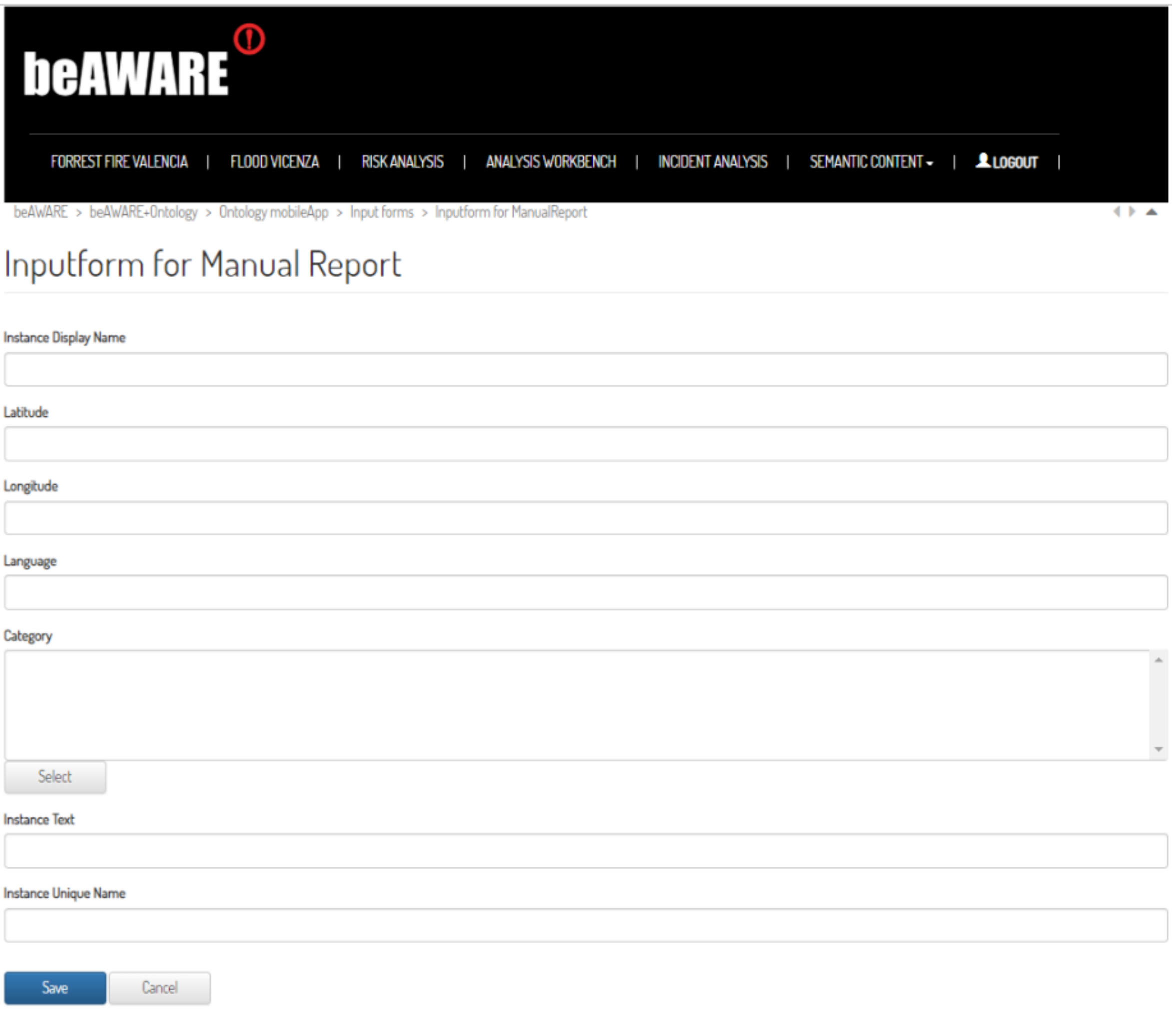

beAWARE
FORREST FIRE VALENCIA | FLOOD VICENZA | RISK ANALYSIS | ANALYSIS WORKBENCH | INCIDENT ANALYSIS | SEMANTIC CONTENT | LOGOUT |
beAWARE > beAWARE+Ontology > Ontology mobileApp > Input forms > Inputform for ManualReport
Inputform for Manual Report
Instance Display Name
Latitude
Longitude
Language
Category
Select
Instance Text
Instance Unique Name
Save
Cancel

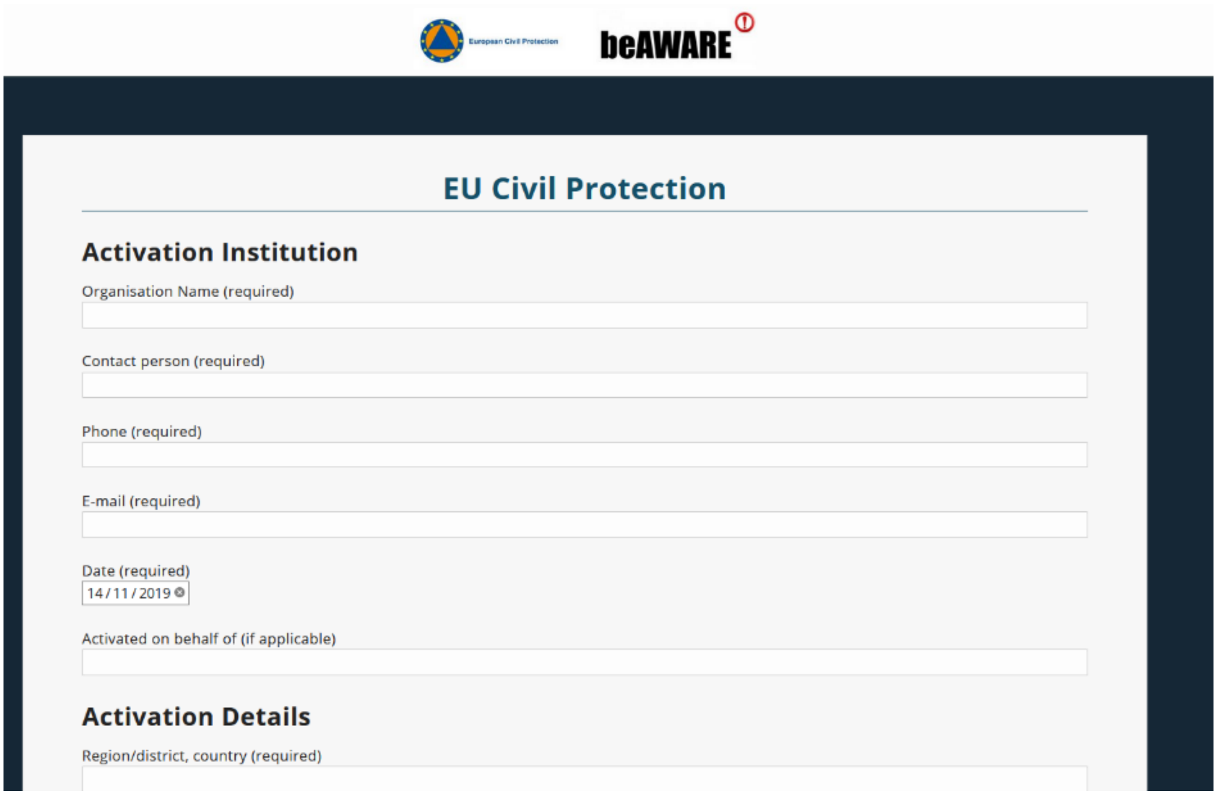
Figure 37: "Offline" manual report.

This phase finishes with the establishment of a supportive advanced command post closer to the emergency area from which the teams will be better organised. A portable or "small scale" PSAP will be easily installed in situ providing the same operational picture to both headquarter and the Supportive command post.

6.3 Phase 3 – Evacuation Phase

The third phase includes a change in the direction of the wind. To simulate this phase a set of artificial weather data are fed into the Real-Time Monitoring and Risk Assessment component which processes this data and send the results to the PSAP.

From that point the situation is judged as very critical since the fire could spread to the north and threaten the educational center that is situated in the area. At that point the decision maker requests for the active engagement of the EU civil protection mechanism. Figure 38 show the form that have been created within the framework of the beAWARE project for the involvement of the EU civil protection mechanism.



EU Civil Protection

Activation Institution

Organisation Name (required)

Contact person (required)

Phone (required)

E-mail (required)

Date (required)
14/11/2019

Activated on behalf of (if applicable)

Activation Details

Region/district, country (required)

Figure 38: EU Civil Protection Form

After this point the progression of fire is visualised on the map by the pins created on the map from user reports and drones' feedback giving aerial images of the fire and the detection of people and vehicles around the spot. Figure 39 illustrates how the fire progression is visualised and how the fire front is moving towards to a specific direction.

A mandatory evacuation order is issued via the alert mechanism of the beAWARE. All operational actions are taken over by the advanced command post that was established in the forestall area during the previous emergency phase.

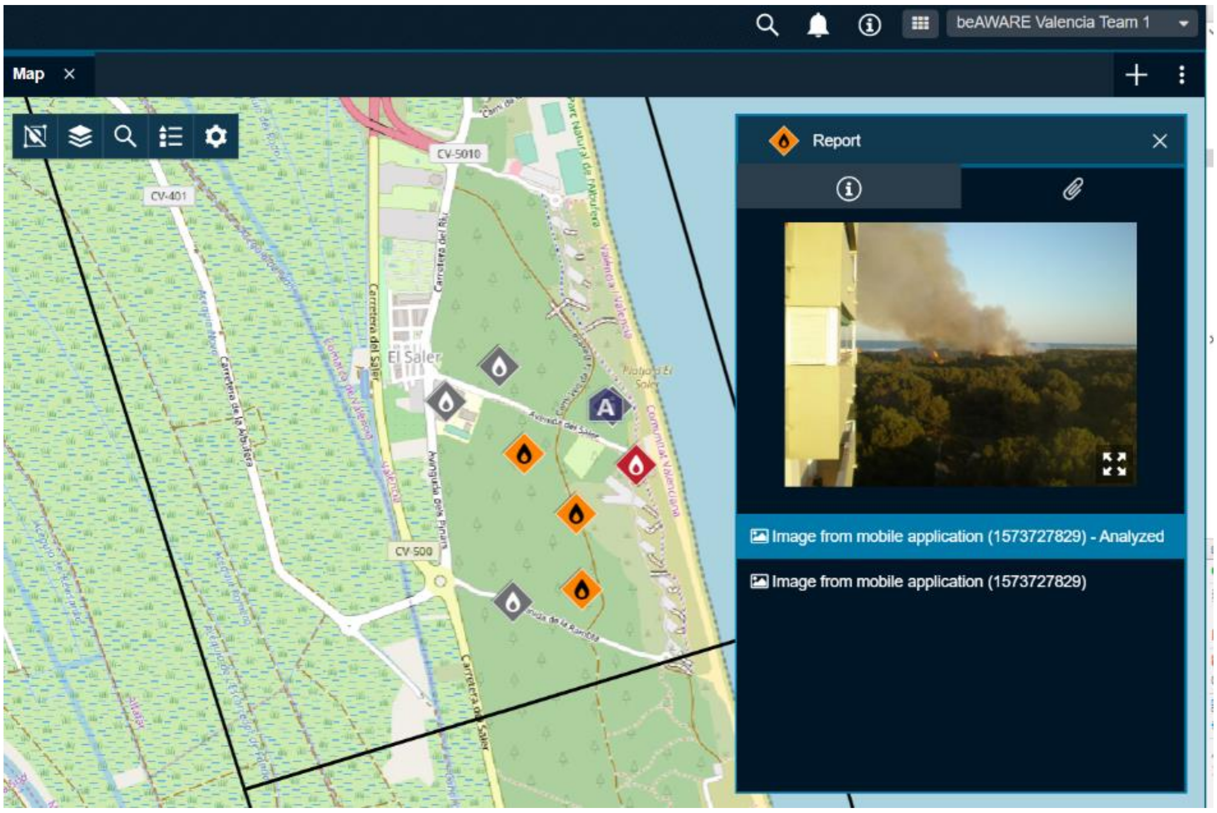
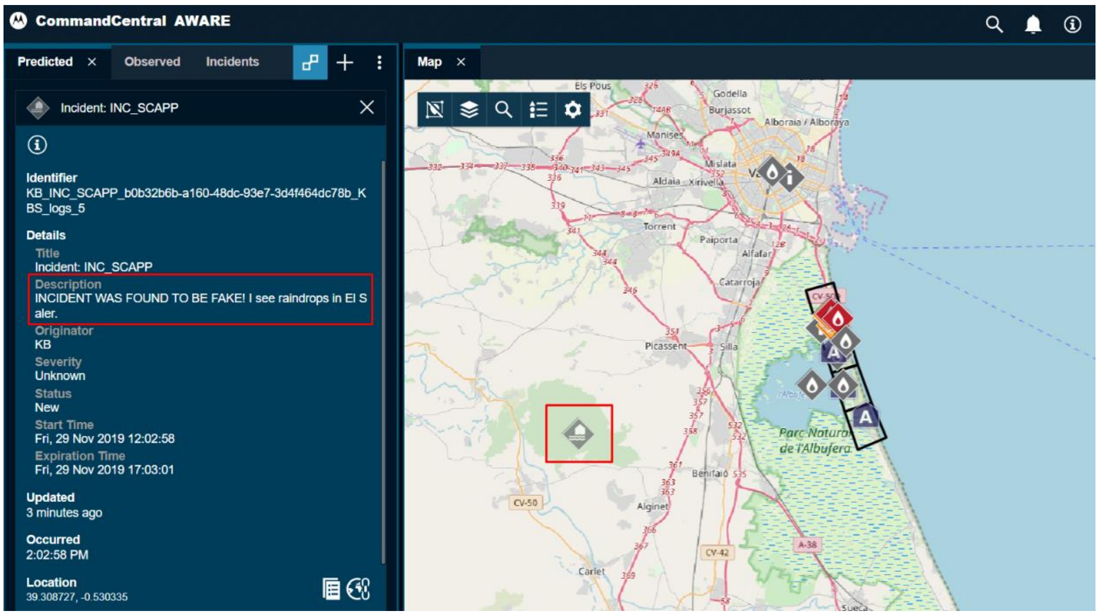

beAWARE Valencia Team 1
Map
Report
Image from mobile application (1573727829) - Analyzed
Image from mobile application (1573727829)
El Saler
CV-5010
CV-401
CV-500

Figure 39: Example of the escalation of the situation.

For this session of the pilot, a number of tasks are planned to be assigned to the FRs' teams realising the prescribed scenario of the fire emergency through the standard operating procedure. The group of units that will be involved are the firefighters taking fire prevention tasks, the police units to regulate the traffic and protect the public, a civil protection volunteer team and a team to provide medical services. In the final release of the system, through an effort to provide more thorough visual analysis reports to the beAWARE system we extended the object detection functionality to include the detection of animals and wheelchair users separately from the generic "person" category. Figure 5 presents some example of the Visual Analysis components for these additions.

As the emergency rapidly escalates it is foreseen for this phase the system to collect a large amount of information from different sources overflowing from a massive volume of data. At this point we have selected to demonstrate a use case of an intentional misleading action from users providing incorrect information to the system. To handle this case an additional validation tool has been implemented and integrated to the beAWARE. The second validation layer (or Validator, VAL) works alongside with Knowledge Base Service (KBS) and the Crisis Classification module (CRCL) to detect erroneous instances. VAL is a standalone module which crosschecks every message that flows into the system with the crisis classification component in order to determine if the input is valid. In detail, what is checked is whether the declared incident type (e.g. Fire, Flood, Precipitation, Heatwave, …) is in accordance with the metrics provided by CRCL for the location and time of the particular instance. The validation results are propagated to PSAP with a flag of "FAKE" when necessary (see Figure 40).



Incident: INC_SCAPP

Identifier
KB_INC_SCAPP_b0b32b6b-a160-48dc-93e7-3d4f464dc78b_KBS_logs_5

Details
Title
Incident: INC_SCAPP
Description
INCIDENT WAS FOUND TO BE FAKE! I see raindrops in El Saler.
Originator
KB
Severity
Unknown
Status
New
Start Time
Fri, 29 Nov 2019 12:02:58
Expiration Time
Fri, 29 Nov 2019 17:03:01

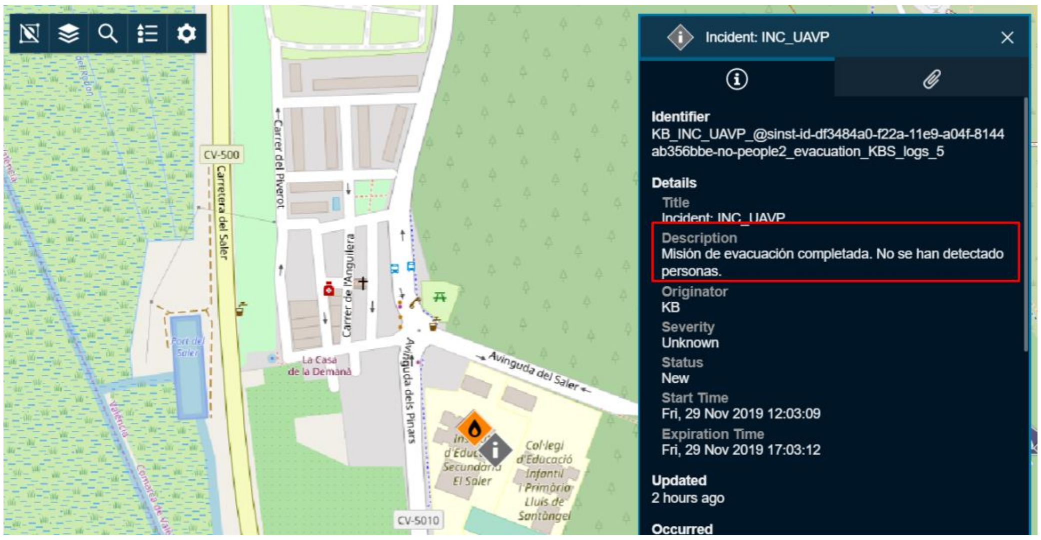
Updated
3 minutes ago

Occurred
2:02:58 PM

Location
39.308727, -0.530335

Figure 40: Example of a Fake Incident detected from the Validator component.

This phase ends with the performance of the evacuation task from the Drones. This task has been added at the final prototype in order to assist authorities at the evacuation mission. DA detects if people are still present in an area under evacuation and creates relevant reports about the status of the mission such as the "evacuation mission is completed. People were detected" or "not detected" depending on the visual content that is recorded from the drones. (See Figure 41)



Incident: INC_UAVP

Identifier
KB_INC_UAVP_@sinst-id-df3484a0-f22a-11e9-a04f-8144ab356bbe-no-people2_evacuation_KBS_logs_5

Details
Title
Incident: INC_UAVP
Description
Misión de evacuación completada. No se han detectado personas.
Originator
KB
Severity
Unknown
Status
New
Start Time
Fri, 29 Nov 2019 12:03:09
Expiration Time
Fri, 29 Nov 2019 17:03:12

Updated
2 hours ago

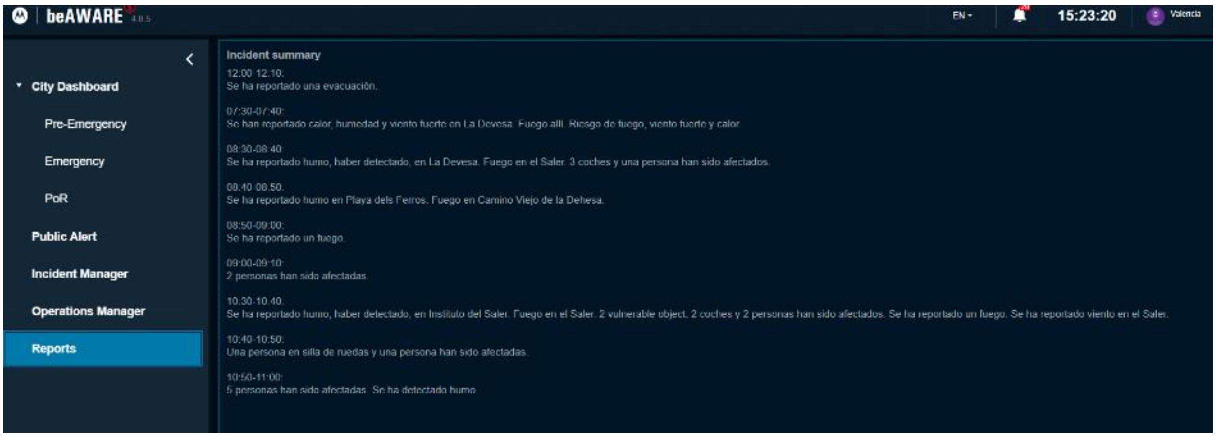
Occurred

Figure 41: Example of the report after the Evacuation Mission of the Drones.

6.4 Fade Out

The purpose of this phase is to demonstrate how the beAWARE supports direct and easy communication, between national authorities, rescue teams and citizens and facilitates the distribution of information even in the final phase of the emergency management cycle

At this phase FR teams after completing their tasks, switch their tasks status to completed. The Authorities remove the alert by sending a public message through the beAWARE platform: Recovery continues until all systems return to normal.



Incident summary

12:00 12:10.
Se ha reportado una evacuación.

07:30-07:40:
Se han reportado calor, humedad y viento fuerte en La Devesa. Fuego allí. Riesgo de fuego, viento fuerte y calor.

08:30-08:40:
Se ha reportado humo, haber detectado, en La Devesa. Fuego en el Saler. 3 coches y una persona han sido afectados.

08:40 08:50.
Se ha reportado humo en Playa dels Ferros. Fuego en Camino Viejo de la Dehesa.

08:50-09:00:
Se ha reportado un fuego.

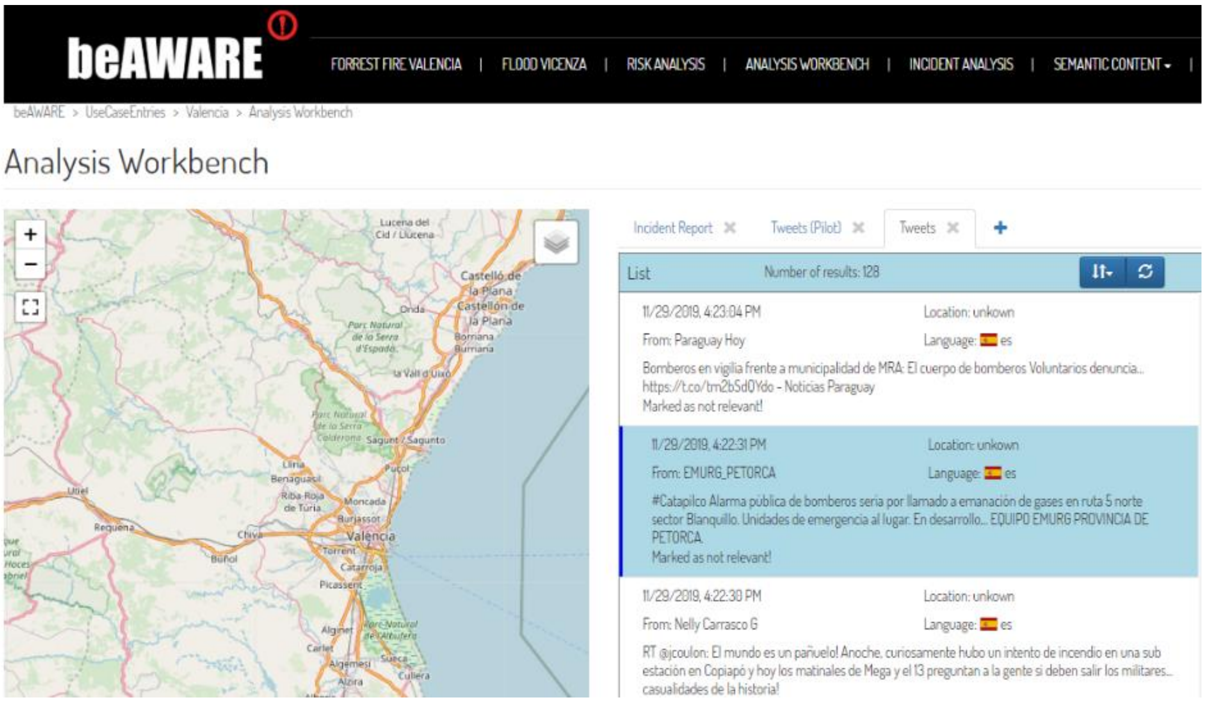
09:00-09:10:
2 personas han sido afectadas.

10:30-10:40.
Se ha reportado humo, haber detectado, en Instituto del Saler. Fuego en el Saler. 2 vulnerable object. 2 coches y 2 personas han sido afectados. Se ha reportado un fuego. Se ha reportado viento en el Saler.

10:40-10:50:
Una persona en silla de ruedas y una persona han sido afectadas.

10:50-11:00:
5 personas han sido afectadas. Se ha detectado humo.

Figure 42:Example of the summary report.


beAWARE
FORREST FIRE VALENCIA | FLOOD VICENZA | RISK ANALYSIS | ANALYSIS WORKBENCH | INCIDENT ANALYSIS | SEMANTIC CONTENT
beAWARE > UseCaseEntries > Valencia > Analysis Workbench
Analysis Workbench
Castelló de la Plana
Sagunt / Sagunto
Valencia
Requena
Incident Report
Tweets (Pilot)
Tweets
List

Number of results: 128

11/29/2019, 4:23:04 PM — Location: unkown
From: Paraguay Hoy — Language: es
Bomberos en vigilia frente a municipalidad de MRA: El cuerpo de bomberos Voluntarios denuncia... https://t.co/tm2bSd0Ydo - Noticias Paraguay
Marked as not relevant!

11/29/2019, 4:22:31 PM — Location: unkown
From: EMURG_PETORCA — Language: es
#Catapilco Alarma pública de bomberos seria por llamado a emanación de gases en ruta 5 norte sector Blanquillo. Unidades de emergencia al lugar. En desarrollo... EQUIPO EMURG PROVINCIA DE PETORCA.
Marked as not relevant!

11/29/2019, 4:22:30 PM — Location: unkown
From: Nelly Carrasco G — Language: es
RT @jcoulon: El mundo es un pañuelo! Anoche, curiosamente hubo un intento de incendio en una sub estación en Copiapó y hoy los matinales de Mega y el 13 preguntan a la gente si deben salir los militares... casualidades de la historia!

Figure 43: Analysis Workbench

Last, a wrap-up summary provides an overview of the crisis. The wrap-up summaries are generated semi-manually on User’s request (see Figure 42). Moreover, the Analysis Workbench tool can be used for consolidating the results of the previous phases, displaying the total number of incidents, the people that were involved, the severity and the location per incident, the source of each incident report, the number of the relevant or irrelevant tweets, the incidents that were automatically filtered out as fake etc.(Figure 43).

7 Summary and Conclusions

This deliverable presents the status of the Final beAWARE System. In addition, it includes some updates regarding the architecture and the integration of all the services. This document presents the main functionalities of the final version of the beAWARE platform in detail.

This deliverable summarizes the system capabilities, individual components, interactions among components and flows, as well as the deployment and hosting infrastructure. The final version of the system was used to run the last use case; in summary, all pilots were successfully run using the platform.

The process for reaching the final version of the platform started off with the use case requirements document to ensure that the implemented platform covers the stated requirements and used the pilots as validation scenarios. System requirements were derived from the use case requirements, which in turn contribute to the specification of the capabilities of the different system components. The bulk of this document provides an overview of the different platform components and highlights the interfaces and dependencies between them.

8 Appendix 1: System Functionalities - final system

beAWARE components	Final System
KB	• Ontology changes as needed • KB Analysis Dashboard • KBS Additional Validation Services.
FROST server	• Event/Threshold detection in alignment with crisis classification
Crisis Classification	• Assess the risk and crisis severity level based on data from heterogeneous sources (e.g. photos, text messages, IoT, sensors data, social media, mobile Application). • Risk assessment services & real-time monitoring system to identify crisis event's severity level (full functionality of the Real-Time Monitoring and Risk Assessment component for the fire and heatwave pilots) • Interoperability with European Forest Fire Information System (EFFIS), European Flood Awareness System so as to integrate risk/impact maps
Image analysis	• Recognition of more object categories • Final updates of current implementations • Detect animals • Detect people in wheelchairs
Video analysis	• Recognition and tracking of more object categories • Final updates of current implementations • Extend Object Detection and Tracking to include animal detection and people in wheelchair
ASR	• Fine tune language models (in order to improve recognition accuracy) • Include denoising techniques • Connect to the legacy call centre
Drones Platform	• Support sending video in RT • Send video to multiple destinations
Social Media	• Communication with MTA to get extracted location of tweets • Annotation for clustering
Text analysis	• Produce references to BabelNet and geographical databases, and map these references to ontological concepts. • Semantic abstraction over entities, locations, events and concepts. • Produce conceptual structure integrating various aspects of domain and linguistic meaning (e.g. modality, tense). • Geolocation • Relation extraction strategy
Report generator	• Produce summaries • Integrate reference to BabelNet and geographical databases into the generation process. • Apply statistical generator base on UD
Mobile application	• Extended team management together with User/Role Management • Task management • Include Attachments in the Task Reports.

beAWARE components	Final System
PSAP	• Extend Dashboard capabilities to alert the authorities when exceeding predefined thresholds • Enhance Metrics and map Visualization by adding more details and special icons • Enhance Operational Management to track task progress, modify tasks, trigger events and alert the user following predefined rules and conditions • Incident Management: edit and update incident details like Priority, certainty, status etc • Standard Humanitarian Icons