beAWARE

Enhancing decision support and management services in extreme weather climate events

700475

7.9

Final technical evaluation report

Dissemination level:	Public
Contractual date of delivery:	Month 36, 31 December 2019
Actual date of delivery:	Month 36, 31 December 2019
Work package:	WP 7: System development, integration and evaluation
Task:	T7.3-Overall Technical Testing of beAWARE platform
Type:	Report
Approval Status:	Final version
Version:	V0.5
Number of pages:	69
Filename:	D7.9_beAWARE_Final_technical_evaluation_report_2019-12-31_v0.5.docx

Abstract

This document comprises the technical evaluation of the components in beAWARE System. This deliverable is iterative and the current version corresponds to the final release compiled in M36. This document details the technical aspects of the outcome of the final pilot from a technical performance perspective. The document is structured in two parts. The first part details the performance indicators used. The second part presents the current evaluation of the system according to the performance indicators defined in the first part.

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	26.11.2019	Document initiation and assignments distribution	CERTH
V0.2	10/12/2019	Incorporate contributions from IBM & IOSB	IBM+IOSB
V0.3	16/12/2019	Incorporate contributions from CERTH & MSIL	CERTH & MSIL
V0.4	20/12/2019	Incorporate contributions from UPF	UPF
V0.5	21/12/2019	Internal Review	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
CERTH	Gerasimos Antzoulatos	gantzoulatos@iti.gr
CERTH	Anastasios Karakostas	akarakos@iti.gr
MSIL	Itay Koren	itay.koren@motorolasolutions.com
UPF	Gerard Casamayor	gerard.casamayor@upf.edu

Executive Summary

This deliverable contains the technical evaluation of the final integrated beAWARE platform. This report is the third of an iterative evaluation process of the beAWARE development cycle and together with D7.4 delivered in M18 and D7.6 delivered in M24 consist a set of a three-step evaluation study.

The final version of the platform represents the most important milestone of the project. In the final pilot the full range of the beAWARE technologies were demonstrated. This document aims to report the technical evaluation of the performance of the final product with respect to the final demonstration that took place in Valencia, Spain.

This technical evaluation is based on the assessment plan and the performance indicators that were introduced in D1.1 and D1.3 and were refined in D7.6.

The document is structured in two parts. The first part provides an overview of the beAWARE components along with the indicators selected to measure the performance of each component. The second part presents the results of the evaluation according to the performance indicators defined in the first part.

Abbreviations and Acronyms

ACID	Atomicity, Consistency, Isolation, Durability
AP	Average Precision
AMICO	AAWA's flood forecasting model
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
EFFIS	European Forest Fire Information System.
FPS	Frames per second
FROST-Server	**FR**aunhofer **O**pensource **S**ensor**T**hings-Server
GPU	Graphics Processing Unit
GUI	Graphical User Interface
KB	Knowledge Base
KBR	Knowledge Base Repository
KBS	Knowledge Base Service
K8s	Kubernetes
MS	Milestone
M2M	Machine-to-machine
MSB	Message Bus
MTA	Multilingual Text Analyser
MRG	Multilingual Report Generator
NER	Named Entity Recognition
NMI	Normalized Mutual Information
ObjD	Object detection
OWL	Web Ontology Language
PSAP	Public-safety answering point
P2	Prototype 2
RAM	Random Access Memory
SMA	Social Media Analysis

SMC	Social Media Clustering
UC	Use Case
VRS	Visual River Sensing
Wacc	Word Accuracy
WER	Word error rate
WP	Work Package

Table of Contents

1 INTRODUCTION 9
1.1 Purpose of this document. 9
1.2 Structure of the report. 9
2 OVERVIEW AND EVALUATION METHODOLOGY 10
2.1 Global view 10
2.2 Technical Evaluation Methodology 10
2.3 Topics of Evaluation 10
2.3.1 Social Media Monitoring 10
2.3.2 FROST-Server 12
2.3.3 Communication Bus 12
2.3.4 Technical Infrastructure 13
2.3.5 Crisis Classification 15
2.3.6 Text Analysis 17
2.3.7 Automatic Speech Recognition 21
2.3.8 Visual analysis 22
2.3.9 beAWARE Knowledge Base 23
2.3.10 Multilingual Report Generator 28
2.3.11 Drones Platform 30
2.3.12 Public Safety Answering Point 31
2.3.13 Mobile Application 32
3 TECHNICAL EVALUATION 33
3.1 Social Media Monitoring 33
3.2 Communication Bus 34
3.3 Technical Infrastructure 37
3.4 Crisis classification 41
3.4.1 Evaluation of the Early Warning component 42
3.4.2 Evaluation of the Real-Time Monitoring and Risk Assessment component 43
3.5 Text Analysis 44
3.6 Automatic Speech Recognition 46
3.7 Visual analysis 47
3.7.1 Final Version of the Emergency Classification (EmC) 48
3.8 Drones 51
3.8.1 Drones Platform 51
3.8.2 Drones Analysis 53
3.9 beAWARE Knowledge Base 57
3.9.1 Knowledge Base Service 58
3.10 Multilingual Report Generator 61
3.11 Public Safety Answering Point 64
3.12 Mobile Application 65

4 CONCLUSIONS 68

List of Figures

Figure 1: beAWARE technical infrastructure 14
Figure 2: Kubernetes cluster - worker nodes 14
Figure 3: Kubernetes microservices view 15
Figure 4: architecture of the final version of the text analysis module, as shown in D3.4 18
Figure 5: User interface for the Knowledge Base 24
Figure 6: Visualization of the available GIS data 28
Figure 7. Comparison of clustering techniques, with fine-tuned DBSCAN outperforming 33
Figure 8: Message bus statistics 35
Figure 9: Cloud Object Storage details 35
Figure 10: Object Store statistics 36
Figure 11: Message Bus topics 37
Figure 12: Git Hub repository 38
Figure 13: beAWARE Jenkins 39
Figure 14: beAWARE cloud infrastructure 39
Figure 15: MongoDB instance 40
Figure 16: MySQL instance 41
Figure 17: Execution Time of Early Warning component 43
Figure 18: Speed comparison of Visual Analysis on all three pilots. 48
Figure 19: EmC final version confusion matrix. 48
Figure 20: Images classified as 'fire'. 49
Figure 21: Images classifies as 'smoke'. 49
Figure 22: Images classified as 'other'. 50
Figure 23: smoke detection with drones 52
Figure 24: drone footage of person in danger 52
Figure 25: Drones support the school evacuation 53
Figure 26. Example of an image classified as *smoke* (upper) and an image classified as *other* (lower) 55
Figure 27. Example of a correctly detected target (upper) and a misdetection (lower) 55
Figure 28. A video capture during an evacuation mission at the yard of a school without any trapped people present. 56
Figure 29: Normalized Confusion Matrix over a drone ‘smoke’ video sequence. 57
Figure 30: Kafka bus message processing time (by the KBS), in relation to the incident count 58

Figure 31: Messages processed by KBS broken down to semantic Fusion and Reasoning operations. 59
Figure 32: Fusion and Reasoning duration for “incident validation” messages by the KBS 61

List of Tables

Table 1 65

1 Introduction

1.1 Purpose of this document.

This report details the technical aspects of the outcome of the final system as a part of a cyclic process of prototyping, testing and evaluation that was adopted for the development of the beAWARE platform. This technical evaluation is centred around the performance of the components of the platform, based mainly on the findings of the final pilot which took place in Valencia (Spain) on the 14th of November 2019.

1.2 Structure of the report.

Similar to the previous version, this evaluation report is structured in 4 sections.

The second section presents the methodology used for the technical evaluation of the components. Each subsection is divided in two parts devoted to: 1) a technical overview of each component with a focus on the last additions and 2) the indicators used to evaluate their performance.

In section 3 the results of the technical evaluation are presented mainly based on the input of the final beAWARE pilot that took place in Valencia.

Last, Section 4 presents the conclusions obtained by the elaboration of the evaluation methodology

2 Overview and Evaluation Methodology

2.1 Global view

The beAWARE architecture is roughly made up of the following conceptual layers:

1. **Ingestion layer**, containing mechanisms and channels through which data is brought into the platform; Within this layer we can classify two modules: The Social Media Monitoring and the FROST- Server. (Section 2.3.1 & Section 2.3.2).
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. (Sections 2.3.3 & Section2.3.4).
3. **Business layer**, containing the components that perform the actual platform-specific capabilities. (Sections 2.3.5 - 0).
4. **External facing layer**, including the mobile application and PSAP (Public-safety answering point), interacting with people and entities outside the platform. (Sections 2.3.12 & 0)

2.2 Technical Evaluation Methodology

The evaluation is based on the assessment plan and the performance indicators that were introduced in D1.1 & D1.3 and furtherly refined in D7.6.

2.3 Topics of Evaluation

2.3.1 Social Media Monitoring

Social Media Monitoring comprises two individual modules: Social Media Analysis (SMA) for crawling and validating Twitter posts and Social Media Clustering (SMC) for grouping tweets in a spatiotemporal manner.

As it has been described in previous deliverables, SMA collects tweets in languages of interest (i.e., English, Italian, Greek, and Spanish) that contain preselected keywords in relation to flood, fire, and heatwave incidents, by using Twitter's Streaming API[1]. After the crawling of posts, a three-step validation process, which was introduced in the second prototype, aims to filter out fake or irrelevant tweets. The first step concerns the detection of fake posts, the

[1] [1] https://developer.twitter.com/en/docs/tweets/filter-realtime/overview

second step checks for unrelated emoticons or emojis inside the text, and the third step classifies tweets as relevant or irrelevant to the examined use cases, based on their visual and textual information. Each tweet that is not filtered out by the validation procedure is forwarded to the Multilingual Text Analyzer (MTA) for concept and conceptual relation extraction and to the Knowledge Base Service (KBS) to populate respective incidents.

The SMC component consumes messages from the MTA, in order to base grouping on the location detected by this module. When a sufficient number of tweets are collected or significant time passes since the last received tweet, SMC performs spatial clustering. When it is completed, the clusters are presented as separate HTML files, which are called Twitter Reports. Each Twitter report contains the list of tweets it comprises and is sent to the KBS so as to create a corresponding incident. This version of SMC that is connected with MTA and utilizes the extracted locations is first introduced in the final system and, moreover, a first evaluation of the methodology is included in deliverable D4.3 (M35).

With respect to the evaluation of the SMA module's performance, the following indicators are used:

Performance Indicators	Precision, recall, and F-score
Definition	In classification tasks, the precision for a class is the number of true positives divided by the total number of observations labelled as belonging to the positive class. Recall is the number of true positives divided by the total number of observations that actually belong to the positive class. The F-score considers both precision and recall and can be calculated as the harmonic mean of these two measures.
Domain	Machine learning
Range	From 0.0 (0%) to 1.0 (100%)
Limitations	A limitation with respect to the F-score is the fact that one may be unable to distinguish low-recall from low-precision systems.

Moreover, for the evaluation of the SMC module's performance, the next indicator is used:

Performance Indicators	Normalized Mutual Information (NMI)
Definition	The Mutual Information (MI) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" obtained about one random variable through

	observing the other random variable. Normalized Mutual Information (NMI) is a normalization of the MI score to scale the results between 0 (no mutual information) and 1 (perfect correlation).
Domain	Probability theory
Range	From 0.0 to 1.0
Limitations	Whenever the ratio between the number of members and the number of clusters is small the NMI becomes too high which is called “selection bias problem”.

Finally, regarding the evaluation of the SMA and SMC modules in the frame of the third beAWARE pilot, a qualitative assessment is provided.

2.3.2 FROST-Server

To collect and store time series data, the FROST-Server is used in the beAWARE platform. Since there are no sensors available in the pilot region of Valencia, no changes at the FROST-Server itself have been done. To support the pilot, weather measurements and weather forecasts are automatically imported. Since this doesn’t affect the work, already done in D7.6 we refer to the evaluation, there.

2.3.3 Communication Bus

The communication bus serves as a central point of communication between different system components. Its main mode of operation is publish / subscribe, which supports different parts of a composite application to be unaware of each other but still manage to communicate upon need.

The bus is in charge of notifying interested and registered components when new items which are of interest to them have been received or calculated by another component.

The final prototype exhibited a more challenging use of the communication bus with respect to main performance and scalability indicators such as, the amount of topics used, the amount of subscribers and publishers, the rate in which messages were sent through the bus, and the size of messages sent. An important new driver of messages in this prototype is the drones platform which continuously sends messages about video chunks made available throughout the flight of the drone.

With respect to the evaluation of the module’s performance, the following indicators are used:

Performance Indicator	Number of different topics / subscribers / publishers supported

Definition	The bus should support enough such entities as required by the beAWARE system. Tests will vary independently the three dimensions, namely topics, subscribers, and publishers.
Domain	Scalability / elasticity
Range	Values will be tested up to 100 since it's not anticipated that a larger amount would be required
Limitations	n/a

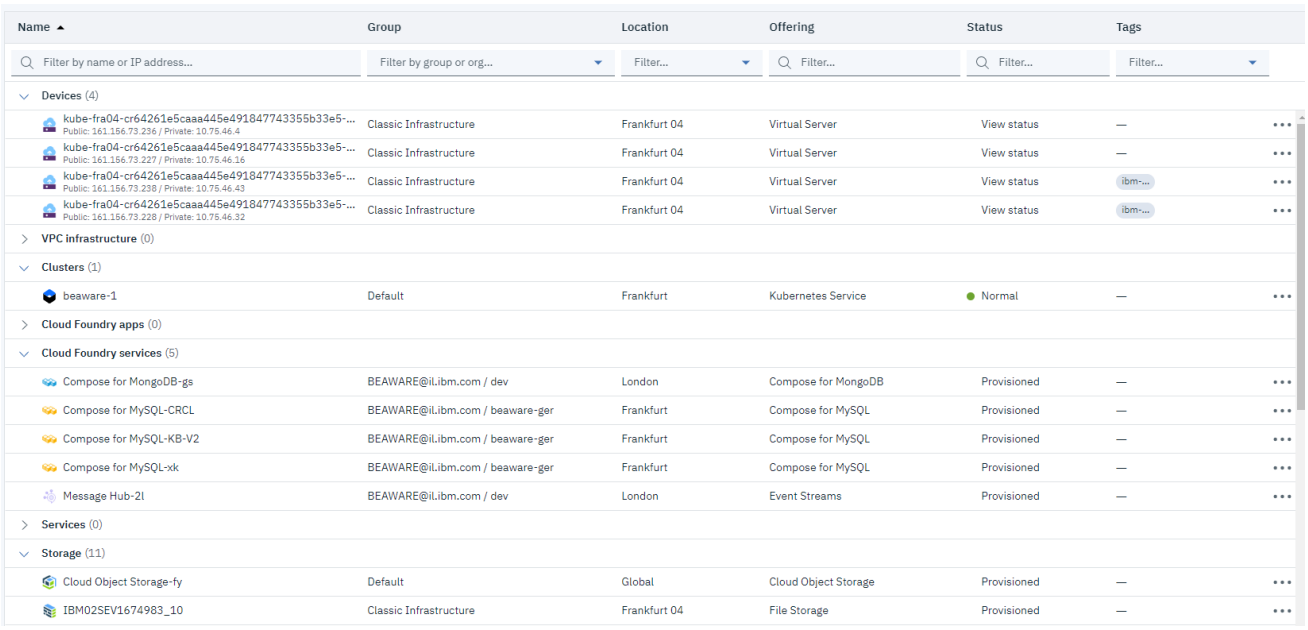
Performance Indicator	Message throughput through the bus
Definition	Amount and length of messages that can be sent through the bus during a certain time range
Domain	Scalability / throughout. Tests will vary independently the three dimensions, namely topics, subscribers, and publishers
Range	Values will be tested up to 100 messages / per second of up to 1 K length messages since it's not anticipated that a larger amount would be required
Limitations	n/a

2.3.4 **Technical Infrastructure**

The technical infrastructure of the beAWARE platform is comprised of a cloud-based Kubernetes cluster which holds all the individual components (microservices) which provide the beAWARE capabilities, in addition to cloud-based services for data storage and messaging.

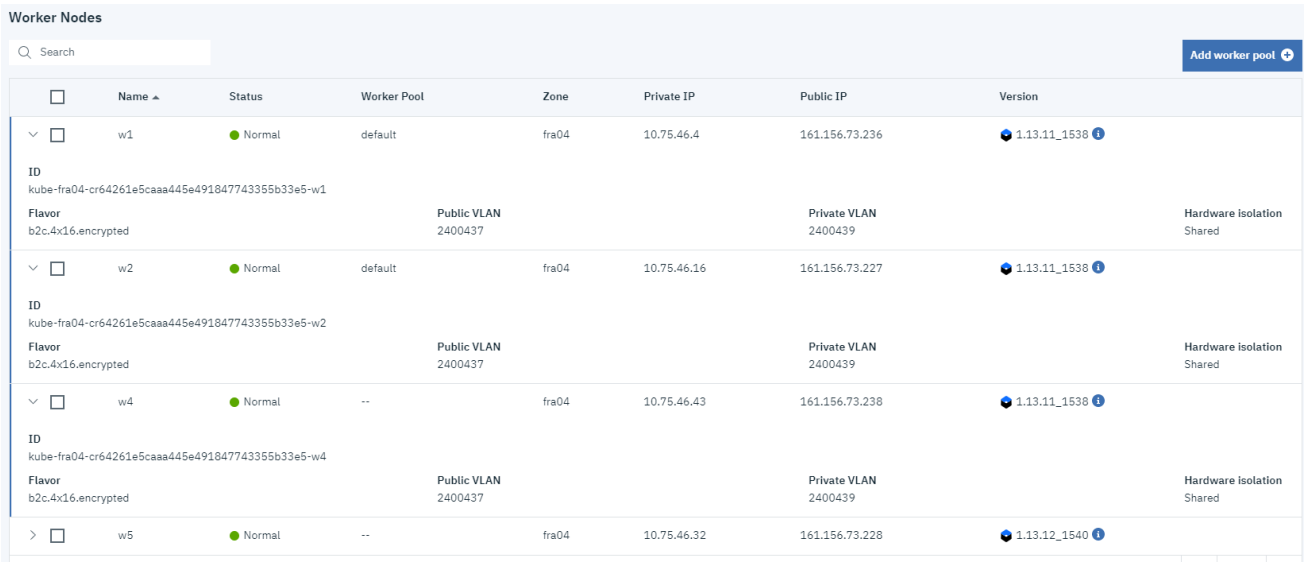
The Kubernetes cluster consists of 4 worker nodes, each one having 4 cores and 16GB of RAM. The worker nodes host all the beAWARE microservices.

In the final demonstration we exercised the technical infrastructure to a much larger degree due to the deployment of more components into the cluster, utilizing more resources, and the deployment of additional back-end services, mainly different kinds of data stores. The main aim is to be responsive to platform components requests as they arrive. Towards the final prototype we enlarged the cluster by adding a new Kubernetes working node due to the growing demand for resources.



Name	Group	Location	Offering	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Devices (4)					
kube-fra04-cr64261e5caaa445e491847743355b33e5-... Public: 161.156.73.236 / Private: 10.75.46.4	Classic Infrastructure	Frankfurt 04	Virtual Server	View status	—
kube-fra04-cr64261e5caaa445e491847743355b33e5-... Public: 161.156.73.227 / Private: 10.75.46.16	Classic Infrastructure	Frankfurt 04	Virtual Server	View status	—
kube-fra04-cr64261e5caaa445e491847743355b33e5-... Public: 161.156.73.238 / Private: 10.75.46.43	Classic Infrastructure	Frankfurt 04	Virtual Server	View status	ibm-...
kube-fra04-cr64261e5caaa445e491847743355b33e5-... Public: 161.156.73.228 / Private: 10.75.46.32	Classic Infrastructure	Frankfurt 04	Virtual Server	View status	ibm-...
VPC infrastructure (0)					
Clusters (1)					
beaware-1	Default	Frankfurt	Kubernetes Service	Normal	—
Cloud Foundry apps (0)					
Cloud Foundry services (5)					
Compose for MongoDB-gs	BEAWARE@il.ibm.com / dev	London	Compose for MongoDB	Provisioned	—
Compose for MySQL-CRCL	BEAWARE@il.ibm.com / beaware-ger	Frankfurt	Compose for MySQL	Provisioned	—
Compose for MySQL-KB-V2	BEAWARE@il.ibm.com / beaware-ger	Frankfurt	Compose for MySQL	Provisioned	—
Compose for MySQL-xk	BEAWARE@il.ibm.com / beaware-ger	Frankfurt	Compose for MySQL	Provisioned	—
Message Hub-2l	BEAWARE@il.ibm.com / dev	London	Event Streams	Provisioned	—
Services (0)					
Storage (11)					
Cloud Object Storage-fy	Default	Global	Cloud Object Storage	Provisioned	—
IBM02SEV1674983_10	Classic Infrastructure	Frankfurt 04	File Storage	Provisioned	—

Figure 1: beAWARE technical infrastructure



Worker Nodes

Search — Add worker pool

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

	ID	Flavor	Public VLAN	Private VLAN	Hardware isolation
w1	kube-fra04-cr64261e5caaa445e491847743355b33e5-w1	b2c.4x16.encrypted	2400437	2400439	Shared
w2	kube-fra04-cr64261e5caaa445e491847743355b33e5-w2	b2c.4x16.encrypted	2400437	2400439	Shared
w4	kube-fra04-cr64261e5caaa445e491847743355b33e5-w4	b2c.4x16.encrypted	2400437	2400439	Shared

Figure 2: Kubernetes cluster - worker nodes

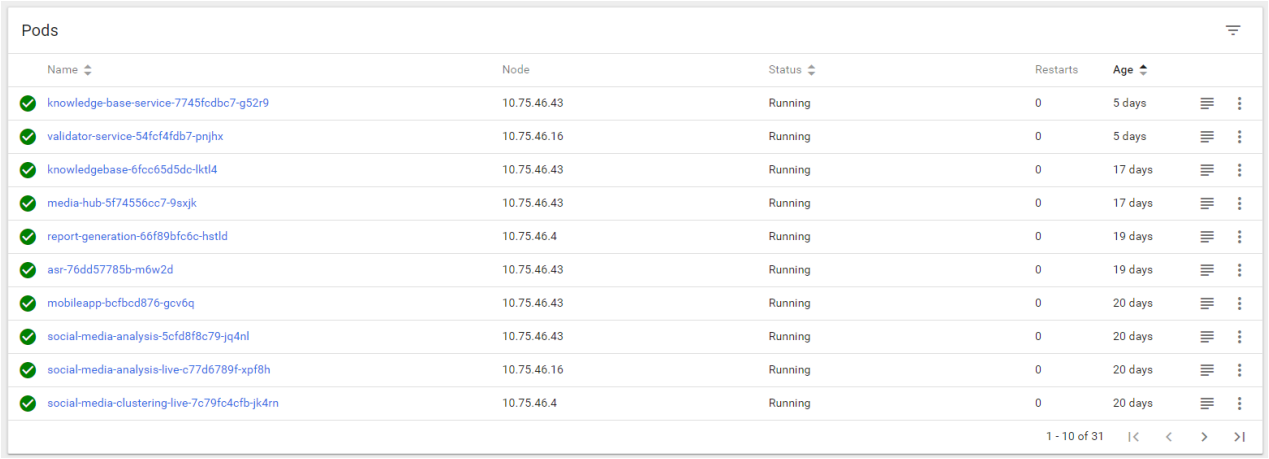


Figure 3: Kubernetes microservices view

To monitor the performance, detect slowdowns and determine data storage efficiency we used the results of the Flood pilot. The results and some instances of the components are presented in section 3.3 .

To monitor the performance during the 3rd pilot as indicative for the final version of the system we monitored the load and latency of the core infrastructure (message bus and object storage) and determined that it supported well the requirements of the individual components and no noticeable delays were observed.

2.3.5 **Crisis Classification**

The Crisis Classification component encapsulates the necessary technology to process the available forecasts from prediction models (weather, hydrological etc.) and data obtained from sensors as well as other heterogeneous sources to estimate the crisis level of a forthcoming event or to monitor an ongoing event. Relying on the results of the analysis, Crisis Classification component generates the appropriate warning alerts to timely notify the authorities as well as the meaningful metrics to support the visualisation tools at the beAWARE's dashboard.

Briefly, the functionalities of the Crisis Classification module established into the earlier phases of the platform, as mentioned in the deliverables D3.1 and D3.4, are the following:

a) *Early Warning* component estimates the crisis level of a forthcoming extreme natural event (heatwave, flood and fire), by relying on the various type of forecasts. The assessment of the severity of the imminent crisis is provided in the whole Region of Interest (global level) along with the assessments in smaller areas.
Furthermore, the mechanism to integrate Flood Hazard maps and Risk/Impact maps is implemented.

b) *Real-Time Monitoring and Risk Assessment* component enables the assessment of the severity level of a crisis in progress based on the heterogeneous real-time information. Fusion involves measurements from sensors, such as real-time weather observations, which are combined with local and dynamic information from citizens and first responders through incident reports sent from their mobile applications. The proposed Risk Assessment algorithm employs this information and estimates the risk/severity of the ongoing flood locally in the specific areas and/or globally in the whole region of interest. The generated outcomes are presented in various plots at beAWARE dashboard as well as at PSAP.

In the last development period, *Early Warning* component was updated to use the Weather Fire Index instead of the Simple Fire Index for the estimation of the expected fire danger. The predictions of this index and the overseen fire danger level are obtained from European Forest Fire Information System (EFFIS) portal. The results of the early warning are transmitted to the PSAP and beAWARE dashboard. Specifically, in the PSAP map, crisis and information managers receive indications regarding the estimations of the expected fire danger crisis in various pre-defined locations.

As concerns the *Real-Time Monitoring and Risk Assessment* component in the final period of development, the data from weather sensors fuse along with the outcomes of multimedia (image, video) data analytical modules and text analysis module of beAWARE. The goal is to dynamically assess the risk and severity level of the ongoing fire crisis by exploiting the obtained information from citizens and first responders' teams, that are nearby in the area where the fire crisis is in progress. Each time where a new incident with multimedia content is imported to the system and the analysis module produces the results, *Real-Time Monitoring and Risk Assessment* component receives the analysis and proceeds to the necessary updates of the severity level in the zone where the specific incident has taken place.

It is worth to note, that the Crisis Classification component is able to support various types of categorisation of the overall risk and crisis severity including the colour-coding. Thus, a 6 levels scale, which is similar to the EFFIS's categorization, has been implemented. However, in the fire pilot, Crisis Classification has adopted a 5 levels scale serving the end-users' needs and requirements.

With respect to the evaluation of the module's performance, the following indicators are used:

Performance Indicators	Number of forecasting and real-time observations
Definition	Number of forecasts, real-time observations that Crisis Classification components receive and handle during the pre-Emergency and Emergency phases.

Domain	Emergency Management Systems
Range	Real numbers
Limitations	Prediction models cannot produce any valid forecasts

Performance Indicators	Number of messages
Definition	Number of messages that generated as outcome of the performance of Crisis Classification
Domain	Computing
Range	Positive integer number
Limitations	n/a

Performance Indicators	Execution Time
Definition	Estimate the execution time in seconds over each one of the algorithmic steps of the Crisis Classification components.
Domain	Computing
Range	Positive real number
Limitations	n/a

2.3.6 **Text Analysis**

The text analysis component addresses T3.2 “Concept and conceptual extraction from multilingual text”. It enables the beAWARE platform to process textual inputs in the languages targeted in the project, English, Greek, Italian and Spanish, and produce an ontology-ready output that can be integrated into the semantic repository by the KBS.

For the third development period, UPF has re-designed the text analysis pipeline to produce an integrated linguistic structure from which to perform the extraction of concepts and relations. Obtaining this structure, described in detail in D3.4, involves reconciling overlapping annotations produced by the improved versions of the disambiguation and geolocation components, and marking all multiwords as a single unit (or token) before conducting the deep parsing of the input texts, so that nodes of the resulting dependency graph correspond to either individual words, locations or disambiguated meanings.

The concept and relation extraction component operates on this structure by simplifying the

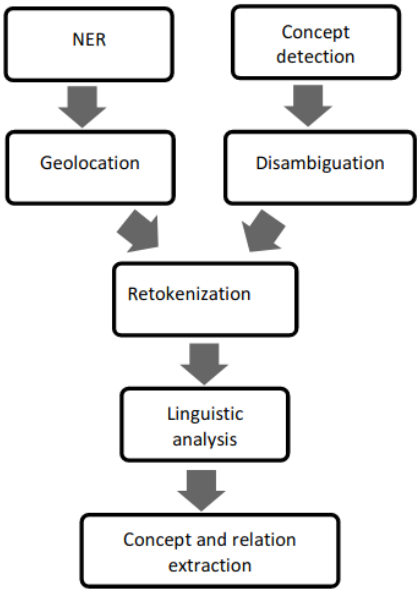


Figure 4: architecture of the final version of the text analysis module, as shown in D3.4

graph obtained from each sentence or tweet and mapping the meanings and locations to classes of the beAWARE ontology. The final version of the component has been improved to use mappings from BabelNet-based meanings to ontology classes obtained using semi-automatic methods. It has also been extended to detect and extract states related to incidents, e.g. hypothetical status of an event, its magnitude, etc.

The self-assessment plan, as described in deliverables D1.1, D1.2 and D1.3, foresaw two automatic quantitative evaluations of extracted concepts and of extracted relations, both against manually annotated corpora. A manual qualitative evaluation of the resulting conceptual representations was also planned, which should be conducted in terms of their completeness and expressiveness. In the deliverables reporting results for T3.2 -D3.3, D7.6 and D3.4-, quantitative evaluations were broken down into separate evaluations for each of the components making up the text analysis pipeline: syntactic dependency parsing and deep parsing (D3.3), and concept detection, disambiguation and geolocation (D7.6 and D3.4).

Introducing separate evaluations affected the baselines and performance indicators proposed in the self-assessment plan, which were replaced with baselines specific to each component.

The two qualitative evaluations in D7.6 and D3.4 focused on the structures integrating both extracted concepts and relations between them, and produced for the second and third pilots.

As explained in D3.3, the performance indicators used for the evaluation of the linguistic analysis tasks differed a bit from those proposed for the quantitative evaluation of relation extraction in D3.1. The table below describes the final version of the indicators.

Performance Indicator	Unlabelled Attachment Score (UAS) and Labelled Attachment Score (LAS)
Definition	Indicate the correctness and completeness of the extracted linguistic relations that are the basis for conceptual relation extraction. Unlike UAS, LAS considers the type of relation.
Domain	Surface and deep syntactic dependencies using UD or PTB tagsets.
Range	The values of these metrics are between 0 and 1.0.
Limitations	These metrics do not evaluate the final conceptual extractions directly, but the linguistic relations from which they are derived. They cannot asses the significance of errors for the final relation extraction task.

The following two tables describe the performance indicators used for the quantitative evaluations of the linguistic analysis, concept detection and disambiguation components. The indicators have been kept the same across deliverables detailing self-assessment plans and deliverables describing text analysis in WP3 –D3.3 and D3.4-, except that the precision and recall-based F1 scores are now reported instead of reporting separate precision and recall scores.

Performance Indicator	F1 of detected concepts
Definition	This metric compares the terminological concepts automatically detected by the concept detection component against a manually annotated gold-standard.
Domain	Concept mentions detected on textual inputs
Range	The values of this metric are between 0 and 1.0.

Limitations	This metric outlines errors in the delineation of concepts boundaries but cannot indicate the type and thus the severity of such errors. In addition, F1 cannot capture the implications of inter-annotator agreement (Cohen's kappa coefficient) in the attained upper bound performances.

Performance Indicator	F1 of disambiguated concepts
Definition	This metric compares the disambiguated references to BabelNet synsets produced by the entity linking component against a manually annotated gold-standard.
Domain	BabelNet synsets annotated on textual inputs
Range	The values of this metric are between 0 and 1.0.
Limitations	F1 indicates erroneous sense assignments but cannot assess the semantic distance between the assigned and expected sense.

The geolocation component was not foreseen in the DoW and therefore no performance indicator was specified in the various versions of the self-assessment plan. In this document we will report the same F1 metric used in D3.4:

Performance Indicator	F1 of detected locations
Definition	This metric compares locations detected by the system against a manually annotated gold-standard. Actual locations are compared -using the reference ids from geographical databases- rather than just annotations of mentions in the text.
Domain	Geolocated mentions of locations in texts
Range	The values of thi metric are between 0 and 1.0.
Limitations	This metric does not account for geographical distance between locations, nor does it account for inter-annotator agreement.

The self-assessment plan set improvements of 5% and 15% over best-performing baseline as lowest and highest expectations for each of the performance indicators described in the tables above.

2.3.7 Automatic Speech Recognition

The Automatic Speech Recognition (ASR) component is used in combination with Multilingual Text Analyser (MTA) in order to automatically extract information from emergency calls and audio messages. Until the second prototype, the Italian and Greek acoustic models had been adapted to case-specific recorded speech, in order to enhance emergency-related terminology and corresponding dictionaries had been cleared from erroneous or rare words. A call-center solution was also integrated in the platform, in order to receive emergency phone calls, and a relevant function was developed, able to fetch recorded calls and forward them to ASR. During the call, the caller is able to determine his/her language, through an Interactive Voice Response (IVR), in order for the call to be forwarded to the corresponding ASR language model.

As it has already been described in D3.4, at the final version of the component, the focus was mainly on the Spanish model, which was also the official language of the third pilot. The Spanish model was enhanced by adding missing words and locations in the Spanish dictionary from a set of phrases created by PLV. The Spanish language model (LM) was adapted accordingly, by extending the initial LM with new word sequence probabilities from the generated dataset. Additionally, some technical issues affecting recognition accuracy were fixed, including the format of the language models and the quality of the audio files coming from the Mobile App. Finally, for the needs of the Fire Pilot (Blended Phase), in collaboration with PLV, their dedicated call center was integrated to beAWARE and emergency calls were fetched and transferred to ASR component.

With respect to the process of evaluating the performance of ASR, the following performance indicators are used, which were also described in D1.1:

Performance Indicator	Word error rate (WER)
Definition	WER is a common metric for measuring the performance of a speech recognition system, by comparing the reference transcription (ground truth) and the ASR output (hypothesis of what was said). It includes: substitution errors (S), i.e. miss-recognition of one word for another, deletion errors (D), i.e. words are missed completely, and insertions (I), i.e. extra words introduced into the text output by the recognition system. WER is defined as: WER=(S+D+I)/N, where N is the number of words in the reference. It is usually expressed as percent word error %WER, which is WER*100%.
Domain	Speech recognition

Range	The values of this metric are larger than 0, having no upper bound.
Limitations	Since the WER metric doesn't have an upper bound, it doesn't measure how good a system is, but only shows that one is better than another. Additionally, at high error rates the measure gives far more weight to insertions than to deletions.

Performance Indicator	Word accuracy (WAcc)
Definition	WAcc is another metric commonly used for measuring the performance of speech recognition systems and is computed as WAcc = 1-WER. It is usually expressed as percent word accuracy, which is defined as %WAcc = 100 - %WER.
Domain	Speech recognition
Range	The upper bound for the values of this metric is 1, with no lower bound.
Limitations	WER can be larger than 1 and as a result, WAcc can be smaller than 0.

2.3.8 **Visual analysis**

Visual analysis in the beAWARE project is carried out by the IMAGE ANALYSIS and VIDEO ANALYSIS components and their overall objective is concept extraction from visual content (images/videos). Several modules have been developed and integrated:

- Emergency classification, so as to determine which images/videos contain an emergent event or not (i.e. a fire, smoke or flood event). This module participated in the 3rd pilot.
- Object Detection and Tracking, so as to find people, animals and vehicles that exist in impacted locations.
- Face Detection, so as to accurately count persons inside shelters and places of relief.
- Dynamic texture localization, so as to localize fire or flood dynamic textures in images/videos and estimate the severity level of the detected people and vehicles in the same area.
- Visual River Sensing performs visual analysis on videos from static surveillance cameras installed by the river, in order to estimate the water level and generate alerts, in case of threshold exceeding. This module has already been demonstrated in the Flood pilot and evaluation results have been presented in D7.6. The module was not used during

the Fire pilot and consequently it will not be evaluated in this deliverable. However, due to several improvements since the second prototype, its performance has been evaluated again and results were presented at D3.4.

- Sensitive content blurring, so as to protect the privacy of targets inside the visualized images/videos on the platform if needed.

The following tables define the performance indicators that will be used in this report for the visual analysis components:

Performance Indicator	Classification Accuracy
Domain	Image Classification
Definition	Classification accuracy is an adjusting percentage score that indicates the percentage of correct predictions. In other words, it is the ratio of True Positives and True Negatives over all samples.
Range	The values of this metric are between 0 and 1.0. Higher is better.
Requirements	To perform this evaluation, annotated data must exist or be prepared.

Performance Indicator	Average Processing Time (seconds)
Domain	Image and Video Processing
Definition	It is the average time in seconds that the components need to process a single item (image or video)
Range	All positive numbers. Higher means faster.
Requirements	None

2.3.9 **beAWARE Knowledge Base**

The Knowledge Base (KB) constitutes the core means for semantically representing the pertinent knowledge and for supporting decision-making. The Knowledge Base Service (KBS) receives notifications from the other beAWARE modules (e.g. the analysis components) and populates the KB with newly available data. By applying reasoning rules, the overall situation is assed and decision-making is supported. The semantic content in the KB is based on the beAWARE ontology, which represents the data in a well-defined formalism. The Knowledge Base also provides a user interface (see Figure 5) for (i) accessing the risk maps, (ii) analysing the available messages (see D4.3 for more information about the analysis workbench), (iii)

incident map and (iv) incident list to visualize and navigate through the available semantic content.

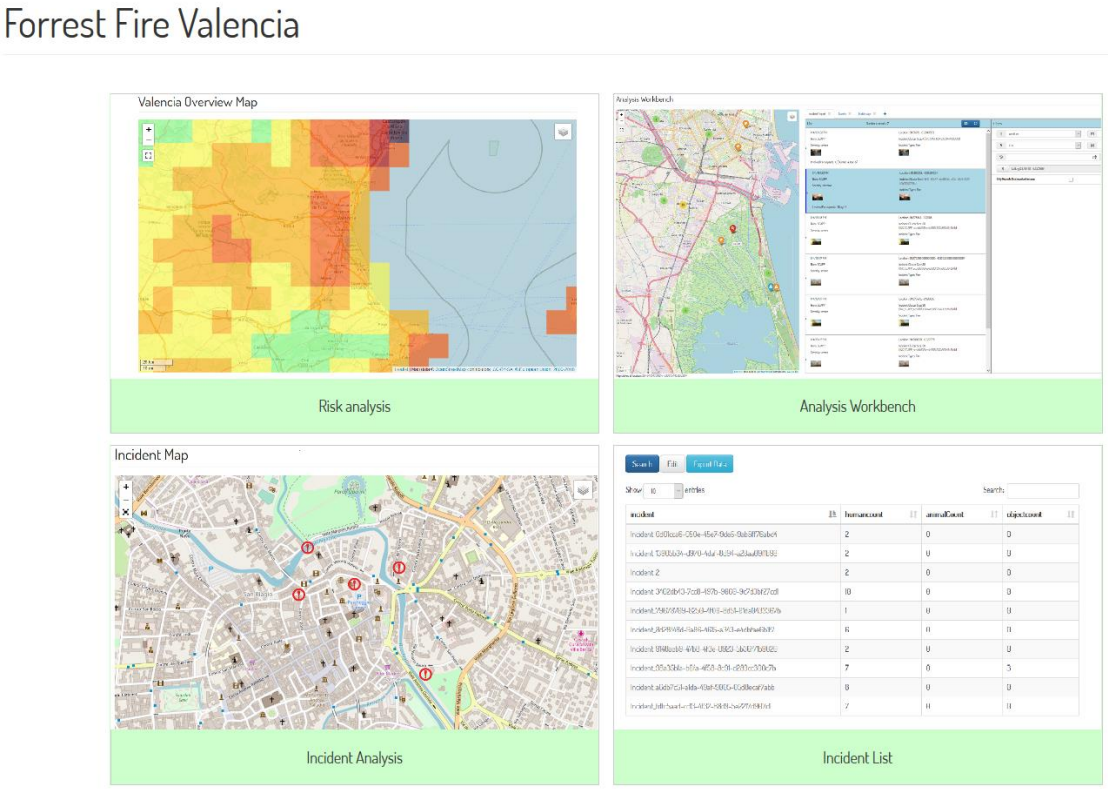


Figure 5: User interface for the Knowledge Base

Both KB and its service (KBS) continuously change in response to the maturation of the system. This happens, on one hand, due to the enrichment of the ontology in order to take into account new concepts relevant to the beAWARE UCs and on the other hand due to the insertion of new features and components used to extract further and more accurate information. Like explained in D4.3 and D7.8 new concepts (e.g. animals, people in wheelchairs, …) have been included to represent the new capabilities of the analysis components.

A quantitative evaluation of the ontology is not possible. Therefore, we refer to well-known metrics and tools, which allow a qualitative evaluation of the ontology. Therefore, with respect to the evaluation of the module's performance, the following indicators are used:

Performance Indicator	Ontology consistency
Definition	Assess whether an ontology model is syntactically and semantically consistent. Typically performed with the help of a reasoner (e.g. Pellet, HermiT).
Domain	Parse model and check for inconsistencies.

Range	Only 1 of 2 values returned: (1) True (consistency checks succeed) OR (2) False (consistency checks fail). Some reasoners also provide explanations in case of failure.
Limitations	• For very complex models, consistency checking and explanations generation is time- and resource-consuming. • Explanations may be too complex to follow.

Performance Indicator	Ontology quality
Definition	Diagnose and repair potential pitfalls in the modelling approach that could lead to modelling errors. Can be performed with the help of relevant software tools (e.g. OOPS! – OntOlogy Pitfall Scanner!).
Domain	Parse model and check for modelling pitfalls.
Range	Three types of pitfalls: critical, important, minor. Possible negative consequences may also be calculated.
Limitations	Relying on third-party services entails risk in case the services are discontinued in the future.

Performance Indicator	Ontology structure
Definition	Assess the quality of the ontology's structure with regards to attribute richness, width, depth and inheritance. Relies on graph-based and schema evaluation metrics. Can be performed with the help of relevant software tools (e.g. OntoMetrics).
Domain	Parse model and generate values for the metrics.
Range	$R_{\geq 0} = \{ x \in R \mid x \geq 0 \}$
Limitations	Relying on third-party services entails risk in case the services are discontinued in the future.

beAWARE Knowledge Base Service

The interaction between the beAWARE Knowledge Base and the Knowledge Base Service (KBS) is based on the execution of complex and elaborate queries from the latter to the first.

With respect to the evaluation of the module's performance, the following indicators are used:

Performance Indicator	Semantic fusion execution time
Definition	Assess the execution duration of processes that populate incoming knowledge to the ontology (semantic fusion) in relation with the volume of data already existing in the ontology. This should reveal any underlying scalability weaknesses of either the KB or the KBS when the stream of data during a crisis dilates.
Domain	Run a simulation of the Valencia pilot to generate values for the metrics.

Range	Positive real numbers for time values where lower is better.
Limitations	Execution times are expected to vary, based on the provided computing resources of the deployment environment. Additionally, the network communication overhead affects the overall performance. For our evaluation, WG was deployed on the cloud servers, and the KBS was deployed on CERTH's premises on a Virtual Machine with 5GB of RAM, 4-core CPU and an SSD.

Performance Indicator	Semantic reasoning execution time
Definition	Evaluate the execution duration of semantic reasoning mechanisms. In a nutshell, the latter undertake the interlinkage of discovered knowledge and the investigation for new/underlying knowledge in the ontology. These tasks are expected to present an increase of execution times proportionate to the volume of data already in the ontology.
Domain	Run a simulation of the Valencia pilot to generate values for the metrics.
Range	Positive real numbers for time values where lower is better.
Limitations	Execution times are expected to vary, based on the provided computing resources of the deployment environment. Additionally, the network communication overhead affects the overall performance. For our evaluation, WG was deployed on the cloud servers, and the KBS was deployed on CERTH's premises on a Virtual Machine with 5GB of RAM, 4-core CPU and an SSD.

Performance Indicator	Kafka Bus message handling times
Definition	KBS input arrives via the Kafka bus in the form of various message types (topics). Each topic requires different actions, i.e. a dedicated sequence of queries towards the WG. These actions apparently present a variable complexity, thus a study on the temporal performance per message type is of special interest.
Domain	Run a simulation of the Valencia pilot to generate values for the metrics.
Range	Positive real numbers for time values where lower is better.
Limitations	Execution times are expected to vary, based on the provided computing resources of the deployment environment. Additionally, the network communication overhead affects the overall performance. For our evaluation, WG was deployed on the cloud servers, and the KBS was deployed on CERTH's premises on a Virtual Machine with 5GB of RAM, 4-core CPU and an SSD.

Performance Indicator	KBS messages validation
Definition	The validation component reads the output of the KBS and processes it in order to detect potentially erroneous incidents. This process includes parsing the Kafka bus messages and exchanging messages with the Crisis Classification component to crosscheck it with environmental metrics. The average duration for a message to be validated illustrates the impact of this new component to the system.
Domain	Run a simulation of the Valencia pilot to generate values for the metrics.
Range	Positive real numbers for time values where lower is better.
Limitations	-

The performance indicators demonstrated in this section have the execution duration values as a common factor. Consequently, a set of timers has been injected in the code of the KBS to calculate and log all required times. The generated datasets also contain associations with the volume of stored incident reports at that moment, as a metric of scalability from user-generated incoming data.

beAWARE geoServer

Risk maps are used to articulate and visualize risks at the asset level. Next to those risk maps, additional layers with use case specific information have been integrated (see Figure 6). The 2nd version of the beAWARE platform has been extended to support the 3rd pilot in the area of Valencia. Therefore, external data sources (e.g. locations of hydrants in the area) and information about past events (burned areas in in the years before) have been integrated. Those maps ca be displayed in the KB UI and can be accessed by other modules (in this case the crisis classification component) via a standardized interface (Web Map Service; WMS).

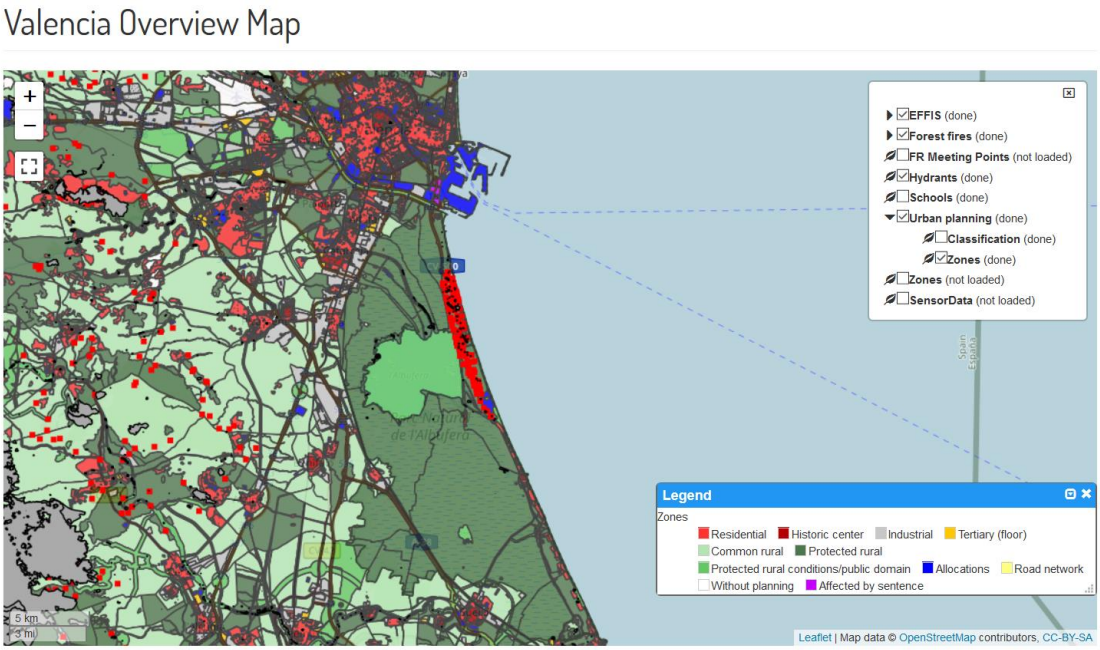


Figure 6: Visualization of the available GIS data

A dedicated technical evaluation was not performed for the risk maps. They are integrated in the overall beAWARE platform and part of the 3rd pilot. Therefore, the evaluation is done in the Evaluation report of the final system in D2.8.

2.3.10 **Multilingual Report Generator**

Starting from contents in the knowledge base, the report generation module produces multilingual text providing to the users of the platform with relevant information about an emergency. Two types of reports have been implemented, short situational updates typically 1 or 2-sentence long, and wrap-up summary reports issued at the end of an emergency and containing multiple multi-sentence paragraphs.

Work for the final release has largely focused on updating the module to new ontology contents and in improving the quality of the wrap-up summaries. This has involved improving the methods for mapping ontological representations onto linguistic structures, and on improving the methods for hybrid rule-based and statistical multilingual text generation. As an important by-product of the work in beAWARE, multilingual datasets have been developed for training the models and resources used for text generation.

As explained in D5.3, the evaluation strategies and indicators used in WP5 deliverables evaluate multilingual generation rather than text planning, as the latter was addressed with simple ad hoc methods due to user requirements. For this reason, the indicators proposed for text planning in the self-assessment plan (see D1.1 and D3.1) have been dropped in favour of a more thorough evaluation of linguistic generation. Evaluation strategies carried out include automatic qualitative and manual qualitative evaluations for multiple languages. The manual

quantitative evaluations in the self-assessment plan using the questionnaires introduce in D7.6 have been finally excluded from this deliverable. This decision, already introduced in D5.3, has been adopted due to the nature of the 3rd pilot, where all reports produced by the system were already tailored to the specific emergency scenario and would have produced artificially inflated results if evaluated using questionnaires.

Below are the tables describing the performance indicators used for the evaluation of the multilingual report generation module. BLEU was already proposed in D1.1 and reported in the technical report for the second pilot D7.6, while METEOR and TER were introduced in D5.3.

Performance Indicator	BLEU
Definition	Precision-oriented N-gram-based comparison of sentences in system generated text against gold text.
Domain	Texts in each of the beAWARE languages.
Range	From 0 to 1.0.
Limitations	Based on strict word matching, cannot account for synonyms or semantically-related words. Favours shorter system texts.

Performance Indicator	METEOR
Definition	Recall-oriented unigram comparison of sentences in system generated text against gold text.
Domain	Texts in each of the beAWARE languages.
Range	From 0 to 1.0
Limitations	Based on stemming and synonyms, the correlation of the metric with human judgements depends on the quality of language-specific stemming tools and synonymy dictionaries.

Performance Indicator	TER
Definition	Comparison of sentences based on minimum number of edits -insert, delete, replace and shift single words- required to transform system sentence to gold sentence.
Domain	Texts in each of the beAWARE languages.

Range	From 0 to 1.0
Limitations	Based on strict word matching, cannot account for synonyms or semantically-related words.

2.3.11 **Drones Platform**

The drones platform is a service to connect providers of drones, drones' services, and customers, to easily configure, launch, and monitor drone related activities. The drones platform consists of 3 components: 1) the Drones server, 2) the Drones edge device, 3) the Platform Dashboard.

The essence of the drones platform capabilities is the combination of route planning and drones agnostic autonomous dynamic piloting, with the provisioning of data and metadata collected by the drone, making it available to interested beAWARE analysis components.

In the second and final iteration of the Drones Platform, based on the fire use case requirements, we concentrated on supporting the transmission of video from the drone to back-end, supporting additional analysis components to consume that data. Work included controlling the bit rate, employ compression mechanisms, based on available bandwidth and capacity of the corresponding drones video analysis component.

The following tables provide the definition and description of the main properties of each of the pertinent performance indicators.

Performance Indicator	Dynamic route planning
Definition	Ability to define parts of the flight plan dynamically in real-time while in the middle of a flight
Domain	Flexibility
Range	Binary (0 or 1)
Limitations	Limited by the battery life for a single flight

Performance Indicator	Bi-directional interaction with the platform
Definition	Ability to send imagery at an appropriate rate and consume back analysis results sent by the platform
Domain	Performance
Range	Positive numbers – the higher the better
Limitations	Limited by the performance of the network connectivity

2.3.12 **Public Safety Answering Point**

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.

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 final version, we have extended the information displayed on the map having the ability to see more details in “drill down” mode for a specific event, the set of map icons extended to differentiate per incident type and also for new were added for metrics display.

In addition, we have reworked the color coding together with PLV team, added the ability to present the zone of interest with rectangular boundaries, improved the clarity of alerting mechanism by displaying the radius of the population being alerted on the map and improved the command and control picture.

In the Operations Manager module, we have added the ability to modify an existing task and in the PA an ability to re-send a previous message with different parameters

The following tables provide the definition and description of the main properties of each of the pertinent performance indicators.

Performance Indicators	Visualisation time
Definition	Visualisation time is the time needed by our interface to display the data received. Specifically, for the PSAP component, visualisation time refers to the number of seconds between an incident or metric report is received until the time the data is visualised on the Map or the Dashboard.
Domain	Computing
Range	The values of this metric are larger than 0.0, having no upper bound.
Limitations	-

2.3.13 **Mobile Application**

The mobile application is the interface used by citizens and first responders to interact with the beAWARE platform.

In the first prototype, it was possible to send multimodal reports and receive public alerts. For the second prototype, the app was extended with basic team- and task management functionality. In the final version the team functionality was extended to be able to specify a team-name and -profession, which can be now used to send public alerts to a specific group of first responders. Furthermore, the user interface and user experience were improved to adapt commonly used patterns in mobile applications.

With respect to the evaluation of this module, the following indicator are used:

Performance Indicator	Number of met requirements
Definition	Number of the user requirements (listed in D2.10) that are realized in the mobile app.
Domain	Requirements
Range	Number of requirements defined in D2.10
Limitations	

Performance Indicator	Usability
Definition	Clear and user-friendly visualization of different information layers gathered from disparate data sources
Domain	Visualization and interaction
Range	5-point Likert scale.
Limitations	Each report should be assessed by multiple UI elements

3 Technical Evaluation

In this section, an evaluation report is provided. The evaluation performed is in accordance with the criteria and methodology spelled out in the previous section and carried out by the performance indicators defined in the first part.

3.1 Social Media Monitoring

The Social Media Analysis (SMA) module has not been modified since the second prototype, so the evaluation stands the same as described in deliverable D7.6 (M26). In short, the adapted text classification technique (to estimate the relevancy of a tweet) has been evaluated on a dataset of 1,000 human-annotated tweets in Italian about flood, achieving a precision of 84%, a recall of 89% and an F-score of 87%. Furthermore, it has been examined whether the validation layer improves the results. Indeed precision and F-score have been raised to 96% and 93% respectively.

On the other hand, the complete version of Social Media Clustering (SMC) has been integrated and evaluated after the second prototype. The experiments concerned determining which clustering technique is the most suitable for the spatial grouping of tweets in the frame of disaster incidents. The dataset consisted of 88 synthetic Spanish tweets about fires, which have been created by PLV specifically for the 3rd pilot of beAWARE in Valencia, Spain. A comparison was realized between 16 clustering methods, using Normalized Mutual Information (NMI) score as the evaluation metric. The results in Figure 7 show that our fine-tuned DBSCAN implementation (*eps* set to 0.001 and *minPts* set to 3) outperformed the other algorithms, managing to predict the correct number of clusters and achieve an NMI score of 1.0.

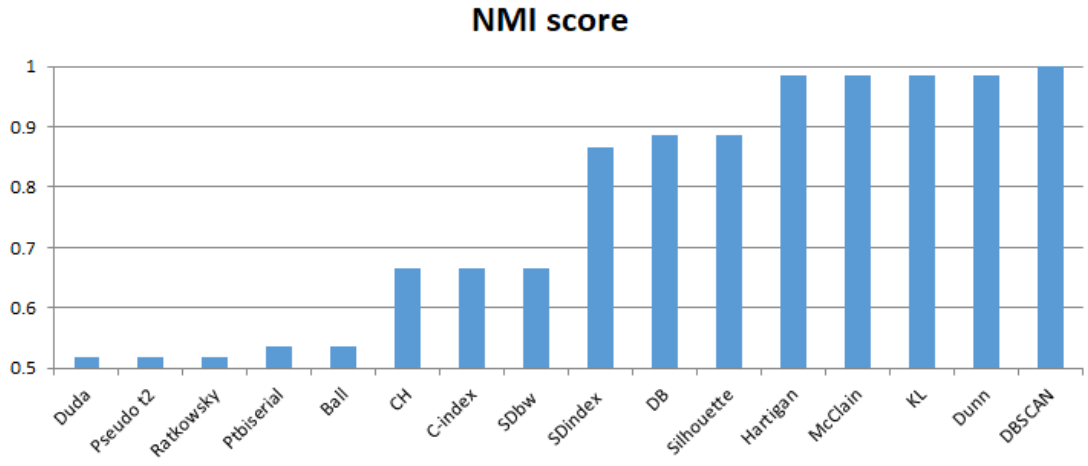


Figure 7. Comparison of clustering techniques, with fine-tuned DBSCAN outperforming

As far as it concerns the third beAWARE pilot, 45 tweets have been crawled in total, out of which 3 were found fake, 3 contained unrelated emoticons, 5 were estimated as irrelevant to fire incidents and 34 were estimated as real and relevant. Checking on the content of these tweets, the above classifications were correct. Based on these 34 validated tweets, 17 Twitter reports were created and displayed as incidents. In general, both SMA and SMC and the involved services (e.g., verification, relevancy estimation) have all worked as expected and no issues have been raised during the pilot.

3.2 Communication Bus

The main purpose of this component is to provide generic communication capabilities among different beAWARE components. It is used to send messages and notifications among components. In a microservices based architecture, such as beAWARE has adopted, there is a need for communication among different microservices, and the communication bus fills this requirement as a means for components to declare the availability of a new piece of information, combined with components their interest to be notified. Extensive work has been done in beAWARE to reach an agreed upon list of topics and their corresponding formats.

The communication bus is configured, upon deployment, with the necessary set of topics as agreed upon between the different components. In addition, the message structure of each message in each topic is agreed upon and documented by the cooperating components. The communication bus supports the number of different topics required for a beAWARE installation, along with the associated aggregated throughput in all topics. That assertion was validated in the 3 project pilots and in the continues testing of the platform. Moreover, in the third pilot we enhanced the drones platform to support a continuous flow of video chunks thus exercising both the object store and the message bus heavily, by sending video chunks every 3 second over a period of approximately 15 minutes per flight session. BeAWARE experienced no problems coping with the required throughput exhibiting a reasonable latency. A representative session included sending 296 video files, corresponding to about 15 minutes of video. The duration of message submission to the message bus was: 228 ms on average, with a standard deviation of 33 ms.

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

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

Some representative figures of the load on the message bus while simulating the third pilot workloads.

Object storage statistics are provided as well in Figure 10, indicating more than 36K files stored with a total size of 16 GB.

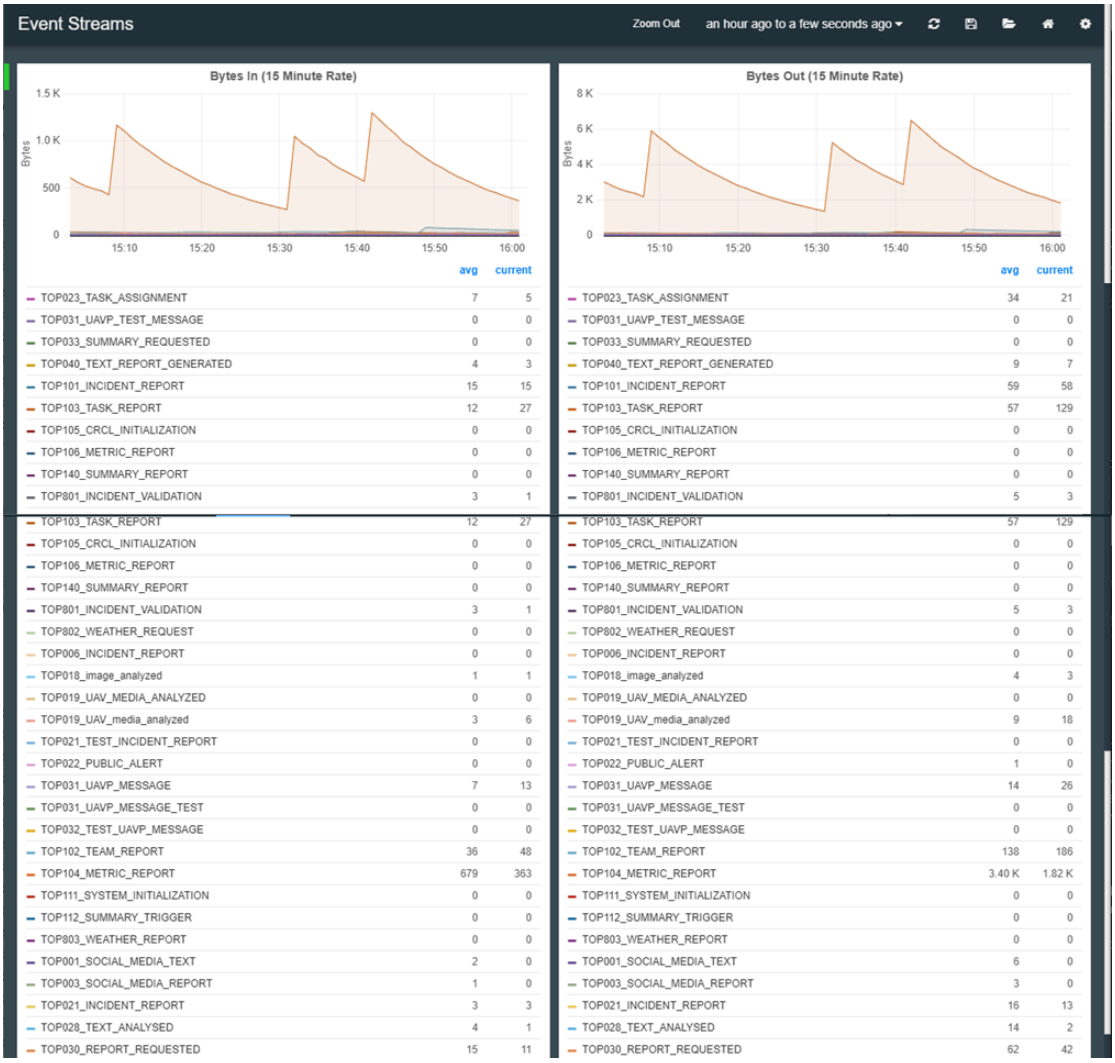


Figure 8: Message bus statistics

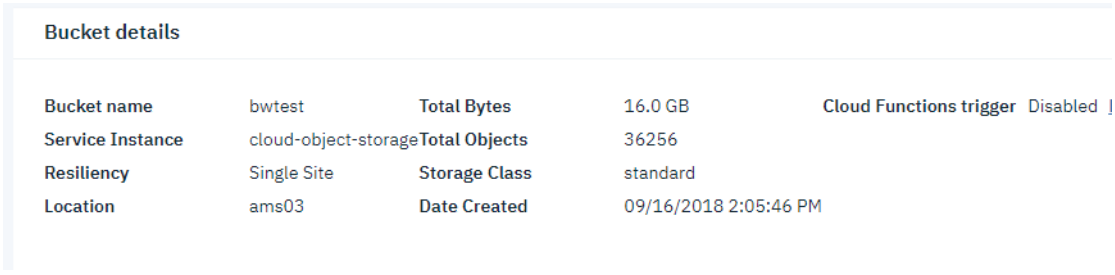
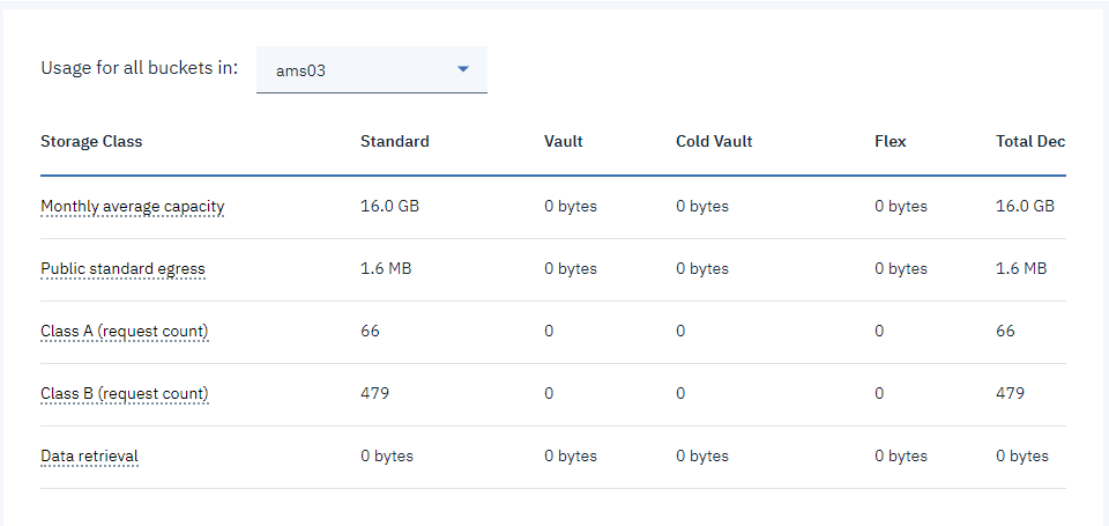


Figure 9: Cloud Object Storage details



Usage for all buckets in: ams03

Storage Class	Standard	Vault	Cold Vault	Flex	Total Dec
Monthly average capacity	16.0 GB	0 bytes	0 bytes	0 bytes	16.0 GB
Public standard egress	1.6 MB	0 bytes	0 bytes	0 bytes	1.6 MB
Class A (request count)	66	0	0	0	66
Class B (request count)	479	0	0	0	479
Data retrieval	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes

Figure 10: Object Store statistics

Scalability and performance measures

There are many scalability dimensions in the communication bus. The deployed system comfortably accommodates the load of the beAWARE pilots, and has the capacity to support a higher load, given the current installation and deployment. In addition, there are various scalability factors, affecting performance, that can be applied when the system load gets considerably larger.

1. Number of servers / brokers - For scalability and fault tolerance the communication bus can run with several servers acting as cooperating message brokers. Currently beAWARE's communication bus is deployed over 5 brokers. The number of brokers can be scaled up based on need, but for the foreseeable future there is no expectation that the platform would require more brokers to be deployed. Replication factor for beAWARE's topic is 3, thus we ensure that sent messages are available in at least 3 brokers, such that the platform can continue normal operations even in the unlikely event of two brokers being unavailable simultaneously.
2. Number of topics - Each topic forms a separate unit to which messages can be sent and through which messages can be consumed. In such a manner the entire spectrum can be divided between different processes distributed over different nodes and have the overall load to be distributed between different clients and different broker entities. Currently in the communication bus there are 42 topics declared and used operationally (up from 28 used at the flood pilot).
3. Number of partitions - The partition is the unit of total order within the communication bus. Every topic is divided into 1 or more partitions. The number of partitions of a topic can be scaled up and down based on need. BeAWARE uses a single partition per topic.

During the fire pilot the heaviest user of the message bus was the drone platform. In every session (flight) the drone sent one message per 3 seconds (exercising heavily also the object store which received an upload request from the drone every 3 seconds, and a corresponding download from the video analysis component every 3 seconds). Total amount of messages per session amounted to 296.

Currently there are 42 topics defined in the beAWARE message bus, as can be seen in Figure 11.

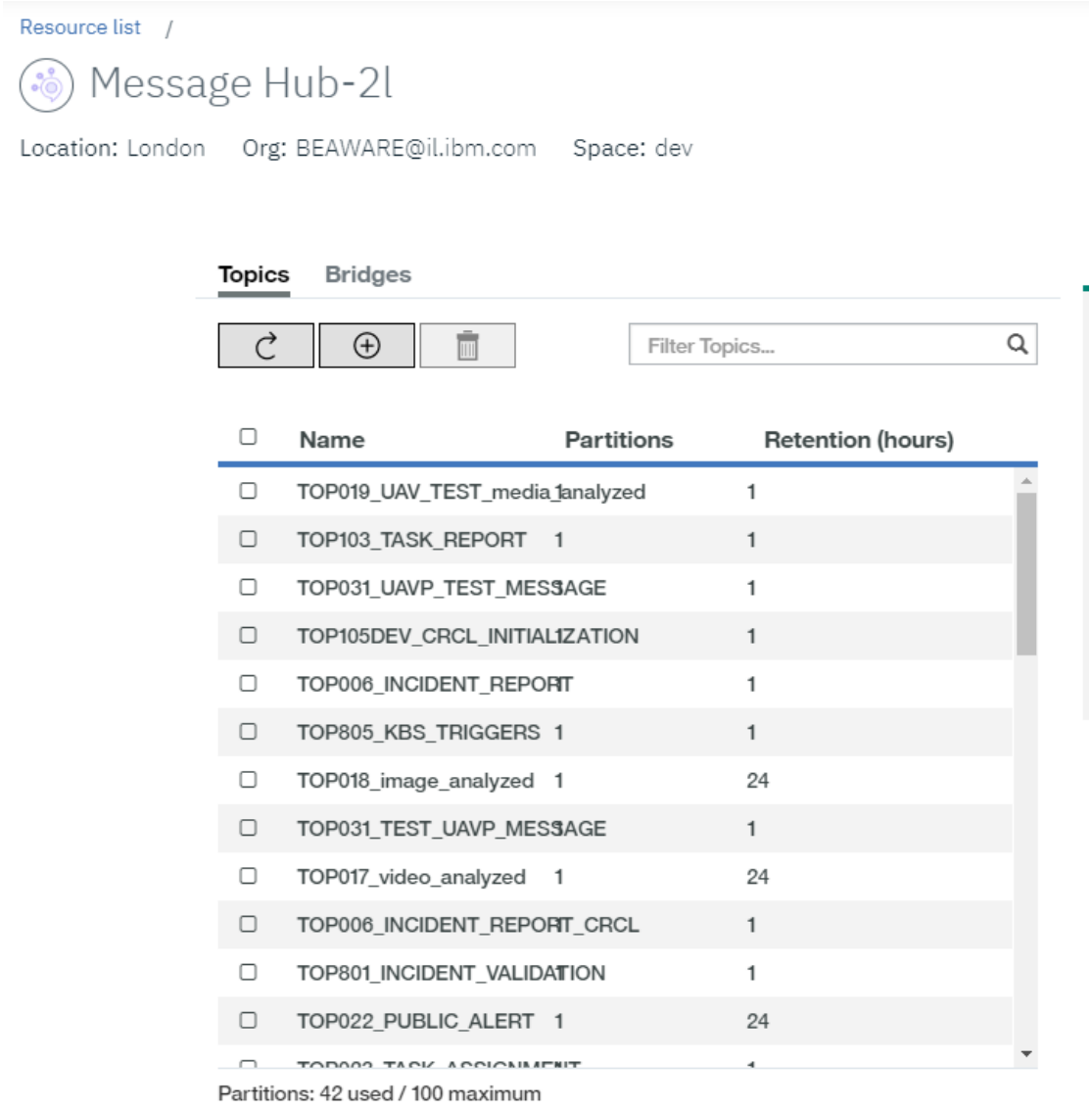


Figure 11: Message Bus topics

The beAWARE message bus has been operation for over 2 years and no issues were reported on its availability, scale, and performance. Some performance numbers observed during one of the latest tests of the system can be seen in Figure 11.

3.3 Technical Infrastructure

The technical infrastructure relies on 3 major components, namely, a git hub repository (Figure 13) for source control, which is hooked to a Jenkins instance (Figure 13) for CI / CD, and finally a target deployment infrastructure in the form of a Kubernetes cluster running on the IBM cloud. The IBM cloud infrastructure provides also the necessary middleware required for the operation of the system in the form of storage and messaging services. A glimpse through the cloud dashboard can be seen in Figure 14.

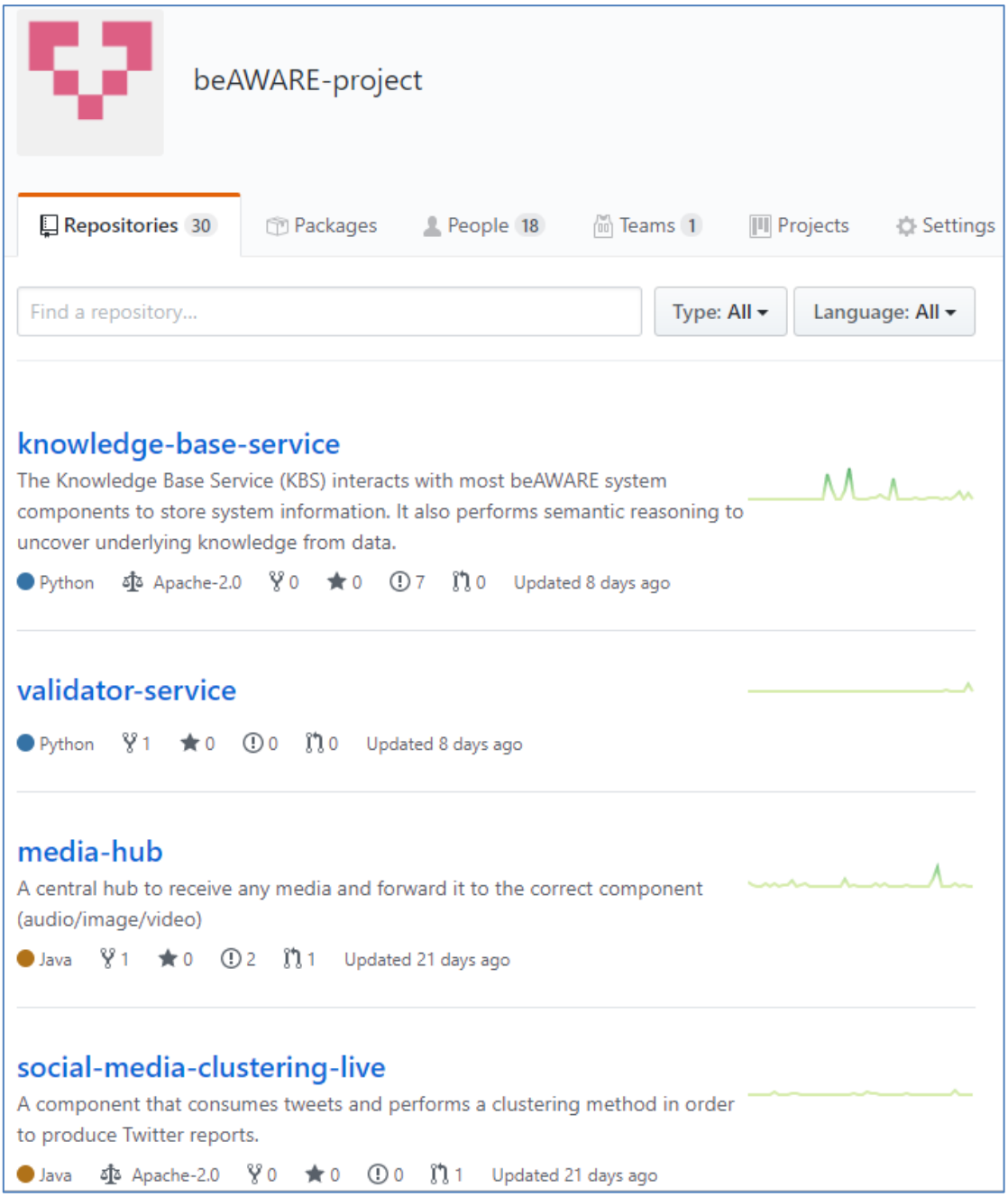


Figure 12: Git Hub repository

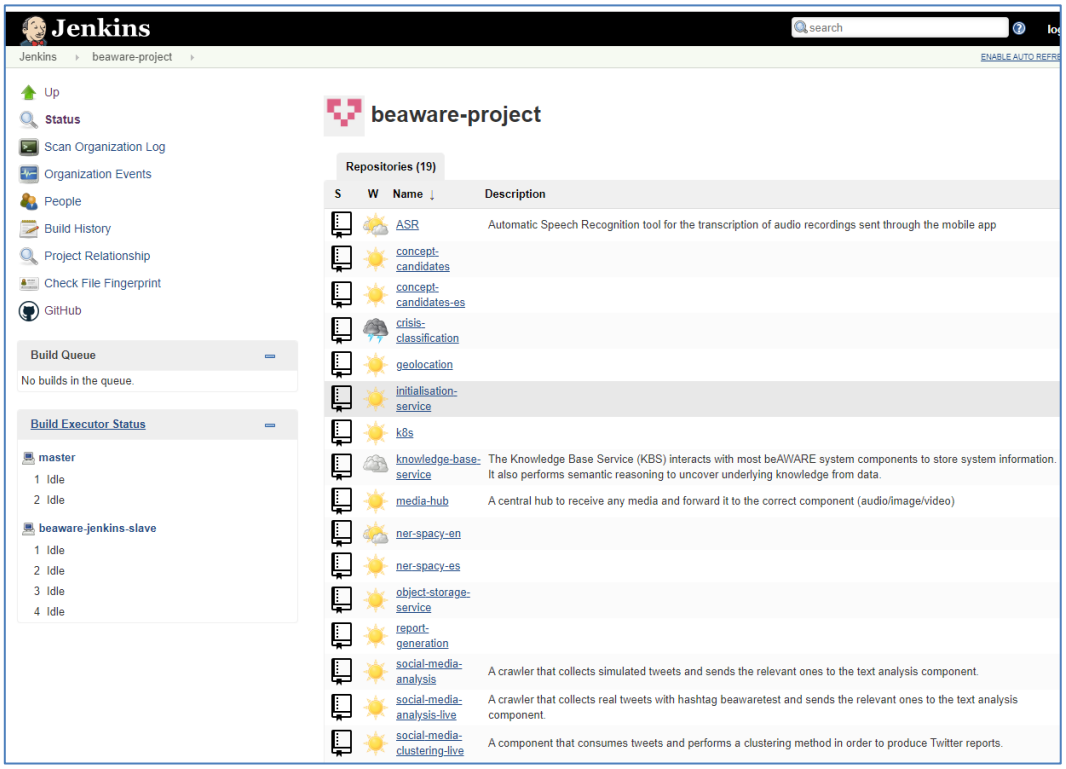


Figure 13: beAWARE Jenkins

The supporting data stores consist of the object store, which is used to share files between different components (for example an image that needs to be analyzed). During the fire pilot the heaviest user of the object store was the drone platform. In every session (flight) the drone uploaded a video file every 3 seconds, and a corresponding download from the drones analysis component every 3 seconds). Total amount of files per session amounted to 612, for a total size of 78.4 MB.

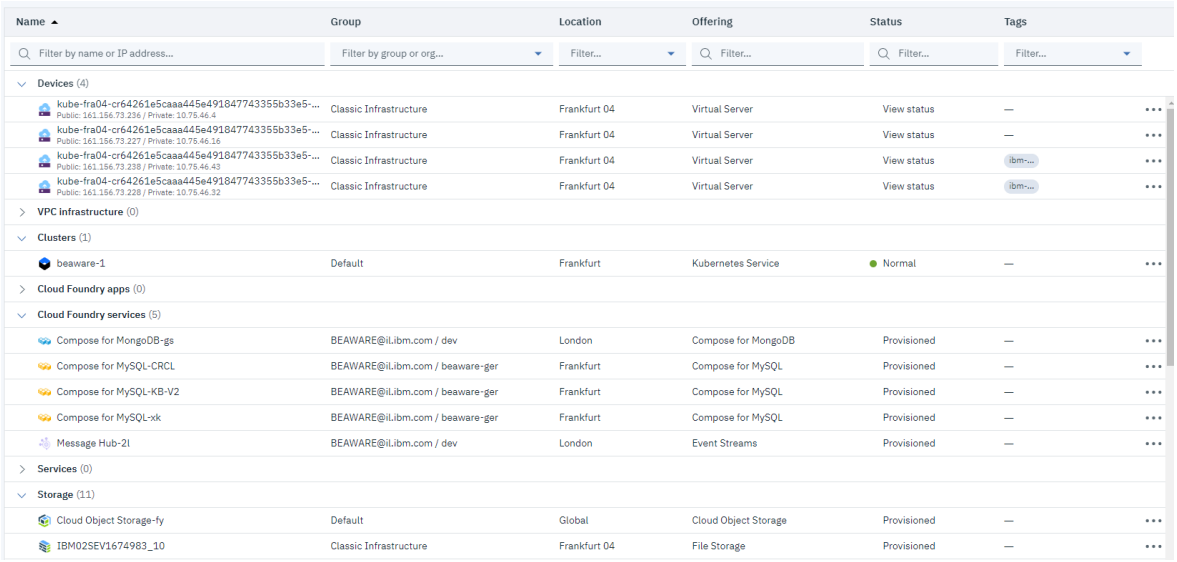


Figure 14: beAWARE cloud infrastructure

An instance of Mongo DB can be seen in Figure 15, is used mainly by the social media component.

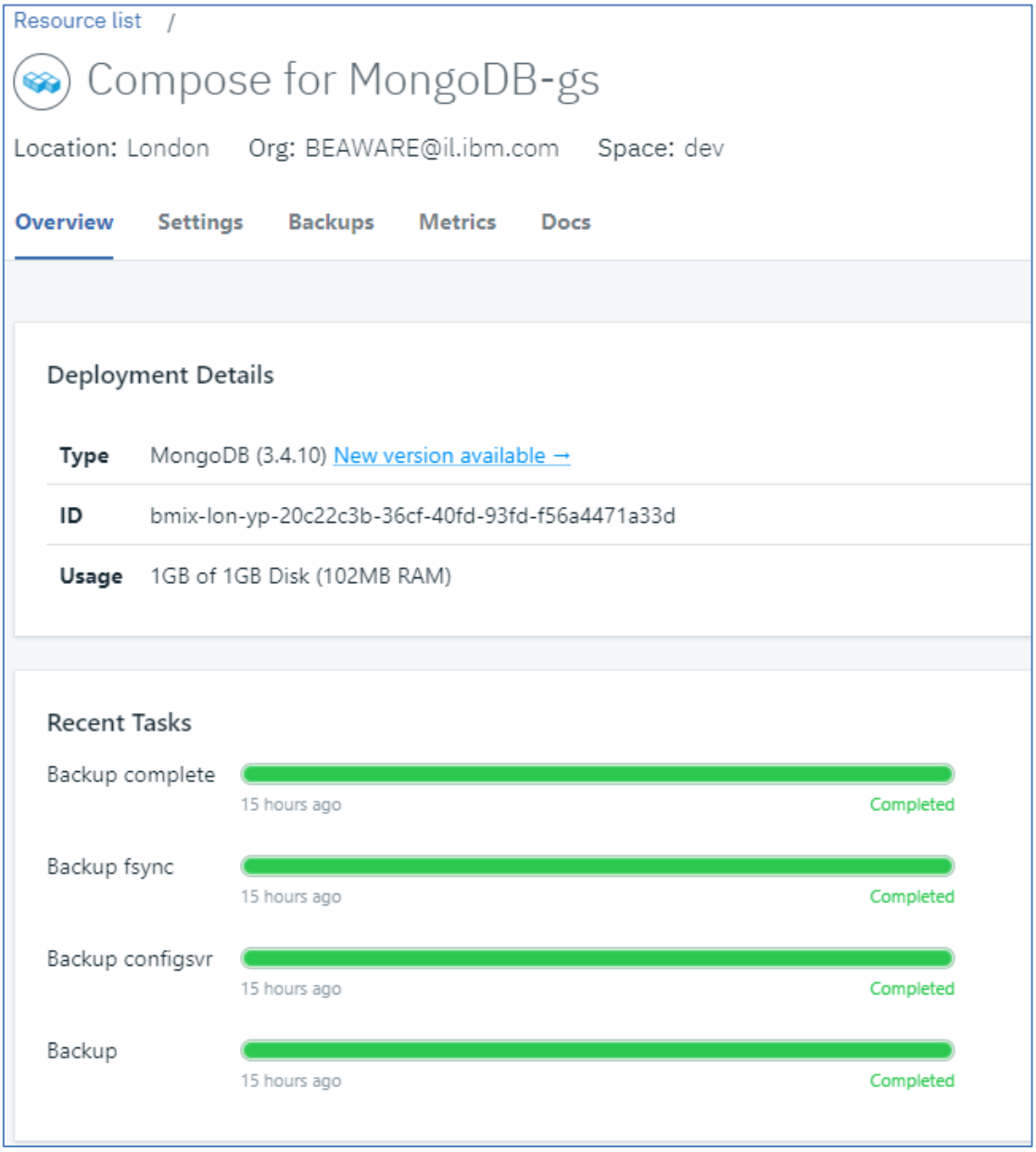


Figure 15: MongoDB instance

Finally, 3 instances of MySQL are deployed ad used by the KB and the crisis classification modules (Figure 16)

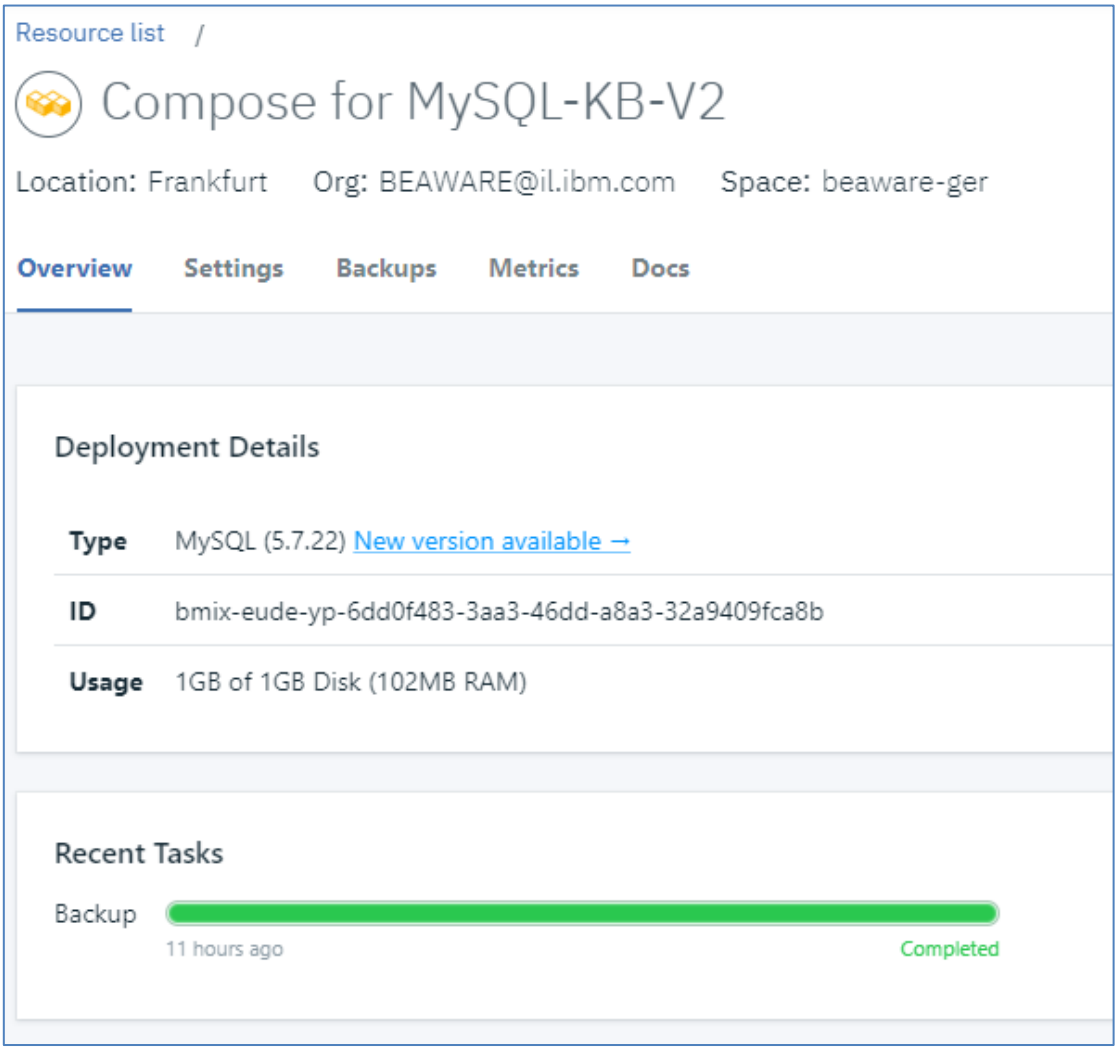


Figure 16: MySQL instance

3.4 Crisis classification

The goal of this section is to exhibit the evaluation results of the final version of the Crisis Classification component in terms of the performance indicators. The evaluation process relies on the performance of the *Early Warning* and the *Real-Time Monitoring and Risk Assessment* components in terms of the amount of data (forecasts, real-time observations) that they can handle, the execution time as well as the accuracy of the analysis results.

The core functionalities and approaches of these components have not been modified since the second prototype, however some functionalities have been adjusted or enriched, especially in the Real-Time Monitoring and Risk Assessment component, so as to meet the

specific requirements and needs of the fire pilot. In the following subsections these modifications will be evaluated under operational conditions during the fire pilot.

3.4.1 Evaluation of the Early Warning component

Briefly, the *Early Warning* component includes the following steps to accomplish the goals of the fire pilot:

Step 1. Data Acquisition from FMI: The weather provisions for various parameters (air temperature, humidity, wind speed/direction, precipitation) in specific locations in the Valencia region are obtained by requests to the FMI OpenData API. In this step, the appropriate messages are generated so as to proceed for presentation in the beAWARE dashboard.

Step 2. Data Acquisition from EFFIS: The predictions for Fire Weather Index and fire danger are obtained with ftp process from EFFIS portal. The obtained files with the data are in the netCDF format.

Step 3. Data Analysis: for each forecasting day out of 9 days period, the analysis includes the following steps:

a. estimate the Fire Weather Index level over the specific locations (points) in the Valencia region by interpolating the FWI values of grid points in the obtained netCDF file.

b. create the appropriate messages including the analysis results of the overall Fire Weather Index per location and forecasting day and forward them to beAWARE dashboard

Step 4. Aggregated Analysis: for each location the estimation of the 1st time that Fire Danger exceeds the 3rd alarm threshold and its maximum value in the forecasting period of 9 days are carried out. The results are forwarded to the PSAP/map.

Step 5. Aggregated Analysis: the mean value and the standard deviation of the Fire Weather Index over the forecasting period for each location are calculated. The outcomes are forwarded to beAWARE dashboard in order to create the error-bar plots.

In order to estimate the execution time of the above steps, a series of experiments (10 runs) were carried out in various date/times (Figure 17). The average execution time of the Early Warning component was estimated to be 35.6 ± 0.85 seconds. In total, the number of forecasts that acquired was 90 values that predict the Fire Weather Index and 480 weather forecasts. Also, 48 messages with analysis results were generated and forwarded to PSAP or beAWARE dashboard. It is worth to note that in the Step 2 exists an overhead around 5 minutes in order to get the netCDF files. The time that is presented in Figure 16 for this step, corresponds to the time that the Early Warning component needs to process the file and extract the useful data. Finally, the execution time of each step during the fire pilot is included in Figure 16. The time of each step, in that case, is comparable with the average time per step.

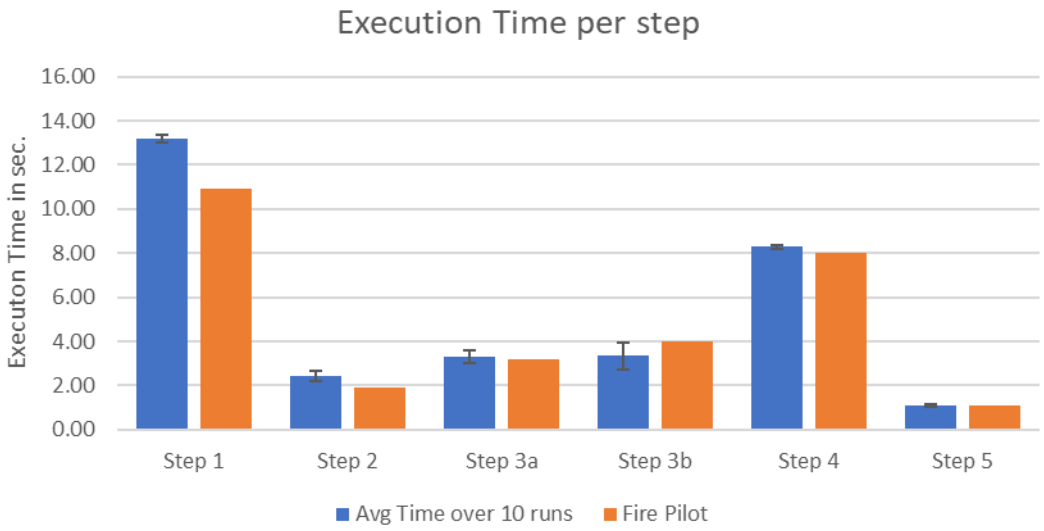


Figure 17: Execution Time of Early Warning component

3.4.2 Evaluation of the Real-Time Monitoring and Risk Assessment component

During the emergency phase, in order to monitor the weather evolution and particularly weather parameters that affect the fire behaviour, such as air temperature, humidity, wind speed and direction, the Real-Time Monitoring and Risk Assessment component is designed to fetch weather observations and present them to the dashboard along with historical weather values and short-term forecasts. For the needs of the Fire pilot, the historical (24 hours before) and real-time weather observations are obtained from two different resources, the State Meteorological Agency[2] (AEMET) OpenData API and from Dark Sky API at specific locations in the Valencia region, as described in D2.8. The short-term predictions are obtained by HIRLAM model of FMI at the pre-defined locations. The average execution time for the whole process is around 2.5 minutes.

The final version of the *Real-Time Monitoring and Risk Assessment* component had enhanced with a new risk assessment approach which is described in details in D3.4 and tested during the Fire pilot in Valencia. The goal of this approach is to assess the severity level of the ongoing fire crisis event dynamically. For this purpose, each time where a new incident with multimedia content is analysed by the corresponding Data Analytics module (i.e. IMAGAN, VIDAN, etc.), the *Real-Time Monitoring and Risk Assessment* component receives the outcomes, estimates both the severity level of the incident and the cluster that it belongs to. Then, aggregates the severity levels of all the clusters in the fire zone, the component assesses

[2] http://www.aemet.es/es/portada

the fire zone's severity level. The results of the analysis are presented in the gauge and traffic light plots in the beAWARE dashboard.

During the fire pilot, the component received 5 clusters of incidents in fire zone 1, which contained images among their members. The Risk Assessment algorithm activated to estimate the severity level of each participant according to the image analysis results. Then, the severities levels of the objects in the images were fused and the cluster's severity level was assessed. The results are presented in Table 1.

Table 1: Results of the Risk Assessment algorithm in Fire Pilot

#Cluster	Incidents - Members	Cluster Severity	Zone
1	1 Image (No Participants - Severe)	Severe	1
2	1 Image (No Participants - Severe)	Severe	1
3	4 Images i. No Participants – Severe ii. No Participants – Severe iii. 2 Humans – Severe iv. 1 Human – Severe, 1 Wheelchair user - Severe 1 Audio 12 Text messages	Severe	1
4	3 Images i. 2 Cars – Unknown ii. 1 Human – Extreme iii. 4 Human – Severe 21 Text messages	Severe	1
5	1 Image (No Participants – Severe) 3 Text messages	Severe	1

Finally, 27 messages were produced and proceeded to PSAP in order to update the severity level of the cluster of incident and simultaneously 2 messages were generated in order to update the severity level of the whole zone and present the results to the dashboard in the corresponding gauge and traffic light plots. The average execution time for the whole process do not exceed the 2.5 seconds.

3.5 **Text Analysis**

The technical report for the second pilot (D7.6) presented an automatic quantitative evaluation of the reports produced by TA from a set of tweets in English and Italian sent to the system and related to the flood emergency scenario. The evaluation was carried out separately for the disambiguation and the concept extraction components, and F1 scores were reported for each. An additional qualitative evaluation was conducted on the final conceptual representations produced by the module. At the time of writing D7.6, no evaluation had been yet reported using 2nd pilot texts. This time, an evaluation of TA using 3rd pilot materials has already been presented in D3.4, so in this document we will only compare the results reported in D3.4 for the 3rd pilot to those presented in D7.6 for the 2nd pilot.

	2ND PILOT (FLOOD D7.6)		**3RD PILOT (FIRE D3.4)**	
	English	Italian	English	Spanish
BFS	0.64	0.40	0.40	0.71
UPF	0.78	0.60	0.56	0.80

Table 2: comparison of disambiguation results between 2nd and 3rd pilots

F1 values for the disambiguation component applied to English and Italian tweets were 0.78 and 0.60 respectively, compared to 0.64 and 0.40 scores of the best first sense baseline. The values for the larger sets of English and Spanish tweets in the 3rd pilot evaluation where 0.56 and 0.80 respectively, compared to 0.40 and 0.71 scores of the best first sense baseline. The drop in F1 scores for English affects both the UPF disambiguation component and the baseline and is due to the 3rd pilot texts being longer and with more lexical variety. The highest results are those of the Spanish texts belonging to the 3rd pilot and is the consequence of our efforts in optimizing the component for this language before the 3rd pilot. Comparing the system results to the baseline, the increase in performance ranges from 9% to 20%, well above the lowest expectations set in the self-assessment plan and surpassing the highest expectations in the case of Italian.

	2ND PILOT (FLOOD D7.6)		**3RD PILOT (FIRE D3.4)**	
	English	Italian	English	Spanish
DBPEDIA SPOTLIGHT	-	-	0.73	0.67
UPF	0.79	0.71	0.76	0.71

Table 3: comparison of disambiguation results between 2nd and 3rd pilots

In the case of the concept extraction, F1 values for English also dropped from the 0.79 of the 2nd to the 0.76 of the 3rd pilot, albeit not as sharply as in the case of disambiguation. Results for Italian and Spanish were identical (0.71). Comparing to the mentions to entities detected by DBPedia Spotlight, our component shows a 3% and 4% improvement for English and Spanish respectively. These results are slightly below the expectations set in the self-assessment plan but should be treated with caution due to the small size of the datasets used for the evaluations.

	3RD PILOT (FIRE D3.4)	
	F1	Geolocated tweets
STANFORD CORENLP	0.58	0.57
UPF	0.58	0.64

Table 4: geolocation results for the 3rd pilot

Geolocation is a relatively recent addition to the text analysis module. While already present in the 2nd pilot version of the analysis module, for performance reasons it was excluded from the evaluation in D7.6. Its evaluation for the 3rd pilot in D3.4 was conducted using a single multilingual dataset containing texts in both English and Spanish, and resulted in a F1 score of 0.58 identical to that of the Stanford CoreNLP state-of-the-art baseline. However, when looking at the ratio of tweets with at least one location correctly geolocated, our system outperforms the baseline by 7 points.

Manual, qualitative evaluations of the output of the analysis module were conducted for the 2nd and 3rd pilots and reported in D7.6 and D3.4. The main advances were due to improvements in the concept detection, disambiguation and geolocation components, which resulted in a wider coverage of the mentions to incidents, impacted objects and locations being detected in the multilingual inputs. As a matter of fact, out of 23 errors detected in the outputs considered in D3.4 only one involved not detecting an incident mention and 3 involved not detecting locations. All other errors where related to states not being correctly extracted, which was a new functionality tested for the first time for the final pilot. It is significant that out of the remaining 19 errors only 5 involved missing hypothetical status of associated events -e.g. risk of fire. This shows that the text analysis module contributes towards verifying the reliability of inputs received by the beAWARE platform.

3.6 Automatic Speech Recognition

The final version of the model has already been evaluated in D3.4. In this deliverable we demonstrate the performance of the ASR component along with the integrated call center during the Fire pilot. However, even though a large set of emergency-related phrases had been prepared for the pilot and additionally the end users had been trained on the use of the platform, however a small amount of audio messages and calls was recorded during the pilot. This could be explained by a network unavailability that was reported by several users. Specifically, 8 audio files were analyzed: 3 audio messages through the Mobile App and 5 emergency calls through the call center. In general, recognition performance on audio coming from Mobile App is significantly better than on audio from the call center. This is because, both the dedicated call center that was integrated to beAWARE platform and the call center of PLV use a third party recording application, which is not possible to be modified. Unfortunately, the recording audio quality of these recorders is very low (Bit-Rate~=13-

16kbps), contrary to the audio quality of audio coming from the Mobile App (Bit-Rate>=256kbps) and the recording frequency is 8kHz, even though the recognition model is trained on 16kHz speech. Nevertheless, even though these recordings had a high average %WER=71%, the recognizer was able to recognize the location and the incident (smoke and fire) in three of them. This is a good showcase of the ability of integration of a call center solution in the platform and indicates the potentiality of even more accurate results, by overcoming a few restrictions of third-party apps. On the other hand, audio coming from the Mobile App had a very low average %WER of 15.1% and locations and incidents were successfully detected.

Regarding time efficiency, since the recognition process is the same for both the audio messages through the Mobile App and the emergency calls through the call center, the required time for analysis (t_a) has no significant differences. The only difference is on the initial period (t_i) required from the creation of the recording until the submission of the file to ASR. Again, the Mobile App has a better performance on this, since it takes less than 1 second until the creation of topic 021 ($t_{i<1}$<1sec). Then, analysis time depends on the length of the audio file. For an audio of ~6secs for example it takes around 20secs. In the case of emergency calls, on the other hand, the audio files are initially stored on a FTP server. Then, with the use of a script that periodically checks the server for new recordings every 6 seconds, the files are uploaded to beAWARE storage server and a message is sent to the message bus, as a 021 topic. Consequently, t_i depends on the time interval between two requests to the FTP server and an average value is 5.5 seconds.

3.7 Visual analysis

For this technical evaluation report, we have gathered all images and videos that were uploaded to the system during the 3rd pilot. It is important to note that even though the pilot was designed to test the final version of the system, the visual analysis components received far less requests compared to the preceding pilots. Specifically, 14 images in total were sent for analysis, including some simulated data that was prepared beforehand in order to ensure that specific aspects and functions would be successfully showcased. Nevertheless, we report the average download, upload and processing times in Figure 18. The component's final version achieves similar speed of operations compared to the version tested during the 2nd pilot, despite the added processing demands of the new functionalities.

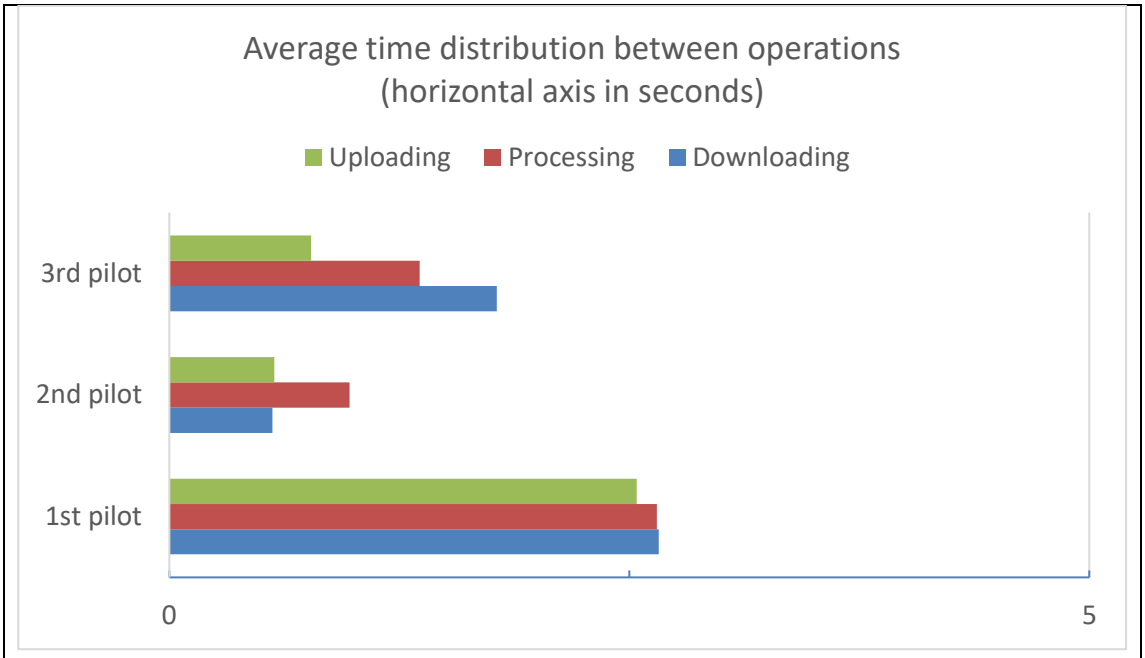


Figure 18: Speed comparison of Visual Analysis on all three pilots.

3.7.1 **Final Version of the Emergency Classification (EmC)**

The final version of EmC was described thoroughly on D3.4. We have already covered extensively the evaluation results on that report, but we also include some of them here for self-containment. Figure 19 presents the normalized confusion matrix for the EmC module. Most of the errors are false positive cases and some false negatives. Interestingly, there is very little confusion between the emergency classes. Moreover, flood is almost never confused with fire or smoke instances. This means that even though there is one unified EmC model for all emergency cases, visual analysis will rarely confuse emergency events.

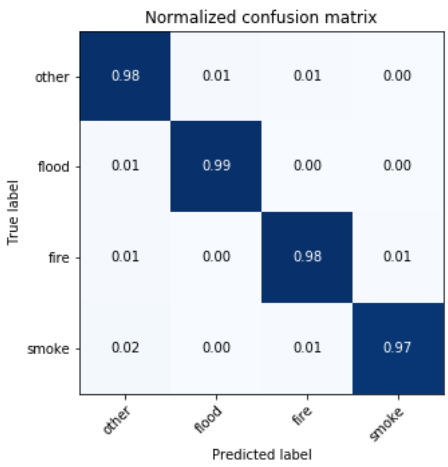


Figure 19: EmC final version confusion matrix.

Qualitative evaluation on the 3rd pilot data follows so as to examine the performance of the rest of the functionalities. Figures Figure 20-Figure 22 show some analysed images from the pilot. Each figure shows images that have been classified by the EmC to a particular category. Notice that in Figure X all the images contain flames, and as such the EmC classifies them as ‘fire’ despite that smoke may be the most dominant texture in some of them. In Figure 21 a false positive case is shown at the right. A fire fighter is depicted throwing water, and is accurately detected by the Object Detection module, but the EmC has misclassified the image to the ‘smoke’ class, possibly due to the pouring water texture which is similar. The image on the left shows some cars and people detected in a smoke incident. Figure X is a compilation of images that EmC found no emergency, but nevertheless contain people or vehicles which are accurately detected.

Figure 20: Images classified as 'fire'.

Figure 21: Images classifies as 'smoke'.




Figure 22: Images classified as 'other'.

3.8 Drones

3.8.1 Drones Platform

The drones platform demonstration highlighted the main capabilities provided by the platform, namely route planning, configuration of flight parameters (such as height and camera angle), autonomous piloting, data sharing in real-time, and dynamic operation of the flight. During the entire flight information flows from the drone to the platform dashboard via the iOS-based client device, including the route of the current stage and imagery transmitted by instruments on the drone.

During the fire pilot the drone platform had three missions.

1. Smoke detection – during the pre-emergency phase, when the conditions were reported as risky for fire scenarios, the drone was deployed to scan a forest area and detect smoke or file. The drone scanned a pre-defined area constantly sending video chunks to the beAWARE platform (see Figure 23). The videos were analysed by the Drones Analysis module, and upon detection of smoke, a corresponding message was posted on the message hub by the Drones Analysis module. That message was picked up by the drones platform and the corresponding analysed video was displayed on the drones platform dashboard. In Figure 23 we can see a view of the drones platform dashboard. On the left-hand side, we can see the route covered by the drone, over a map, including the points which identify its planned route. On the right-hand side, we can see the footage coming from the drone.
2. Identify a person in danger – as can be seen in while performing another scan of the forest area the drone identified a person in danger (fell of a bicycle). Videos were sent in real-time from the drone to the Drones Analysis module. Upon identification of a person in danger a corresponding message was sent through the platform message bus. The metadata sent includes the person location, so a rescue team could be sent to the correct place and rescue the person.
3. Support the school evacuation – Once the school evacuation was ordered and was taking place the drone was sent to scan the school area and identify any remaining people (Figure 25). During a first scan the Drones Analysis module identified a person in the area, and the corresponding analysed video is shown at the right-hand side of the figure. After a team was sent back to the school to evacuate the last remaining people, the drone was called in for another scan which found no people in the area. The school evacuation was than determined to be complete.

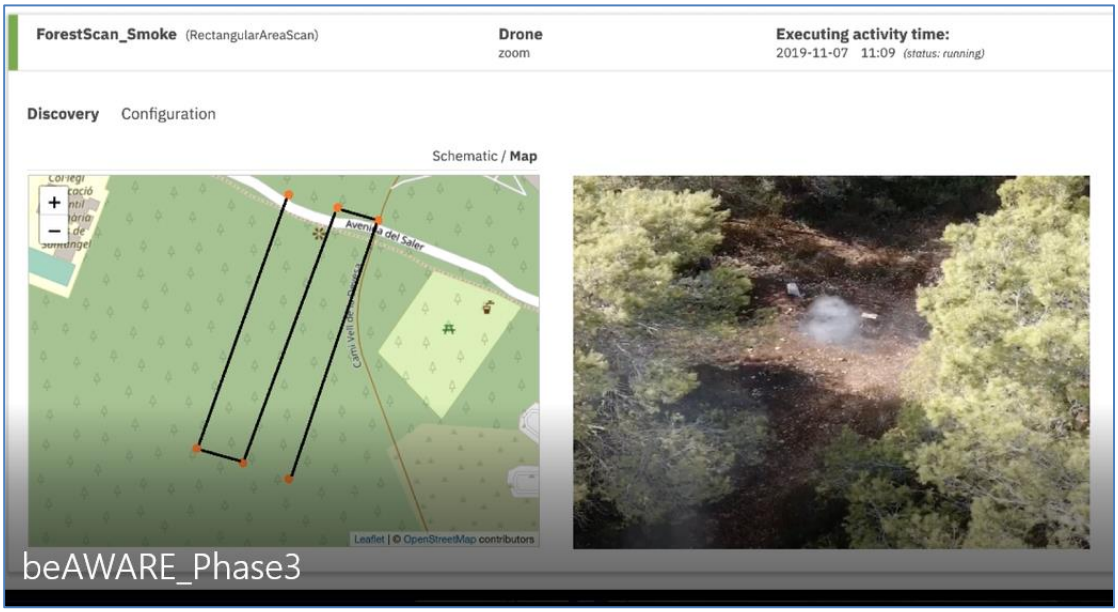


Figure 23: smoke detection with drones

Figure 24: drone footage of person in danger

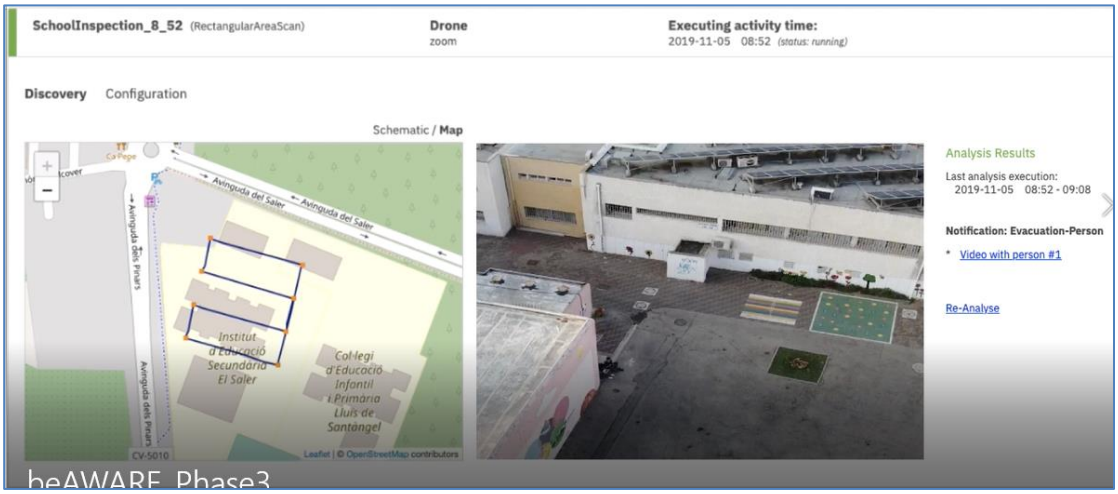


Figure 25: Drones support the school evacuation

Activities to gather current performance numbers were carried out based on the characteristics of the fire use case carried out in Valencia. The activities being tracked include sending video files to be stored on a cloud-based object storage and sending a corresponding message on a cloud-based message bus, which includes a link to a video file stored earlier in the object storage. The Drones Analysis module picked up these videos, analysed them, and sent back messages when interesting information was detected.

Overall 296 video files were sent, corresponding to about 15 minutes of video. The average size of an individual video file was 446,899 bytes, with a standard deviation of 146,257 bytes. The average time it took to submit a video file to the Object Storage was 822 ms, with a standard deviation of 113 ms. Note, that at the time of the performance testing a simple ping to the object storage server, corresponding roughly to the network latency between the client and the server, took 69 ms in average.

Finally, the duration of message submission to the message bus was: 228 ms on average, with a standard deviation of 33 ms.

From these numbers we can conclude that the system at its current configuration can sustain the rate in which video files are fed into the system from the drone.

3.8.2 **Drones Analysis**

Due to Spanish legislation on the use of unmanned aerial vehicles, it wasn't possible to perform an actual drone flight during the pilot in Valencia. Thus, the demonstration of the drone's functionality was performed by using pre-recorded drone footage on the premises of IBM in Israel and subsequently video metadata were fixed to match the location of Albufera Natural Park in Valencia. As mentioned above, 296 video files were transmitted and analysed by Drones Analysis during the third pilot. Each file was a part of a larger sequence and it had an frame rate of 10fps and a duration of 3 seconds. After sub-sampling resulted video chunks

had a frame rate of 5fps, containing 15 frames each. Analysis was focused on three different tasks: a) the detection of smoke in the forest, b) the detection of an injured person and c) the inspection of a school yard under evacuation in order to detect for people. For time-saving during the pilot, in the third task, only the case where the school was deserted was examined (no trapped people were present) in order to mark the end of the evacuation mission and demonstrate the creation of relevant evacuation reports, since the object detection ability was anyway presented in the first task. Thus, the ground-truth annotations of the third task didn't contain any instances of people. The following figures depict some characteristic examples of analysis results for the three tasks. Figure 26 presents examples of the image classification model that was trained to detect hazards in images. The model contains four classes: *'smoke', 'fire', 'flood'* and '*other'*, in case no hazard is detected. Results in Figure 26 are from a sequence that contains smoke sent by Drones Platform. It should be noted that this is a challenging sequence, because the smoke covers a small portion of the image. The upper image is an example of a frame correctly classified as '*smoke'*, probably because the smoke spreads in a wider region of the image, whereas the lower image is misclassified as '*other'*, probably because the smoke is more concentrated in the center. Similarly, Figure 27 presents examples of the object detection model on a video sequence with an 'injured' person. The upper image is an example frame with a correctly detected 'person', whereas in the lower image the person was missed probably because of the distance and the irregular posture.

Finally, Figure 28 depicts a video frame from a video sequence during an evacuation mission, without the presence of any trapped people. During the whole processing there were no false positive detections and consequently, no alerts were created during the mission. At the end of the mission, the Drones Platform sent an 'End-of-mission' flag at a specified field of the communication topic, which was forwarded to KBs along with analysis results. Subsequently, KBs requested from Report Generator an overall report for the whole mission, which in turn produced the following report in Spanish:

"*Misión de evacuación completada. No se han detectado personas.*" - (Evacuation mission completed. No people have been detected).

Figure 26. Example of an image classified as *smoke* (upper) and an image classified as *other* (lower)

Figure 27. Example of a correctly detected target (upper) and a misdetection (lower)

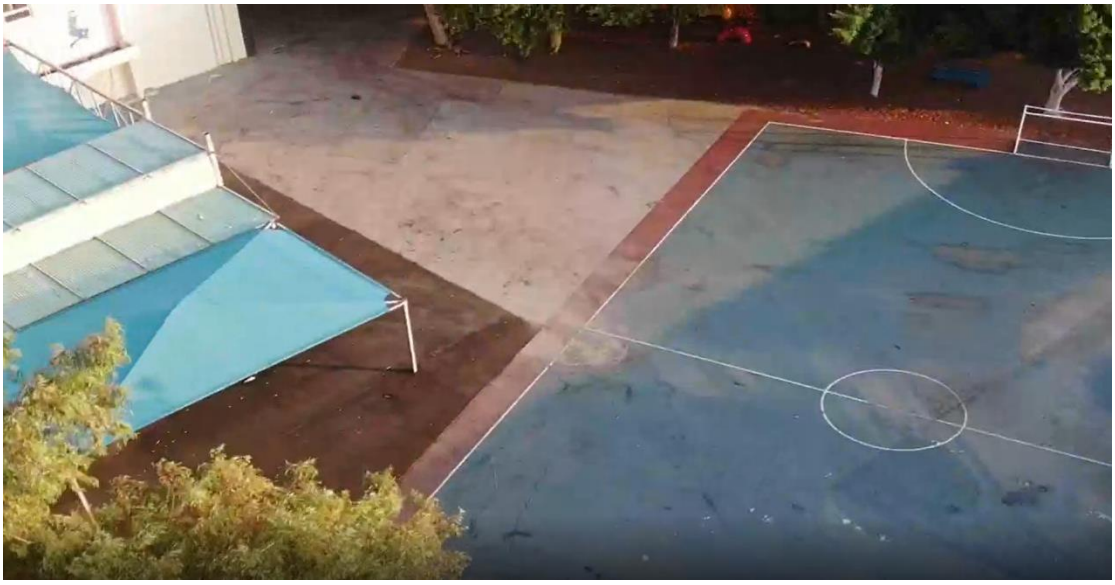


Figure 28. A video capture during an evacuation mission at the yard of a school without any trapped people present.

Table 5 presents quantitative evaluation results of the R-CNN based object detection model that was trained in order to detect people and vehicles from drone footage. As evaluation dataset only the video sequence with the injured person was used, since the evacuation sequence did not contain any target object. Out of 48 video files, 88 frames contained instances of a person. Extracted frames were manually annotated in order to create ground-truth images. Subsequently, the Average Precision (AP)[3] was calculated by estimating the area under the curve of the precision-recall curve. The Mean Average Precision (mAP) is defined as the mean AP over all classes, which in this case coincides with AP. The slightly low precision of the model on the specific sequence is justified by the long-distance view and the irregular posture of the person in several instances. However, even though there are some miss-detections, since the results in drone missions act cumulatively, the detector's performance guarantees the successful detection of the injured person.

Table 5: Object Detection performance of Drones Analysis.

	Class	**mAP%**
	Person	
Average Precision	**59.10%**	**59.10%**

[3] https://github.com/rafaelpadilla/Object-Detection-Metrics

For the evaluation of the image classification model that is used for detecting fire, smoke and flood, the drone 'smoke' sequence of the pilot was used. Out of 48 video files, 231 frames were annotated as '*smoke'* frames and 489 were annotated as '*other'*, which means that they didn't contain any hazard. Figure 29 depicts the normalized Confusion Matrix[4] of the image classification. Results show that 75% of the frames containing smoke were correctly identified, whereas there was a 15% of FNs and 10% of miss-classifications, with the higher portion of these to be classified as '*fire'* (as expected, since usually most fire images contain smoke). The problem with this sequence is that, since it is a simulated fire for the needs of the drone pilot, the scale of the fire and consequently the spread of the smoke is limited, thus in many frames the proportion of the fire in the image is low. This shortcoming along with the different angle view of the drone compared to a terrestrial camera can justify the relatively low performance compared to the performance of the model when used by the Emergency Classification (EmC) Visual Analysis component (see D3.4). However, despite these restrictions, still a 84% of the frames of this sequence that contained smoke was classified as either '*fire'* or '*smoke'*, which guarantees that sometime during the drone mission an alarm is generated for possible fire.

Normalized Confusion Matrix

	Smoke	Flood	Fire	Other
Smoke	0.75	0.01	0.09	0.15
Other	0.1	0.1	0.0	1.0

Figure 29: Normalized Confusion Matrix over a drone 'smoke' video sequence.

3.9 beAWARE Knowledge Base

The Knowledge Base and especially the ontology is the central point for the semantic integration in the beAWARE platform. The development of the ontology already finished in M18 and a detailed technical evaluation can be found in the previous deliverable D7.6. For the final version, the ontology was extended by a few new sub-concepts and datatype-properties. Since those changes don't affect the performance metrics, defined in section 0we disclaim a new evaluation and refer to the previous, still valid evaluation in D7.6.

[4] https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

The execution of the pilot showed that the knowledge base was able to answer all the competency questions needed during the pilot use case.

3.9.1 Knowledge Base Service

Knowledge Base Service (KBS) is responsible for inserting new knowledge to the Knowledge Base (KB) and for inferring new knowledge from the available knowledge. The KBS's performance affects the availability of the inferred knowledge to the rest of the system, and is useful to be as fast as possible to avoid delayed knowledge. The overall performance depends on three main factors, (A) the processing within the KBS (mostly parsing the messages and forming corresponding requests), (B) the networking communication overhead (establishing connections and sending data via the internet), and (C) the KB's response time (processing the KBS requests and produce the reply). The KB is implemented as an instance of WebGenesis tool.

On this subsection, the execution times for KBS processing are presented. The times are measured from the arrival of a message (via the Kafka bus) until the final operations for the message are finished. In some cases, the measurement is broken down to the different semantic operations done by the KBS in order to distinguish the different effect they have on execution time. Additionally, the performance of the second layer of the two-layer validation system is evaluated.

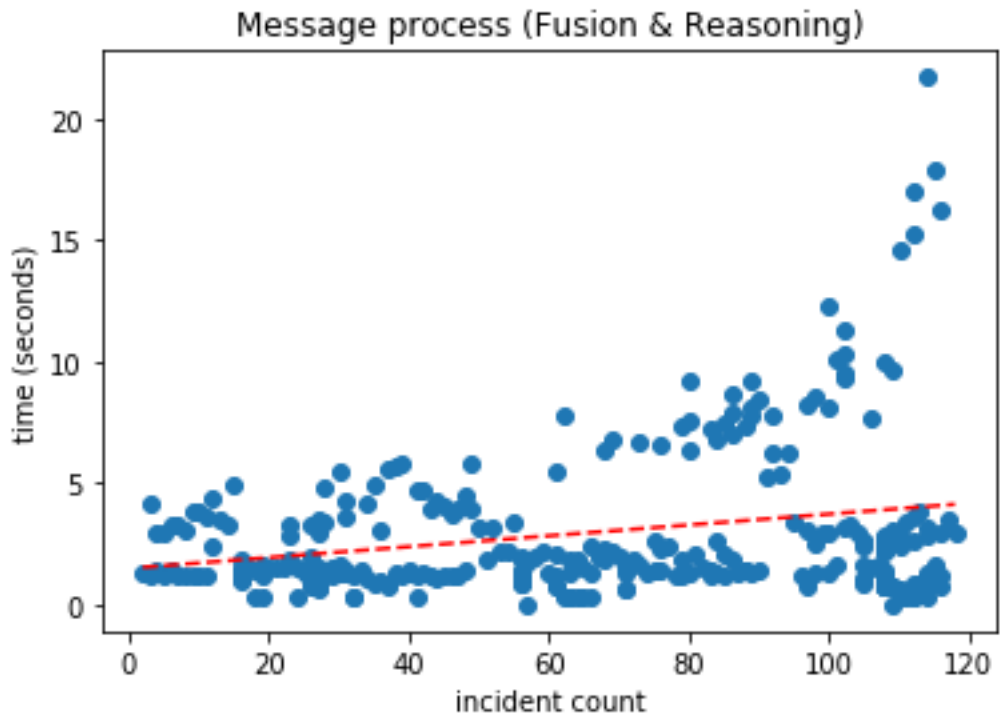


Figure 30: Kafka bus message processing time (by the KBS), in relation to the incident count

Figure 1 shows the processing time for all the messages received by the KBS, in relation to the number of incidents in the Knowledge Base. The majority of the messages were processed in significantly less than 5 seconds. The processing time scales with incident count, however, the rate of increase is low, with few exceptions. The trend line confirms that the increase rate is low in regard with the incident count. The number of messages that were processed by the KBS were 299 and the average processing time was 2.975 seconds.

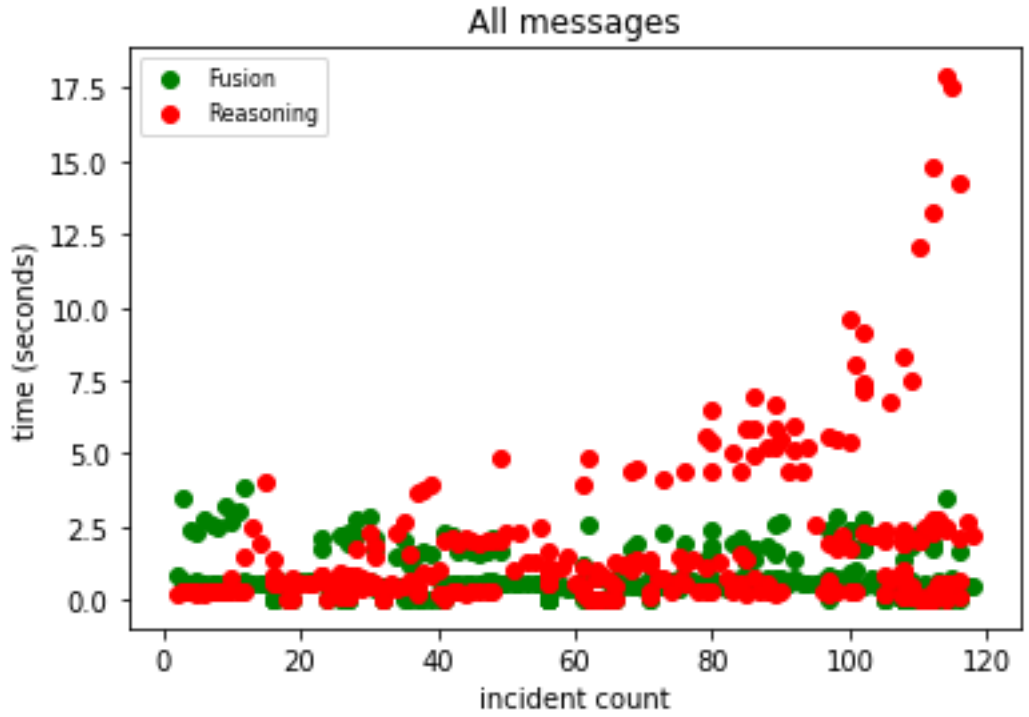


Figure 31: Messages processed by KBS broken down to semantic Fusion and Reasoning operations.

Figure 2 shows the time it took to process the messages broken down to Fusion and Reasoning operations. Fusion (or population) is the process of adding the new knowledge to the KB. Semantic Reasoning (or simply Reasoning) refers to the operations that aim to infer new knowledge from the KB. Within the context of beAWARE KBS, reasoning is done via a set of rules that are applied “on top” of the KB and are implemented as SPARQL queries. In more detail, figure 2 illustrates that the Fusion operations (green dots) remain fast regardless the number of incidents already in the KB, with average 0.791 seconds of processing time. On the contrary, Reasoning operations (red dots) are dependant to the number of incidents in the KB, as they exhibit an upward trend. This is explained by the combinatory nature of reasoning as it automatically explores multiple incidents and their interconnections. Reasoning process in average took 1.871 seconds.

	Average Fusion Time	**Average Reasoning Time**	**Count**

Topic 001, social media text	0.656	0.265	34
Topic 003, social media reports	0.714	3.173	17
Topic 018, image analysis	2.083	7.829	12
Topic 019, UAV analysis	0.79	2.154	16
Topic 021, incident report	0.55	1.078	81
Topic 028, text analysis	1.815	3.049	63
Topic 801, incident validation	0.248	0.19	23

The above table contains the average execution times (in seconds) for Fusion and Reasoning operations, along with the number of the analysis related messages that were consumed by the KBS. Each line shows the metrics for a different category of incoming message. Overall, the fusion operations were executed in less than a second in most cases except for image analysis results and text analysis results. This is explained by the rich information provided the analysis results, which necessitates heavy additions to the KB, thus, is more time consuming. Reasoning operations on average took longer than fusion, due to its combinatory nature as explained earlier. In general, reasoning was executed in reasonable time frame, with the image analysis taking the most time.

Second Validation Step

The second layer of the two-step validation process (or VAL) works in association with the KBS. However, it works asynchronously on the output of the KBS, thus, does not inhibit the effectiveness of the system. In more detail, the system operates as if the VAL does not exist unless a fake message is detected and KBS in informed (via topic 801, incident validation). In this case, KBS updates the Knowledge Base to mark the incident as erroneous, and informs the PSAP to illustrate this to the end users.

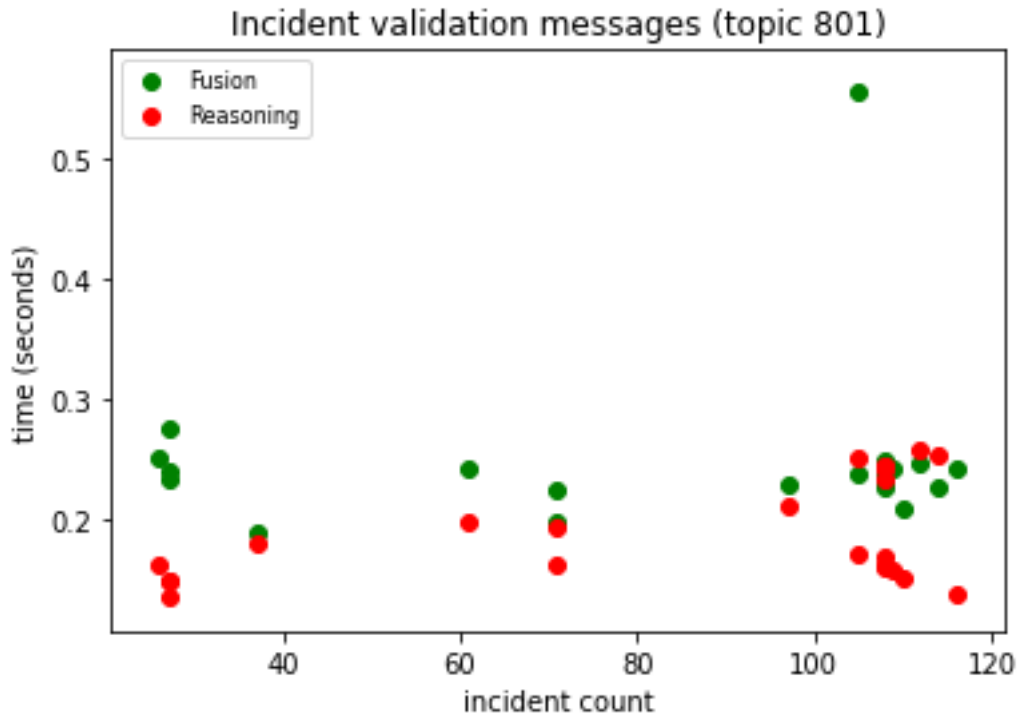


Figure 32: Fusion and Reasoning duration for "incident validation" messages by the KBS

Two sub-procedures contribute to the delay of the validation; firstly, the validation process by VAL that includes exchange of messages with CRCL (for more details read Deliverable 4.3), and secondly, the updates done by KBS. The first sub-procedure took in average 0.636 seconds for each processed message, and the second sub-procedure took 0.438 seconds. Figure 3 shows the execution time of the second sub-procedure, within the context of KBS, i.e. the processing time of incident validation messages (topic 801). With regard to scalability, the validation process remains unaffected by the number of incidents.

3.10 Multilingual Report Generator

For the final prototype, the covered languages were Spanish and English. Since there was no unseen test data to generate from in the framework of the final pilot (using the beAWARE data would have biased the evaluations), we report evaluations on standard datasets. Two types of evaluation were foreseen in D1.3 for the Report Generation module (FORGe): (i) an automatic evaluation using n-gram-based metrics that compares the generated text with a human-written reference (in our case, BLEU, which matches exact words, METEOR, which matches also synonyms, and TER, which reflects the amount of edits needed to transform the predicted output into the reference output), and (ii) a human evaluation according to the quality of the text semantics, grammar and fluency, using Likert scales. The highest expectations according to D1.3 were an error reduction of 10% according to the BLEU score, and Likert ratings of 4/5 or equivalent. In this section, we present the following:

- a qualitative and quantitative evaluations of the FORGe rule-based system in on a standard ontological dataset (WebNLG) to compare to other state-of-the-art generators (the foreseen baseline, the MULTISENSOR generator, is not able to generate from ontologies such as the beAWARE one);
- a quantitative evaluation of the FORGe rule-based system on a large-scale generic dataset to compare to the previous versions of the generator.

The evaluations of the other modules developed within beAWARE but not used in the final system are provided in D5.3.

An intermediate version of the FORGe generator was submitted to the WebNLG challenge, where it obtained the best score according to METEOR and regular scores according to BLEU and TER. However, FORGe outperforms the baseline provided by the organizers for all metrics: +50% for METEOR, 5.9% for BLEU, and 3.7% for TER (see Table 6).

Table 6: WebNLG automatic evaluations

Rank	Team	METEOR	Groups (on sample)	Groups (on all data)
1–2	UPF-FORGE	0.39	A	(A)
1–3	UTILBURG-SMT	0.38	A, B	(B)
2–3	UMELBOURNE	0.38	B	(C)
4–7	UTILBURG-NMT	0.34	C	(D)
4–7	ADAPTCENTRE	0.34	C	(E)
4–7	PKUWRITER	0.33	C	(E)
4–7	UTILBURG-PIPELINE	0.32	C	(E)
8	BASELINE	0.26	D	(F)
9	UIT-VNU-HCM	0.08	E	(G)

Rank	Team	TER	Groups (on sample)	Groups (on all data)
1	UMELBOURNE	0.44	A	(A)
2–5	PKUWRITER	0.51	B	(C)
2–7	UTILBURG-SMT	0.52	B, C	(B)
2–7	UTILBURG-PIPELINE	0.53	B, C	(C)
2–7	UPF-FORGE	0.54	B, C	(C)
6–7	BASELINE	0.56	C	(D)
6–7	UTILBURG-NMT	0.57	C	(D)
8	ADAPTCENTRE	0.72	D	(E)
9	UIT-VNU-HCM	0.84	E	(E)

Rank	Team	BLEU-4	Groups (on sample)	Groups (on all data)
1	UMELBOURNE	48.05	A	(A)
2–3	UTILBURG-SMT	45.90	B	(B)
2–3	PKUWRITER	43.71	B	(C)
4–6	UPF-FORGE	40.03	C	(C)
4–7	BASELINE	37.81	C, D	(E)
5–8	UTILBURG-PIPELINE	37.34	C, D	(D)
4–7	ADAPTCENTRE	36.73	C, D	(F)
7–8	UTILBURG-NMT	35.98	E	(D)
9	UIT-VNU-HCM	5.25	F	(G)

In terms of human evaluation, FORGe got the best evaluations for all criteria, being only outperformed by human texts (WEBNLG). The average score for the three criteria is 2.5/3 (12.5/15), above the highest expectation of 4/5 (12/15); see Table 7.

Table 7: WebNLG human evaluation results

Semantics	Avg	Groups
WEBNLG	2.61	a
UPF-FORGE	2.47	b
MELBOURNE	2.39	c
PKUWRITER	2.39	c
BASELINE	2.36	c
ADAPT	2.31	c
TILB-PIPELINE	2.19	d
TILB-NMT	2.16	e
TILB-SMT	1.96	f
UIT-VNU	1.39	g

Grammar	Avg	Groups
WEBNLG	2.77	a
UPF-FORGE	2.68	b
TILB-SMT	2.42	c
ADAPT	2.30	c
MELBOURNE	2.30	d
TILB-PIPELINE	2.20	e
PKUWRITER	2.08	f
TILB-NMT	1.99	g
BASELINE	1.86	h
UIT-VNU	1.42	i

Fluency	Avg	Groups
WEBNLG	2.58	a
UPF-FORGE	2.34	b
PKUWRITER	2.34	b
MELBOURNE	2.27	cb
ADAPT	2.26	cb
BASELINE	2.25	c
TILB-PIPELINE	2.07	d
TILB-NMT	2.01	e
TILB-SMT	1.81	f
UIT-VNU	1.38	g

Table 8 shows the results of the automatic evaluation of the final version of the generator in English and Spanish using for each input its corresponding reference text(s). The final system fixes a list of errors identified after the analysis of the evaluation of the intermediate system (see D5.3). The first two rows show that in terms of automatic metrics, the extended FORGe and the WebNLG FORGe have almost exactly the same scores on the English data (which are also very close to the WebNLG scores: 40.88, 0.40, 0.55), that is, in spite of reducing drastically the amount of errors in the generated texts (from 275 to 170 errors on the whole 200-text test set), the improvements are not reflected in the automatic evaluation. To compare English and Spanish results, we calculated the scores using one sentence as reference (only one reference per text is available in Spanish). The English scores drop (third row) due to the way the scores are calculated by the individual metrics (BLEU matches n-grams in all candidate references, and METEOR and TER consider the best scoring reference). In the last row of the table, the scores of the Spanish generator look contradictory: the BLEU is 10 points below the English BLEU with the same number of reference (1), but METEOR is 8 points above, that is, the predicted outputs do not match the exact word forms, but they do match similar words. One reason for the low BLEU score could be the higher morphological variation in Spanish. However, the METEOR score is surprisingly high, actually even higher than the highest METEOR score at WebNLG, obtained by ADAPT and calculated with multiple references (0.44).

Table 8: English and Spanish scores according to BLEU, METEOR and TER, with 1 and All references on the 200-triples test set.

Reference set	BLEU	METEOR	TER
EN ($All_{FORGe-2017}$)	39.87	0.40	0.58
EN ($All_{FORGe-Ext}$)	39.33	0.40	0.58
EN ($1_{FORGe-Ext}$)	29.18	0.38	0.65
ES ($1_{FORGe-Ext}$)	18.68	0.46	0.77

In order to compare FORGe to its state at the beginning of the project, we also run an evaluation on the Penn Treebank English corpus. As reported in D5.3, BLEU English scores when using the general-domain Penn TreeBank corpus have improved from 31.78 points to 35.53 (P2) and then to 39.84 (P3). The three numbers have been obtained by comparing the initial text generation component at the beginning of the project, the basic component used in the second pilot and described in D5.2, and the final advanced version reported in D5.3. At P2, an improvement of 11.8% was reached, and by the final pilot, a 25.36% BLEU score increase was achieved compared to the initial system. In terms of error reduction, the generator went from 68.22 (100 minus 31.78) to 60.16 (100 minus 39.84), or an error reduction of about 12%, better than the highest expectation of 10% defined in D1.3.

3.11 Public Safety Answering Point

The main purpose of PSAP is to oversee the entire emergency management effort and to support the work of the Emergency Operations Center.

MSIL led system engineering and architecting best practices and processes, including requirements engineering, functional requirements definition, system requirements definition, and unified and consistent data exchange protocols. In this section we attempt to present a general technical evaluation of the PSAP component based on the indicators defined in section 2.3.12

The usability evaluation was conducted based on the feedback received from an UI/UX expert that went through the system (the Valencia PSAP version) and examined how the user experience can be improved, including how the information is displayed, the connection between the different modules and how the system is used in a situation where loads of information is fed into the system with reference to an emergency situation that the user needs to understand What is the best and easiest way to manage the event and get an accurate snapshot at any given moment

Figure 16 shows the simulation results when varying the number of incident reports received on the PSAP component.

We selected 25, 50, 100, 200, 350 and 500 (Valencia pilot) incident reports. As expected, the propagation delay is lower when incident density increases. As a conclusion, it is noteworthy to be mentioned that the map and the dashboard performance is strongly related to the speed of the servers that the PSAP is using.

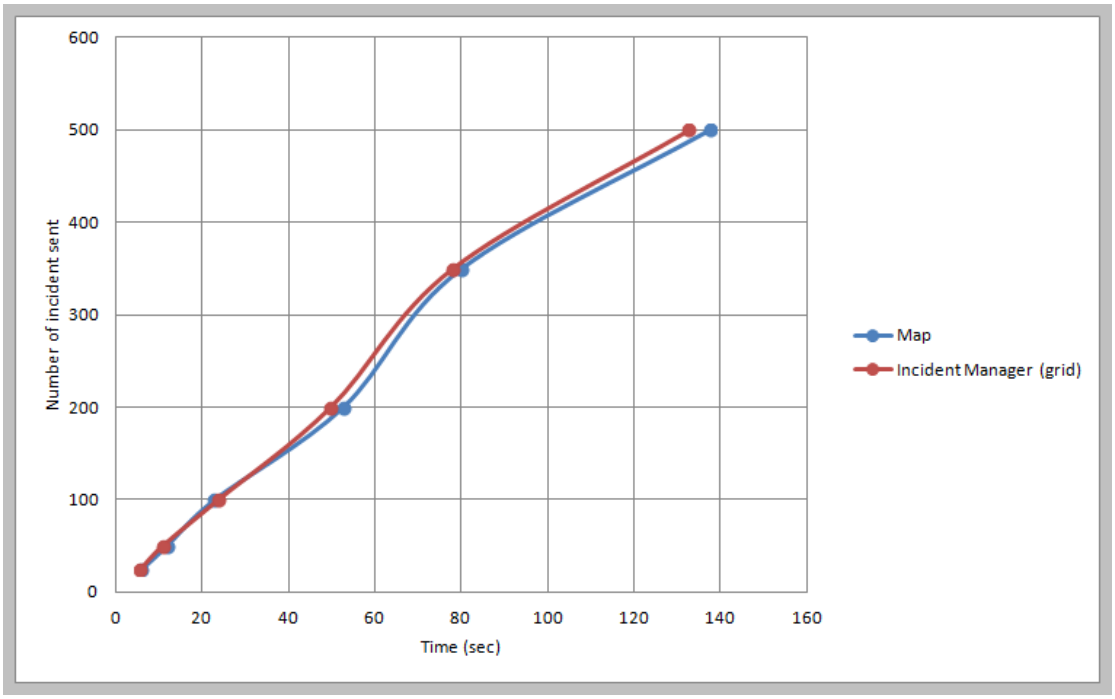


Figure 16: Average visualisation speed when varying the number of incidents received on the PSAP

3.12 Mobile Application

This evaluation of the mobile application is the continuation of the evaluation, done in D7.6. There a table of requirements, gathered from the user requirements in D2.10 was provided. The evaluation showed, that all user requirements were already fulfilled, at least at a basic level, for the 2nd prototype. Therefore, the performance indicator "Number of met requirements" is fully fulfilled.

The mobile application was extended for the final version of the beAWARE platform. The following table will list the user requirements that were affected by the improvements (for a detailed description of the changes in the mobile application, we refer to D7.7).

Table 9

UR#	Requirement name/description	Improvement
UR_103	Flood warnings	Alerts can be restricted to user groups. E.g. only first responders can be set in readiness.
UR_116	Warning people approaching flood areas	A separate notification is shown to people that are approaching an alerted area.

UR_125 UR_131 UR_212 UR_214 UR_215 UR_312 UR_328 UR_336 UR_338 UR_339	(Traffic) warnings, recommendations, evacuation orders	Alert mechanism was extended to allow the selection of specific user groups.
UR_135 UR_227	Specific mobile app for first responder and citizen	Login for first responders to enable all features. First responders can specify their name and profession.
UR_313	First responders status	Teams name and profession can specified.

In addition to the user requirements, the user feedback of the 2nd pilot was picked up and the mobile application was improved by taking those user requirements into account as well. The feedback and the corresponding improvements are summarized in the following table:

User feedback	**Improvement**
Expecting feedback from control room when sending an incident report.	There is an indicator in the list of reports, showing if the message was sent successfully to the beAWARE platform. If an error occurs, the user will get a notification.
Expecting feedback when communicating the status of the task execution.	A notification is shown after the update of the task status was sent.
Capability to send photos / videos etc. after the conclusion of the task.	Using the incident report mechanism and selecting the corresponding category, first responders can send photos, videos or an audio message after concluding a task.

Capacity to display also the reports from the other teams.	We decided not to implement this feature to keep the user interface as simple and clear as possible.
See task status of other teams.	We decided not to implement this feature to keep the user interface as simple and clear as possible.
The app drained too much battery.	Technical improvements made to make the applications execution more efficient.
Implement a ringtone / alarm that will attract the attention of the mobile app operator.	This is improved by fully making use of the native notification system of the mobile device.
Delete the pop-up menu about the incident reports with double-clicking on the map;	Changed in the improved user interface.
The app should show notifications even when it is closed	This is improved by fully making use of the native notification system of the mobile device.
It would be very useful if each notification of new public alert and task assignment has a vibration, because sometimes in the street there could be too much noise to hear the alarm	This is improved by fully making use of the native notification system of the mobile device.

Like in the previous report, the evaluation of the usability is done in a qualitative way, based on the feedback of the involved people. They provided feedback during the debriefing session directly after the pilot as well as by answering a questionnaire. Since the evaluation is based on the users' responses the results can be found in the deliverable D2.8 "Evaluation report of the final system", which will be finished together with this technical evaluation report.

4 Conclusions

The final beAWARE system has successfully integrated a number of new functionalities, as also has shown many improvements to the existing functionalities based on the recommendations given on the previous prototype.

To this end, significant efforts have been made to optimise navigation fluidity and the user experience of PSAP users, with alerting notifications, use of humanitarian icons, and differentiation of icons per incident type for better understanding. Moreover, given the fact that crowdsourcing information is coming into the system from the Twitter or from the mobile application, in the final version of beAWARE, a mechanism was integrated to validate system's traffic and to minimize the likelihood of malicious data impeding System effectiveness.

With regard to the final Pilot the consortium has jointly analysed the evaluation results and has gathered very useful feedback that will receive special attention for the future steps. In short it is summarised as follows per component:

The technical infrastructure includes a complete CI / CD toolchain from source code to docker based microservices being deployed on a Kubernetes cluster. In addition, an array of cloud-based middleware is made available by the cloud and used by different components for capabilities such as storage and messaging.

The establishment of communication between MTA and SMC, to exploit the locations extracted by MTA was successfully established and demonstrated. Improvements in MTA resulted in wider coverage of the module to go beyond pre-scripted concepts and locations reported in incidents, objects and locations detected in the multilingual inputs. This also had significant impact on the production of the situational reports that became richer and more detailed. Concerning the MRG module it should also be noted that the wrap-up summary functionality was stabilized and adequately demonstrated

The Crisis Classification component has successfully encapsulated new functionalities mostly related to the Fire pilot. The integration of information obtained from the European Forest Fire Information System enables Crisis Classification to calculate the fire danger seamlessly. Furthermore, during the emergency phase, the novel fire risk assessment algorithm, used for the more accurate calculation of the overall level of risk, which relies on the exploitation of the multimedia information coming from the citizens' and first responders', has been positively evaluated in terms of usability and performance.

The ASR component performed well on the provided audio messages by achieving low error rate. But most importantly it showcased a successful integration of a call center solution in the platform and, despite some restrictions affecting the quality of the audio and the accuracy of

the recognizer, it succeeded in detecting the locations and the incidents reported in the challenging audio quality coming from the call center.

Furthermore, the visual analysis components successfully served the purposes of the final demonstration. The fire and smoke enhanced classification model was accurate in most of the cases. Moreover, the extension of Object Detection with the detection of animals and people in wheelchairs was tested and the qualitative results showcase the model's applicability in various conditions, such as harsh illumination changes, and scale variance.

Drones platform service proved to be useful in a variety of different scenarios. Additionally, the evaluation of the performance of Drones Analysis module proved the ability of the module for a correct early detection of possible disasters, such as an outbreak of a forest fire and its usefulness in subsequent search and rescue operations or evacuation missions, which was a newly added feature.

For the future the ease of creating and configuring new services shall be explored, in addition to tighter connection between the platform and the control center such that autonomous flight services could be configured and instantiated directly from the command center.

Overall, the outcomes of the final evaluation presented in this document suggest that the technical components have met and sometimes even exceeded the objectives set at the beginning of the project, by providing innovative technological solutions such as early warnings, DSS, reasoning mechanism and machine learning capabilities to the authorities and first responders to achieve more focused and productive collaboration.