beAWARE

Enhancing decision support and management services in extreme weather climate events

700475

7.6

2nd Technical Evaluation Report

Dissemination level:	Public
Contractual date of delivery:	Month 26, 28 February 2019
Actual date of delivery:	Month 30, 18 June 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:	V1.0
Number of pages:	105
Filename:	D7.6_beAWARE_Second_technical_evaluation_report_2019_06_18_v1.0

Abstract

This document comprises the technical evaluation of the components in beAWARE System. This deliverable is iterative and the current version corresponds to the second release compiled in M24. This version extends the content included in the first release and details the technical aspects of the outcome of the second pilot from a technical performance perspective. The document is structured in two parts. The first part details the overall design of the second version of the platform and the evaluation methodology. The second part details 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	22.03.2019	Document initiation and assignments distribution	CERTH
V0.2	12.04.2019	Initial content included	IBM, CERTH
V0.3	19.04.2019	1st integrated draft circulated for comments.	IOSB, MSIL, CERTH
V0.4	05.05.2019	2nd integrated draft circulated for comments.	UPF, IOSB, CERTH
V0.5	10.05.2019	Complete document submitted for review	UPF
V0.6	12.05.2019	Internal review	UPF
V0.7	15.05.2019	Updated document	ALL
V0.8	15.05.2019	Updating doc	MSIL
V0.9	16.05.2019	Second internal review	UPF

Author list

Organisation	Name	Contact Information
CERTH	Ilias Koulalis	iliask@iti.gr
CERTH	Panagiotis Giannakeris	giannakeris@iti.gr
CERTH	Stelios Andreadis	andreadisst@iti.gr
IBM	Benjamin Mandler	MANDLER@il.ibm.com
IOSB	Philipp Hertweck	philipp.hertweck@iosb.fraunhofer.de
MSIL	Itay Koren	Itay.koren@motorolasolutions.com
CERTH	Gerasimos Antzoulatos	gantzoulatos@iti.gr
CERTH	Manos Michail	michem@iti.gr
CERTH	Panos Mitzias	pmitzias@iti.gr
CERTH	Alex. Koufakis	akoufakis@iti.gr
UPF	Gerard Casamayor	gerard.casamayor@upf.edu

Reviewer list

Organisation	Name	Contact Information
UPF	Gerard Casamayor	Gerard.casamayor@upf.edu
CERTH	Anastasios Karakostas	akarakos@iti.gr

Executive Summary

This deliverable contains the technical evaluation of the second prototype of the integrated beAWARE platform. This report is the second of an iterative evaluation process of the beAWARE development cycle and together with D7.4 delivered in M18 and D7.9 that will be delivered in M36 consist a set of a three-step evaluation study. The aim of the document is to report the technical specifications and performance of the second prototype, describe the current issues and provide guidelines for better performance.

The technical evaluation is based on the assessment plan and the performance indicators that were introduced in D1.1 and D1.3. Compared to the evaluation methodology, indicators and baselines prior defined, in the current document these factors have been further developed and revised in light of the experience gained so far during the project work. Moreover, the evaluation methodology was further extended and new indicators were added to expand the coverage to the newly integrated modules

The present document is structured in two parts. The first part provides an overview of the second version of the beAWARE platform, details the methodology used for the technical assessment along with the performance indicators selected for 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

List of Figures

List of Tables

1 Introduction

1.1 Purpose of this document.

This report details the technical aspects of the outcome of the second prototype as a part of a cyclic process of prototyping, testing, evaluation that is followed for the development of the beAWARE platform. This technical evaluation is centred on the performance of the components delivered in the second version of the platform based mainly on the findings of the second pilot which took place in Vicenza (Italy) on 7th March 2019.

1.2 Structure of the report.

The document is structured in 4 sections.

The Second section details the methodology used for technical evaluation of the components of the second prototype Each subsection is divided in two parts devoted to the definition of: 1) the topics of evaluation together with a concise technical overview for each topic and 2) the indicators and measurements used to conduct the evaluation per topic.

In section 3 the focus of technical evaluation is described mainly based on the input of the 2nd beAWARE pilot that took place in Vicenza. Then, the relevant results and performance indicators are presented for each of the system components.

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

2 Overview and Evaluation Methodology

2.1 Global view

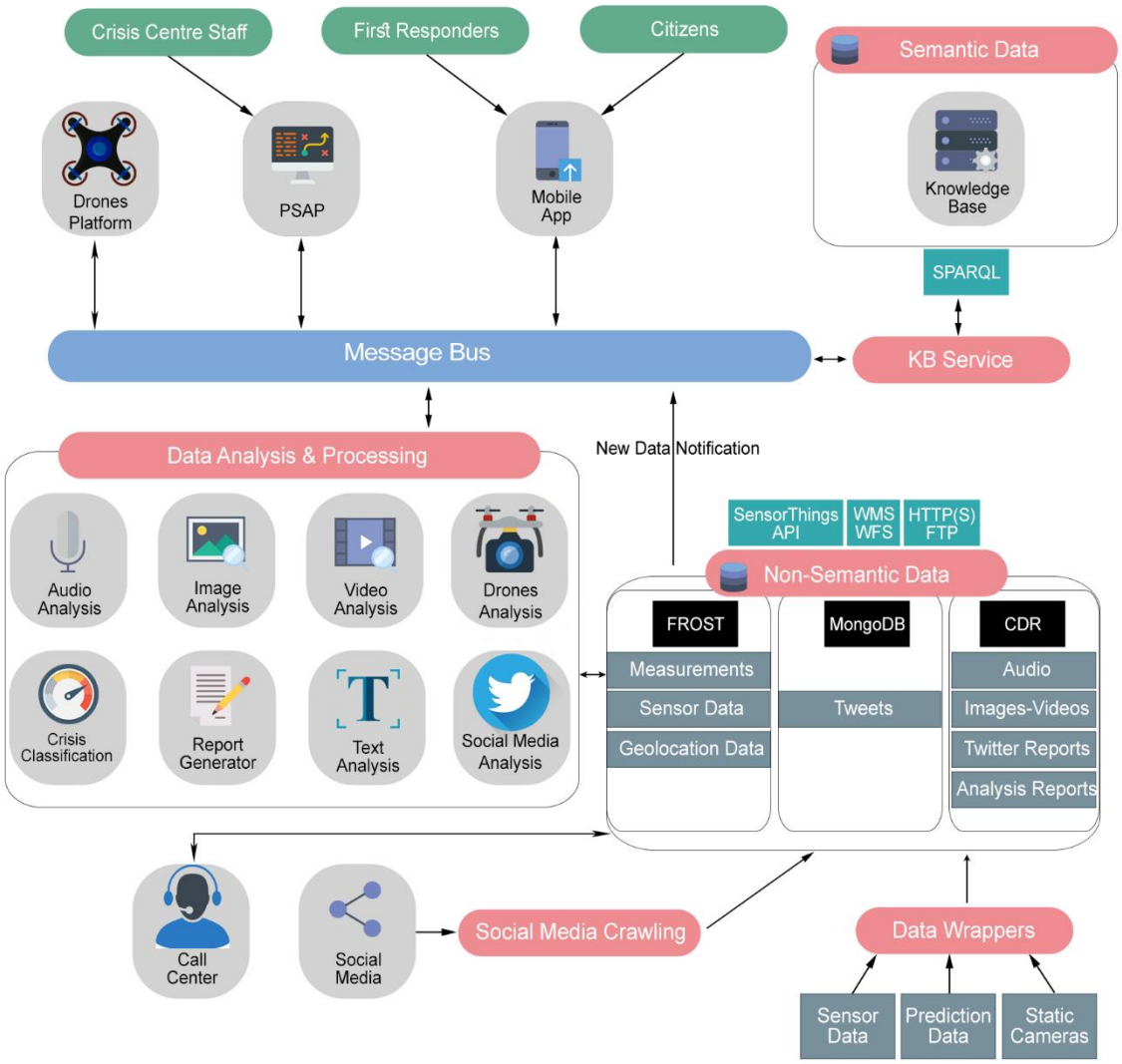


Figure 1: Architectural high-level view

The beaAWARE architecture (Figure 1) 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. (Sections 2.3.1 & 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 & 2.3.4).

3. **Business layer**, containing the components that perform the actual platform-specific capabilities. (Sections 2.3.5 - 2.3.11).
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 will be based on the assessment plan and the performance indicators that were introduced in D1.1 & D1.3. These indicators are refined to meet the updated standards and are presented in the following section divided per evaluation topic.

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.

SMA uses Twitter's Streaming API[1] in order to collect posts in languages of interest (English, Italian, Greek, and Spanish) that contain predefined keywords regarding flood, fire, and heatwave incidents. Following the crawling procedure, a three-step validation process has been introduced in the second prototype, which targets to filter out fake or irrelevant tweets. The first step deals with the present-day problem of hoax news by providing an automatic detection of fake tweets. The second step takes into consideration the emoticons and emojis existing in the posts. For example, a tweet that includes a smiley face or a heart symbol is perceived as unimportant. The third and final step automatically classifies tweets as relevant or irrelevant to the examined use cases, based on their visual and textual information. Although all crawled social media content is stored in the beAWARE MongoDB, only the real and relevant tweets are sent to the Multilingual Text Analyzer (MTA) for concept and conceptual relation extraction and to the Knowledge Base Service (KBS) to populate respective incidents.

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

On the other hand, SMC collects social media-related messages in the beAWARE flow and performs a clustering method on the tweets, based on time and location. The resulted groups are translated to text summaries, named Twitter Reports, and are sent to the KBS to be handled as new incidents. In the second prototype, spatiotemporal clustering depends on predefined coordinates of tweets, but this will change in the final prototype, in order to exploit the extracted locations by MTA.

With respect to the evaluation of the SMA module's performance, the following indicators will be 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.

Regarding the evaluation of the SMA module during the second beAWARE pilot, an additional indicator will be used:

Performance Indicators	Process time
Definition	Process time is the period during which an input is transformed into a finished product by a procedure. Specifically, for the SMA module, process time refers to the number of seconds between the timestamp of the creation of a new post on Twitter and the timestamp at which this tweet is forwarded to the beAWARE system flow, with all the analysis completed.
Domain	Computing
Range	The values of this metric are larger than 0.0, having no upper bound.

Limitations	-

It should be noted here that the SMC's performance can be evaluated using the Normalized Mutual Information (NMI) indicator. However, this evaluation requires an annotated dataset and is a future work that will be described in D4.3 (M34).

2.3.2 **FROST-Server**

The collection and storage of sensor data offer a big potential, since it offers an objective way for decision makers not only to get up-to-date but also to see the previous data. This offers the possibility to determine the evolution of the data.

In the FROST-Server (previously called SensorThings API Server) all available time-series data is stored centrally for the beAWARE project. This contains sensor measurements (e.g. coming from weather stations or water level gauges), water level predictions (coming from AWAA AMICO system) or weather forecasts (coming from FMIs weather models).

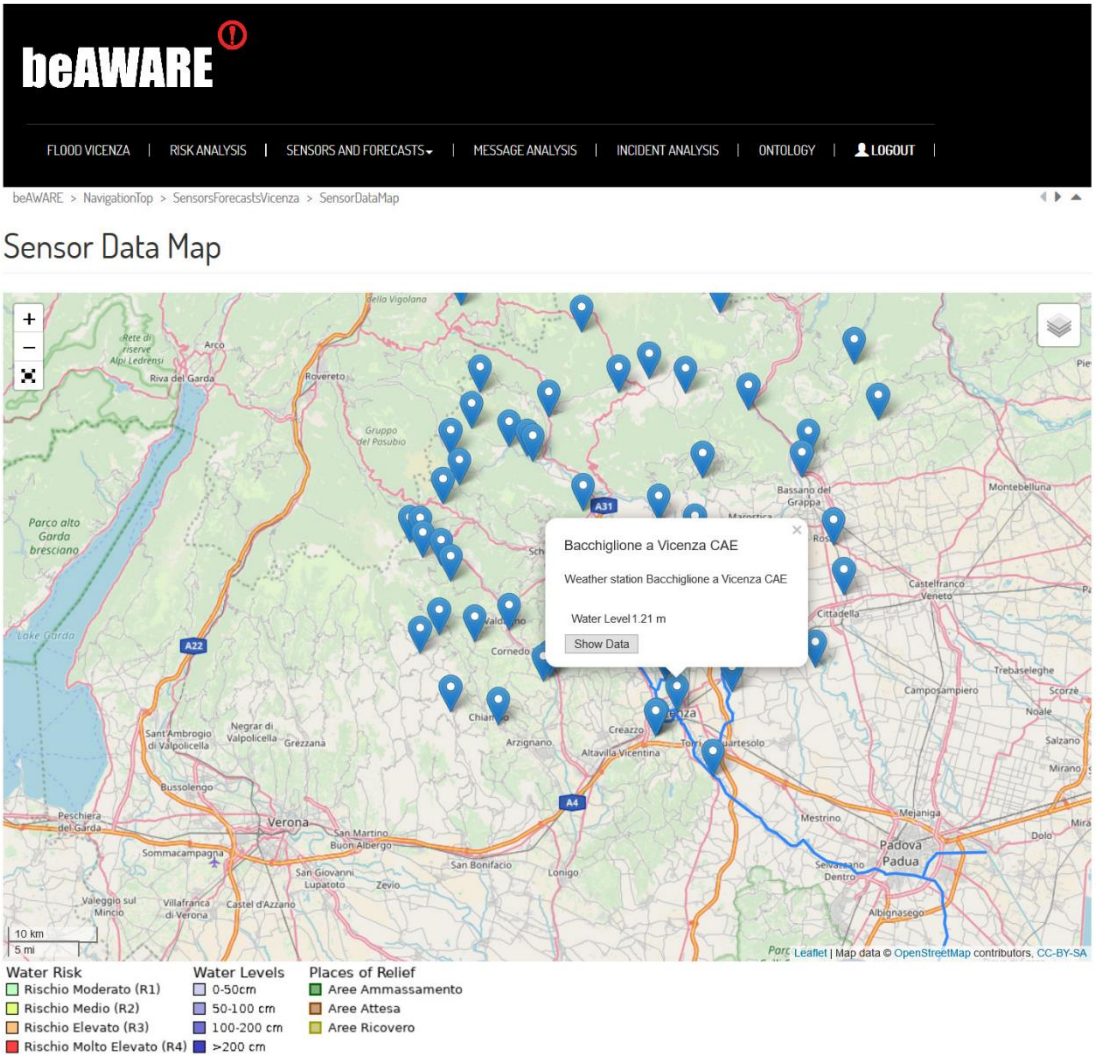


Figure 2: Vizualization of the sensor data map

In comparison to the first prototype, additional data sources have been integrated or the existing ones have been extended. For example, all available weather stations and water level gauges from AWAA have been integrated. In addition, for the 2nd prototype the thresholds for each river section are available, now. Weather forecasts from FMI have been added for all weather stations. The existing mechanism for calculating the minimum, maximum and average values for time intervals was extended to cover the new available data. This allows displaying even large periods, without overburden the client component with too much data points. To analyse the data of the FROST-Server, the visualization was extended to also cover this newly available data (see Figure 2 and Figure 3).

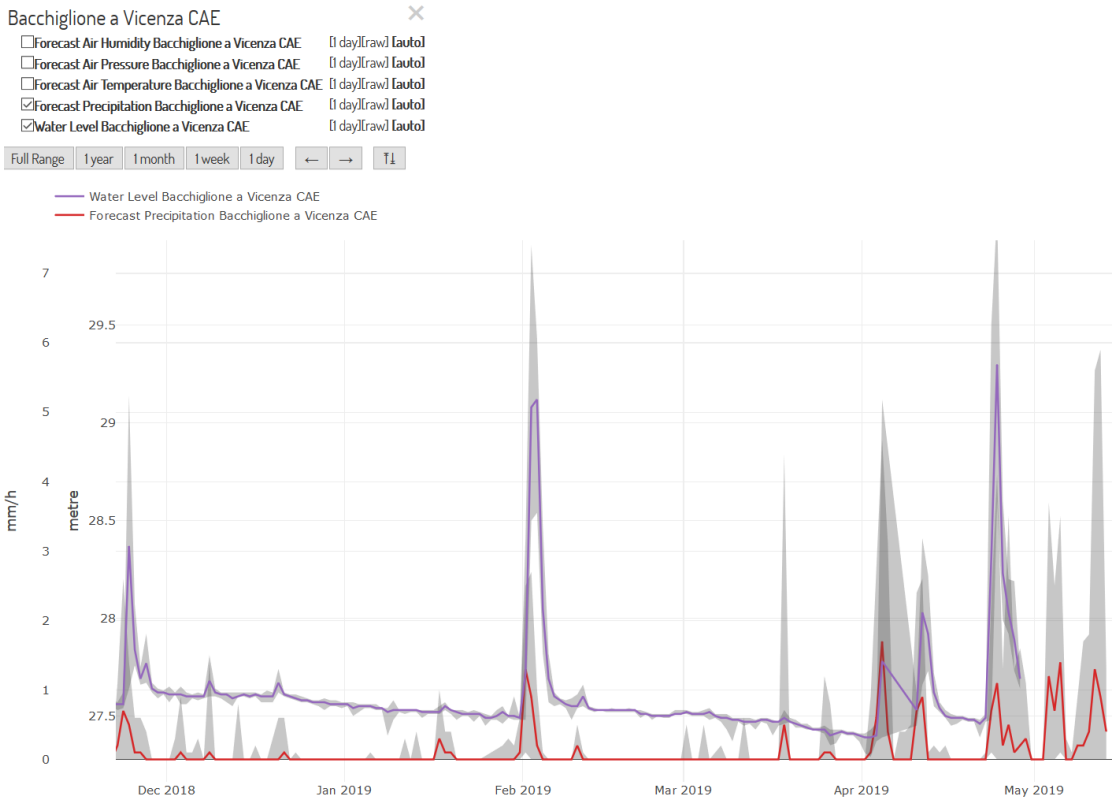


Figure 3: Visualization of the water level measurements as well as forecasted precipitation

For the evaluation of this component, the evaluation strategy, defined in D1.3 page 21 will be used. It is three-fold:

Performance Indicator	**Scalability**
Definition	Able to collect and store large streams of data in a real-time manner. Realtime meaning that the sensor data is retrieved in intervals that are suitable for the use-cases.

Performance Indicator	Correctness
Definition	The data is stored in a database and can be retrieved from it without losing any data.
Performance Indicator	**Integration**
Definition	Heterogeneous sensor data can be queried in a unified way. With identical semantic terms retrieving similar types of data from heterogeneous sensors.
Indicators' Range	**Highest expectation**: All sensor sources available in the project are connected to the platform and their data is stored in real-time. **Lowest expectation**: All necessary sensor data for demonstrating the use cases is available in the data store. The data might not be updated in real-time.

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. In addition, the bus may perform some light transformations on incoming information, upon need.

The second 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.

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 (as can be seen in Figure 4).

The Kubernetes cluster consists of 3 worker nodes, each one having 4 cores and 16GB of RAM, as can be seen in Figure 5. The worker nodes host all the beAWARE microservices, as can be seen in Figure 6, which provides also a glimpse into the resources' consumption in the cluster.

In the second prototype 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.

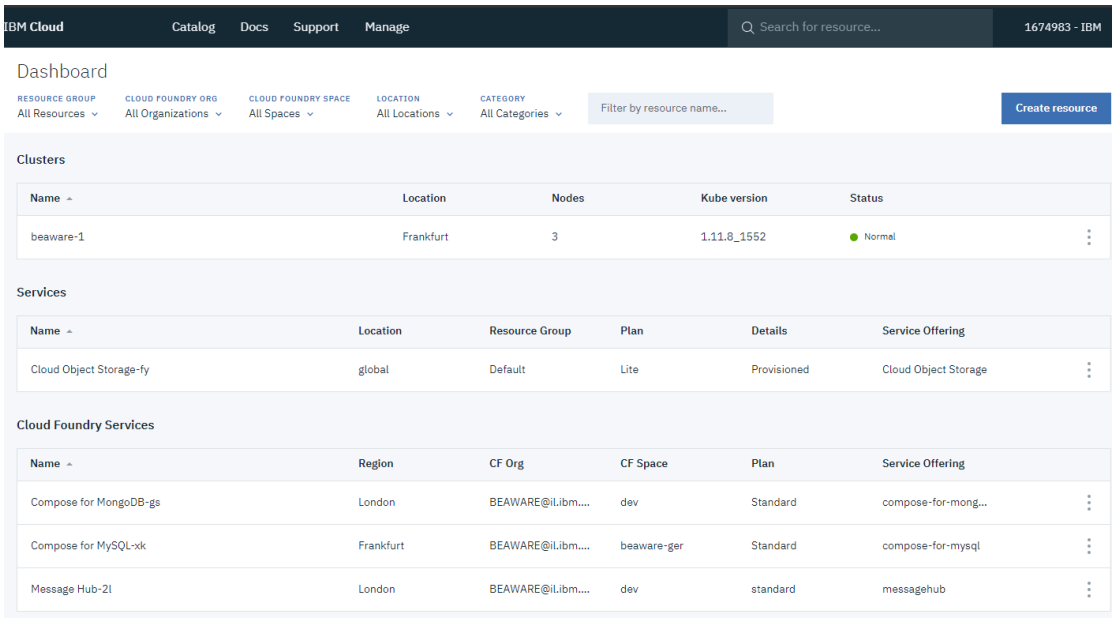


Figure 4: beAWARE technical infrastructure

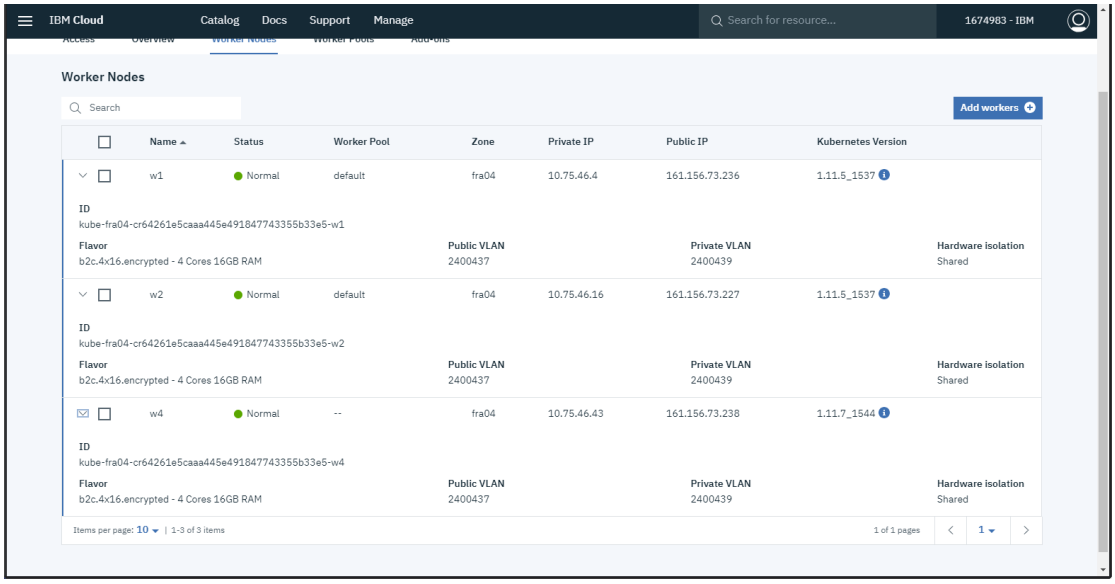


Figure 5: Kubernetes cluster - worker nodes

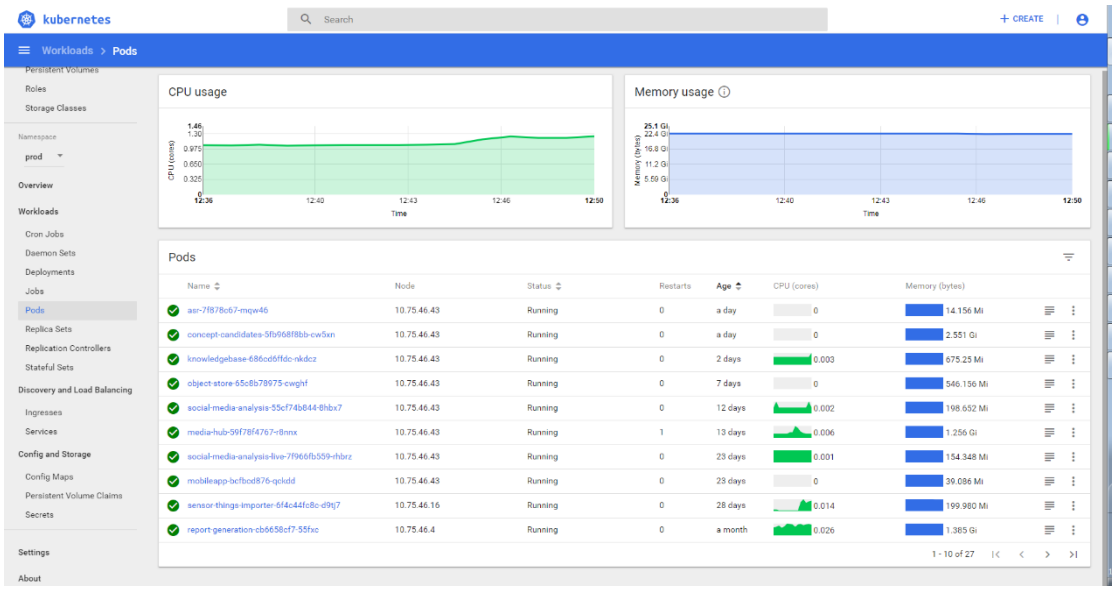


Figure 6: Kubernetes microservices view

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 3.1.4 .

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 in order to support the visualisation tools at the beAWARE's dashboard.

Briefly, the functionalities of the Crisis Classification module that have been developed and integrated during the 2nd phase of development are the following:

a) In the framework of *Early Warning* component, the estimation of the Crisis Level has been enriched by the assessment of the severity of the forthcoming crisis in local level, apart from the global one (in whole Region of Interest). Thus, in the flood pilot, the river sections are divided into 6 groups and for each group the Predicted Crisis Level is estimated. Similar, in heatwave pilot 6 different locations with various climate characteristics in the district of Thessaloniki are chosen and 7 places in Valencia for the fire use case.
   Furthermore, the mechanism to integrate Flood Hazard maps and Risk/Impact maps are implemented. During the flood scenario, those maps are created and

provided by the AAWA in the shapefile format, which is a digital vector storage format for storing geometric location and associated attribute information. These data are stored in the shapefiles as primitive geometric shapes like points, lines, and polygons. The *Early Warning* component extracts the most significant polygons from those maps. Each polygon is related with attributes such as the risk level (scale from 0 to 1), the class of risk (moderate/low, medium, high, very high), the flood intensity/hazard (from 0 to 1) as well as the estimated damage (scale from 0 to 1) as a function of the vulnerability and exposure indicators. These details along with the geographical coordinates (latitude/longitude) of the nodes of the polygons are forwarded to PSAP in order to present into the map. Using the similar mechanism, it is possible to integrate into the beAWARE platform risk maps originated from the European Flood Awareness System (EFAS) portal, as well as maps which illustrate inflammable areas, fuel model maps, fire danger maps from European Forest Fire Information System (EFFIS) portal for the Fire pilot.

b) In the framework of *Real-Time Monitoring and Risk Assessment* component, the risk assessment process during the crisis has been enhanced. Currently, an innovative algorithm to estimate the risk/severity of the ongoing flood, that relies on the exploitation of the local information coming from the citizens' and first responders' mobile application via the appropriate incident reports, is developed. During the 2nd pilot, this algorithm has been evaluated under realistic conditions and its performance has been measured in terms of its precision.

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	Forecasts: hourly data for 55h ahead
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 concept and conceptual relation extraction component implement the multilingual text analysis functionalities of beAWARE, enabling to process textual inputs in the targeted languages (English, Greek, Italian and Spanish) and abstract them into structured, semantic representations that capture their meaning, and can be subsequently reasoned upon for intelligent decision making.

For the second pilot, UPF has extended the first version of the text analysis module by using a wide-coverage text analysis pipeline capable of producing analysis for texts beyond those scripted for the pilots. In addition to some components already present -i.e. surface and deep syntactic parsing, new components have been added for NER, concept extraction, EL and geolocation. NER is addressed using Stanford CoreNLP, while UPF own solutions and models are used for deep syntactic parsing, concept extraction, entity linking and geolocation. The last three are being developed within the scope of beAWARE. In addition, a retokenization and a relation extraction module have also been developed to assist in the creations of connected semantic and ontological representations.

NER and concept extraction produce annotations of single words or multiple consecutive words that express relevant concepts and entities respectively. Entity linking produces disambiguated references to BabelNet and the geolocation component references to two geographical databases, Open Street Maps and GeoNames. The purpose of these tools is to facilitate the detection of contents that

can be mapped to the project ontologies, as well as extracting useful information such as the geographical coordinates of locations mentioned in the texts. In this deliverable we report separate evaluations of our concept extraction and EL components. Geolocation is still in first stage of development and will be evaluated for the third and final prototype.

In addition of deploying the new components, efforts have been placed on integrating their outputs into a single linguistic structure. This structure is a semantic graph, one per sentence in the input text, where nodes correspond to text fragments indicating concepts or entities detected by the NER and concept extraction components, and edges are deep syntactic relations produced by the deep parser. Nodes can be associated with references to BabelNet, Open Street Maps, or GeoNames produced by the entity linking and geolocation components. Creating this semantic structure is addressed by a retokenization module that, before running the deep parsing component, merges multiword annotations into a single token and resolves conflicts between overlapping annotations produced by the other components.

The resulting semantic graphs are then used as the basis for a simple relation extraction strategy that maps words of BabelNet synsets to classes in the ontology and connects them with ontology properties (i.e. relations) if they are connected in the graph. A detailed description of each one of the components as well as the intermediate and final representations will be given in the upcoming D3.4. In this deliverable we provide a first qualitative evaluation of the whole text analysis module based on the textual inputs used in the scope of the 2nd pilot.

The following tables describe the performance indicators used for the quantitative evaluations of the entity linking and concept extraction modules, which match exactly those described in D1.1 and D3.1:

Performance Indicator	Precision and recall of extracted concepts
Definition	These metrics compare the concepts automatically detected by the concept extraction component against a manually annotated gold-standard.
Domain	Concepts detected on textual inputs
Range	The values of these metrics are between 0 and 1.0.
Limitations	These metrics outline errors in the delineation of concepts boundaries but cannot indicate the type and thus the severity of such errors. In addition, these metrics cannot

	capture the implications of inter-annotator agreement (Cohen's kappa coefficient) in the attained upper bound performances.

Performance Indicator	Precision and Recall of disambiguated concepts
Definition	These metrics compare 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 these metrics are between 0 and 1.0.
Limitations	These metrics indicate erroneous sense assignments but cannot assess the semantic distance between the assigned and expected sense.

2.3.7 **Automatic Speech Recognition**

The automatic speech recognition module provides a channel for the analysis of spoken language flowing into the system as audio recordings from citizens and first responders, either through the Mobile App or as emergency calls to a dedicated call center. The purpose of this module is to transcribe speech in four languages (English, Spanish, Italian, Greek). The transcribed text is sent afterwards to the text analysis (MTA) module for semantic extraction. Additionally, during the development phase of the second prototype, a call-center solution was included in the platform, in order to receive emergency phone calls, and a relevant module was developed, able to fetch recorded calls and voice messages and forward them to ASR. During the call, the caller is able to determine his/her language, through an Interactive Voice Response, in order for the call to be forwarded to the corresponding ASR language model. With respect to the process of evaluating the performance of the module, 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. The general objective of those is concept extraction from visual content (images/videos). So far, the following modules have been developed and integrated to fulfil the objective of both components (for a comparison between 1st and 2nd prototype functionalities please refer to D7.5):

- Emergency classification, so as to determine which images/videos contain an emergent event or not (i.e. a fire of flood event). In parallel, this module provides the system with the capability of recognizing and discarding irrelevant visual content. Therefore, it also acts as the initial in-component (internal) validation mechanism for the visual analysis components.

- Object Detection and Tracking, so as to find people and vehicles that exist in impacted locations like parks, streets and highways.
- 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. The module is currently integrated for a static camera in the center of Vicenza, near Ponte degli Angeli (Bacchiglione river), for which three alarm thresholds have been defined by AWAA. In order to evaluate this module, percentage error will be used.
- Drones Analysis tool for analysing drone footages with the aim of detecting people and vehicles in danger.
- Sensitive content blurring, so as to protect the privacy of targets inside the visualized images/videos on the platform.

The following tables include a description of the main properties (domain, range, and requirements) of each performance metric that will be used to evaluate the VISUAL ANALYSIS components. The following evaluation criteria are designed to be in line with deliverables D1.1 and D1.3 which initially set the quality assurance and self-assessment plans.

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	Mean Average Precision (mAP)
Domain	Object Detection, Face Detection

Definition	To calculate the Average Precision (AP), for a specific object class the precision-recall curve is computed by varying the model overlap threshold that determines what is counted as a model-predicted positive detection of the class. The mAP score is calculated by taking the mean AP over all classes.
Range	The values of this metric are between 0 and 1.0. Higher is better.
Requirements	To perform this evaluation, annotated bounding boxes must exist or be prepared for a specific set.

Performance Indicator	Overlap Precision and Recall, F1-Score
Domain	Dynamic Texture Localization
Definition	Overlapping precision measures, the size of the True Positives to the total number of detected textures. Overlapping recall measures the size of True Positives to the total number of the correct annotated data. Both of them require an overlap threshold that will define the minimum size of overlapping spatio-temporal regions between the detected and groundtruth sequences. Overlapping precision and recall can indicate the size of false alarms (i.e. precision) and missing (i.e. recall) pixels or spatio-temporal intervals (voxels) of an event detection algorithm such as dynamic texture localization. F1 is an overall measure of a model's accuracy that combines precision and recall.
Range	The values of this metric are between 0 and 1.0. Higher is better.
Requirements	To perform this evaluation, annotated masks of the groundtruth regions must exist or be prepared for a specific set.

Performance Indicator	Frames per second (fps)
Domain	Image and Video Processing
Definition	The fps metric can tell how many images or video frames per second can be processed by some processing pipeline.

	Thus, this metric is a measure of speed and responsiveness.
Range	All positive numbers. Higher means faster.
Requirements	None

The above criteria are qualitative measures and many of the performance metrics have been measured as part of various beAWARE publications that have been presented in international conferences, workshops and challenges and especially for the VISUAL ANALYSIS tasks some of these can be found also in D3.3. Besides technically evaluating the VISUAL ANALYSIS modules with quantitative results, qualitative examples are going to be presented as well. Moreover, after the completion of the 1st and 2nd pilots the components have been exposed to realistic scenario conditions and their performance will be measured separately for those events based on the data that were communicated to the system during the pilots.

Last, to evaluate is the newly integrated module of Visual River Sensing (VRS), which 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. In order to evaluate the accuracy of estimation, an annotated visual dataset will be used, which contains measurements from water level sensors and as performance indicator we will use percentage error.

Performance Indicator	Percentage Error
Definition	If the true value of a quantity (water level) is defined as X and the estimated value is Xo. Then the relative error is defined as: PE=100*(Xo-X)/X
Domain	Measurements and Error Analysis
Range	The values of this metric are between 0 and 100.
Limitations	The true value of water level is needed for the computation of PE. The footage that was used for evaluation, is from a past flooding event, for which, there are available water level measurements form sensors. The only limitations are posed by the video resolution, which is 640x340. This results in a minimum estimation resolution around 0.04m.

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. It is based on the beAWARE ontology which uses a well-defined formalism.

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.

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 Vicenza 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. For our evaluation, WG has been deployed 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 Vicenza 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. For our evaluation, WG has been deployed 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 Vicenza 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. For our evaluation, WG has been deployed on a Virtual Machine with 5GB of RAM, 4-core CPU and an SSD.

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 and have been introduced in the 2nd version of the beAWARE platform. Risk Maps can be displayed in the KBs UI and they are also used by the crisis classification module to assess the severity of an incident report. To allow a seamless integration, the risk maps are offered through a standardized interface (in this case Web Map Service (WMS)). To implement this interface, a GeoServer (http://geoserver.org/) instance was introduced in the 2nd prototype to store the risk maps and to offer the needed interfaces for the other components.

Next to time-series-data (stored in the FROST-Server), multimedia files (stored in the object store) or semantic data (stored in the knowledge base) GIS data is available. This GIS files are stored in a GIS-Server, called GeoServer[2], which is specialized for

[2] http://geoserver.org/

hosting geospatial data. It offers standardized interfaces (like s Web Feature Service (WFS), Web Map Service (WMS), and Web Coverage Service (WCS), which are used by the crisis classification module to access the available data.

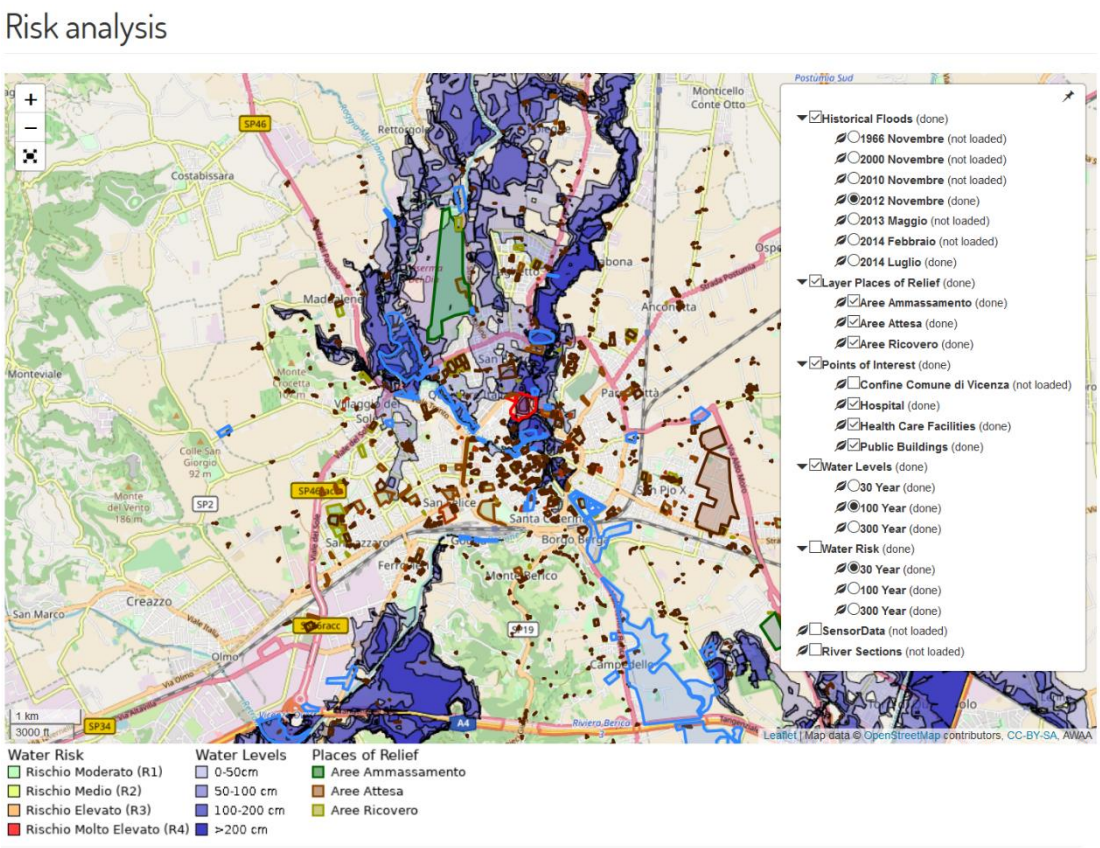


Figure 7: Visualizing the available GIS data.

In order to evaluate the overhead in the time complexity of the usage of risk maps in the whole risk assessment process, we estimate separately the execution time in both crisis phases (pre-emergency and emergency). The results are illustrated in the Subsection **Error! Reference source not found.**

2.3.10 **Multilingual Report Generator**

Starting from contents in the knowledge base, the multilingual report generation modules produces multilingual texts providing to the users of the platform with relevant information about an emergency.

The module used in the first prototype produced multilingual reports providing situational updates to authorities. These reports verbalized into one or a few sentences recent incidents detected by the system along with a description of the impacted objects. For the second prototype new functionality has been added to address the production of wrap-up summaries at the end of a crisis scenario. These

summaries, addressed to authorities, are longer than situational reports and cover the main incidents as the emergency unfolded. Summaries are organized chronologically into separate sections that correspond to one-hour time periods. Within each section, the system produces an account of the incidents detected during that time. Linguistic aggregation methods are used to reduce repetition and produce a more concise and fluent description. Thus, incident descriptions are grouped by common traits, i.e. event type, type of impacted of object or location, and a single mention is produced to the common trait instead of repeating it for each incident.

Work for the 2nd prototype has largely focused on methods for the surface multilingual generation from ontological representations. For this reason, we report a quantitative evaluation of the surface generation component of the module.

Below is the description performance indicator used in the evaluation of the component, which is just one of the indicators listed in D1.1. Additional indicators from the initial set will be added in future evaluations.

Performance Indicator	BLEU
Definition	N-gram-based comparison of a system output against a (set of) reference manually crafted output(s).
Domain	Reports in each of the languages supported in the beAWARE platform.
Range	From 0 to 1.0.
Limitations	This metric only calculates the similarity of word sequences between two texts. It does not account for the linguistic quality of the generated sentences.

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 component was not foreseen in the original proposal and was added for the second prototype. The essence of the drones platform capabilities is the combination of route planning and autonomous dynamic piloting, with the provisioning of data collected by the drone making it available to corresponding beAWARE analysis components.

The drones platform consists of a mobile device which connects to the drone remote control, controlling the drone operation using a programmatic interface. The specific

selection and configuration of a service is performed using the mobile device, and imagery from the drone can be viewed on the device. In addition, there is a server component on which the services are hosted and run. For example, it calculates the flight route and communicates with the mobile device to control the drone; and is responsible for distributing the drone data to the selected destinations. A management dashboard component complements the platform, via which the current route can be seen, as well as the imagery captured by the drone.

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.

In the 2nd prototype, the data visualization and validation mechanism was expanded to provide more accurate details to the users. In the incident map, more attributes

are associated with displayed icons providing more detailed information to the user about the received incidents. Incident manager was improved to allow editing data by the operators, increase reliability, change priory and more. Clustering mechanism of the incidents was also improved including raw and analysed media like images, videos, and audio, stacking of multiple media files from different incident updates. It is worth also to be mentioned that additional message topics were deployed to accommodate new type of information received such as Metric Reports, risk map polygons, summary reports etc. Finally, In the second prototype, it was introduced the first version of the Operation Manager module, to handle incidents and tasks, and monitoring their progress.

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

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 visualisation of different information layers gathered from disparate data sources
Domain	Visualisation and interaction
Range	5-point Likert scale.
Limitations	Each report should be assessed by multiple UI elements

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 team- and task management functionality. First responders can report their position and status to the PSAP. It is also possible to receive tasks from the PSAP and report the status. The incident report mechanism was extended enable the selection of report categories, which is evaluated by the crisis classification component.

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 visualisation of different information layers gathered from disparate data sources
Domain	Visualisation 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.

3.1.1 Social Media Monitoring

The main enhancement of the Social Media Analysis (SMA) module in the second prototype is the integration of the fake tweets' detection as part of a three-step validation of the social media. In order to prove its value, we compare the classification results of the relevancy estimation based on text, with (second prototype) and without (first prototype) fake tweets detection. For the evaluation, a dataset of one thousand human-annotated Italian tweets about floods is selected, while the compared text classification technique uses the Term Frequency Inversed Document Frequency (TFIDF) text representation without stemming and a Random Forest classifier. As seen in Figure 8, the precision of the text classification method without the fake tweets' detection is 84%, recall is 89%, and F-score 87%. The integration of the fake tweets' detection raises precision to 96% and, consequently, F-score to 93%, clearly demonstrating the need to filter out tweets which are not real.

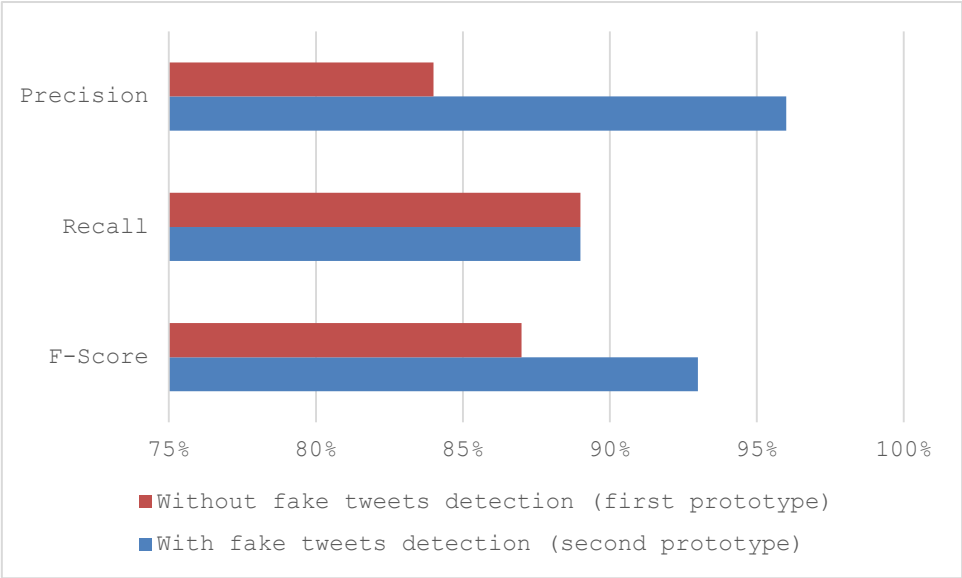


Figure 8: Comparison of text classification results with and without fake tweets detection.

Regarding the second beAWARE pilot, thirteen messages about social media were exchanged in the system flow and five Twitter Reports were created. In order to evaluate the process time of the SMA module, we calculate the time difference between the creation of a post on the Twitter platform and the creation of the

respective message in the beAWARE system (after crawling, fake tweets detection, emoticons check, and relevancy estimation). The average process time of the thirteen posts during the pilot was 5.77 seconds, manifesting how fast a public post can be digested into the beAWARE platform.

3.1.2 **FROST-Server**

The FROST-Server was used as central point for storing the time-series data. All available sensor sources, including weather stations, water level sensor, water level predictions and weather forecasts have been included. The data is imported periodically. The import interval is specifically selected for each source according to the interval new data is produces. By this, the data is available as soon as possible, which is suitable for the use-cases. Therefore, the FROST-Server fulfils the highest expectation defined in the evaluation methodology.

The heterogeneous data has been integrated in a unified way. All available time series data is stored in the FROST-Server and can be access by the SensorThingsAPI standard. This can be seen in the interactive map which provides the entry point for the user to all available time series data. The correctness has been proved by manually validating the results with the expert knowledge of the use-case and the current situation.

The pilot execution showed that the FROST-Server is able to handle all available data as soon as it is available. Therefore, the scalability performance indicator was fulfilled. In the background, the data is stored in a state of the art relational database, called PostgreSQL. The correctness of the written data is guaranteed by the database, due to the fulfilment of the ACID properties (Haerder & Reuter, 1983). In addition, all test before the pilot and during its execution showed the correctness of the written data. Since all available data sources have been integrated, the highest expectations for the integration indicator have been met.

3.1.3 **Communication Bus**

The main purpose of this component is to provide generic communication capabilities among different beAWARE components and participants. It can be used to send messages and notification among components and to share information among various entities. In a microservices based architecture, such as beAWARE has adopted, there is a need for communication among different microservices, and one of the dominant manners to achieve that is by using a distributed publish / subscribe mechanism. The beAWARE used Communication Bus provides the ability for different entities to send and receive messages without having to be aware

specifically of each other. The agreed upon pieces of information to enable such an integration are the topic name which shall be used for a specific kind of interaction, and the message format, such that different entities will be able to understand each other. Extensive work has been done in beAWARE to reach an agreed upon list of topics and their corresponding formats (which are specializations of a generic message format including a header and a message body).

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

The communication bus is configured, upon deployment, with the necessary set of topics as agreed upon between the different components. In addition, the message structure of each message in each topic is agreed upon and documented by the cooperating components. The communication bus supports the number of different topics required for a beAWARE installation, along with the associated aggregated throughput in all topics. That assertion was validated in the 2 first pilots, in which multiple users interacted with the platform successfully. Moreover, in the second pilot we introduced the drone component which exercises both the object store and the messagehub heavily, by sending one image per second over a period of approximately 10 minutes per flight session. BeAWARE experienced no problems coping with the required throughput exhibiting a reasonable latency. A representative session included the ingestion by the platform of 612 images (1 image per second), for a total of 78.4 MB (an average of 128KB per image)

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

The communication bus has been deployed as a central component of the beAWARE platform for over 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 second pilot workloads (including messages from drones (in red) which account for

the highest traffic, and metric reports (in blue coming from devices) which are the second largest contributors:

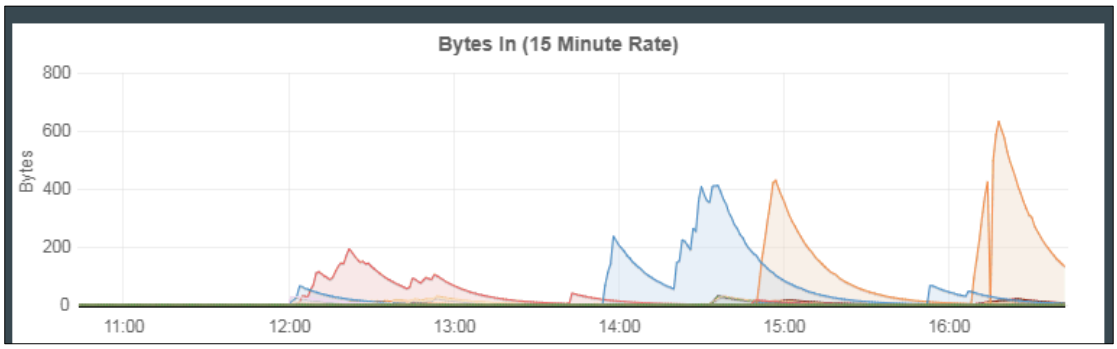


Figure 9: Message Bus usage statistics 1

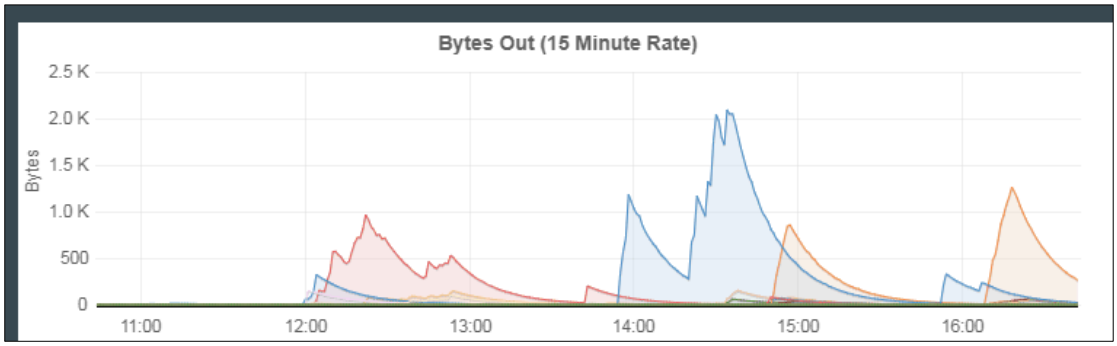


Figure 10: Message Bus usage statistics 2

Usage for all buckets in: ams03

Storage Class	Standard	Vault	Cold Vault	Flex	Total Apr
Monthly average capacity	4.0 GB	0 bytes	0 bytes	0 bytes	4.0 GB
Public standard egress	613.5 MB	0 bytes	0 bytes	0 bytes	613.5 MB
Class A (request count)	4982	0	0	0	4982
Class B (request count)	2425	0	0	0	2425
Data retrieval	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes

Figure 11: Object Store statistics

Scalability and performance measures

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

affecting performance, that can be applied when the system load gets considerably larger.

1. Number of servers / brokers - For scalability and fault tolerance the communication bus can run with a number of servers acting as cooperating message brokers; cooperating for providing continuous service. Running multiple brokers means that for each partition there shall be a single broker acting as the designated leader, and a list of brokers acting as replicas. Currently beAWARE's communication bus is deployed over 5 brokers. The number of brokers can be scaled up based on need, but for the foreseeable future there is no expectation that the platform would require more brokers to be deployed. Replication factor for beAWARE's topic is 3, thus we ensure that sent messages are available in at least 3 brokers, such that the platform can continue normal operations even in the unlikely event of two brokers being unavailable simultaneously.
2. Number of topics - The entire messages space sent and received by the communication bus is divided into topics. Each topic forms a separate unit to which messages can be sent and through which messages can be consumed. Message publishers designate every message they send to a single topic, and message consumers declare a set of topics which are of interest to it. In such a manner the entire spectrum can be divided between different processes distributed over different nodes, and have the overall load to be distributed between different clients and different broker entities. Currently in the communication bus there are 28 topics declared and used operationally.
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; messages order is guaranteed within a partition. Clients consume messages from the partition head while messages are added to the tail. The partition is also the unit of division between different brokers. Thus, each partition is owned by a specific broker / server. The number of partitions of a topic can be scaled up and down based on need. Messages are kept in memory, thus dividing a topic to multiple partitions enables handling large topics regardless of the amount of memory in a single server. BeAWARE uses a single partition per topic.
4. Consumer groups - Consumer groups enhance the scalability of the messaging system, by declaring a group of cooperating consumers, and having the system ensure that each message will reach one member of

each consumer group. Within a group each member is assigned a fair share of partitions to receive messages from. The combined features of partitions and consumer groups contributes to the overall system scalability and load balancing. BeAWARE places each subscriber in its own consumer group.

5. During the flood pilot the heaviest user of the message bus was the drone platform. In every session (flight) the drone sent one message per second (exercising heavily also the object store which received an upload request from the drone every second, and a corresponding download from the image analysis component every second). Total amount of messages per session amounted to 612.

3.1.4 Technical Infrastructure

The supporting cloud services consists of several data stores of several flavors and the message bus which serves as the method of interaction between the different components.

The message bus currently supports 28 topics which are used for different microservices to communicate among themselves, as can be seen in Figure 12.

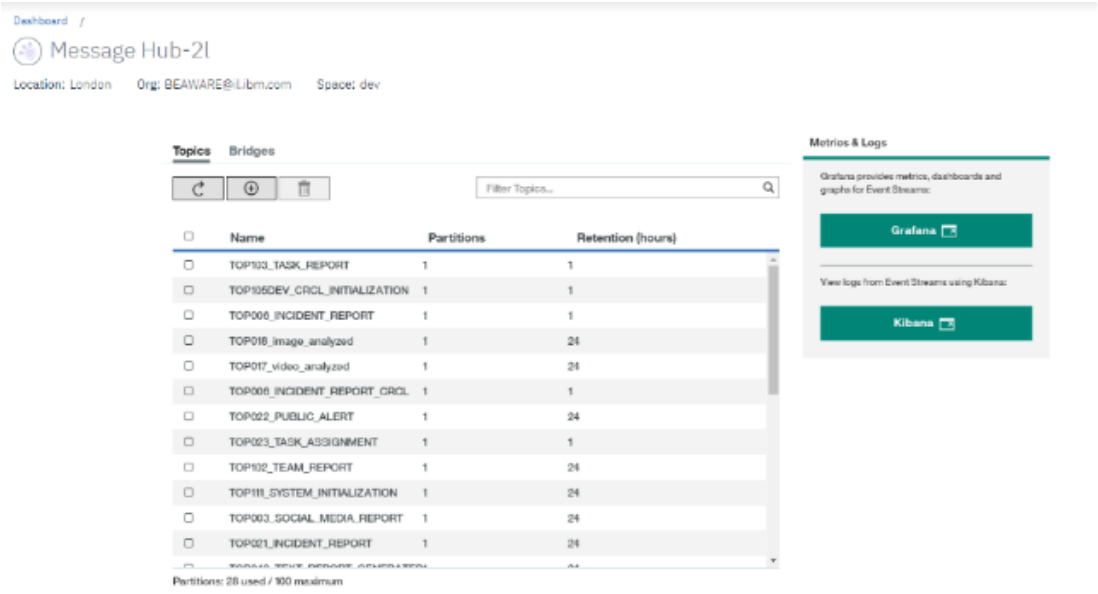
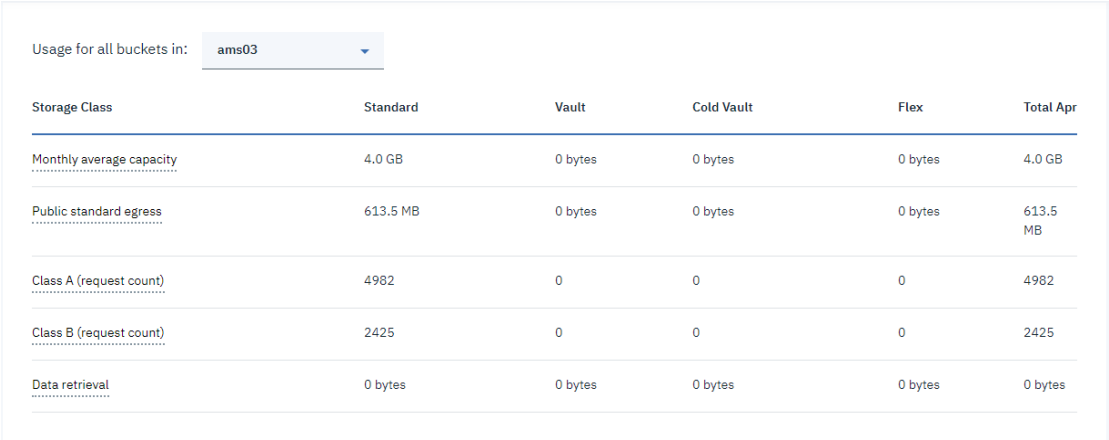


Figure 12: Message Bus

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) – see



Usage for all buckets in: ams03

Storage Class	Standard	Vault	Cold Vault	Flex	Total Apr
Monthly average capacity	4.0 GB	0 bytes	0 bytes	0 bytes	4.0 GB
Public standard egress	613.5 MB	0 bytes	0 bytes	0 bytes	613.5 MB
Class A (request count)	4982	0	0	0	4982
Class B (request count)	2425	0	0	0	2425
Data retrieval	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes

Figure 11. During the flood pilot the heaviest user of the object store was the drone platform. In every session (flight) the drone uploading an image every second, and a corresponding download from the image analysis component every second). Total amount of files per session amounted to 612, for a total size of 78.4 MB.

We used a re-run of the drones flight scenario to measure and evaluate the infrastructure used. As a reminder, the session is comprised of sending an image per second to be stored on the object storage followed by sending a message including the metadata over the message bus.

Results of sending an image to the Object store including the actual storage and the receipt of a notification takes on average 660 ms (with a standard deviation of 102 ms).

Results of sending a message with metadata to Message Hub and the receipt of a notification takes on average 200 ms with a standard deviation of 94 ms

The corresponding network conditions in which the experiments took place: round trip latency from the test machine to the Object Store took an average of 69.2 ms. An estimate of the round-trip latency from the test machine to the Message hub was 56 ms. Note that mostly the network conditions in on field tests, using mobile networks) are not as good as the connection used for these tests.

An instance of Mongo DB, can be seen in Figure 13, is used mainly by the social media component.

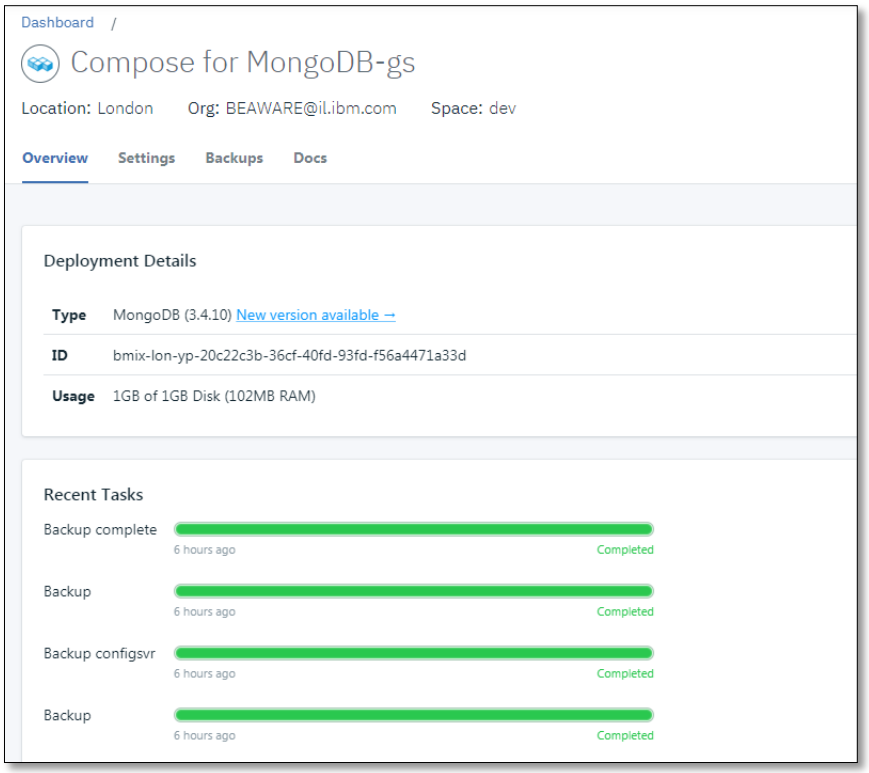


Figure 13: MongoDB instance

Finally, an instance of MySQL, Figure 14 which is used by the KB.

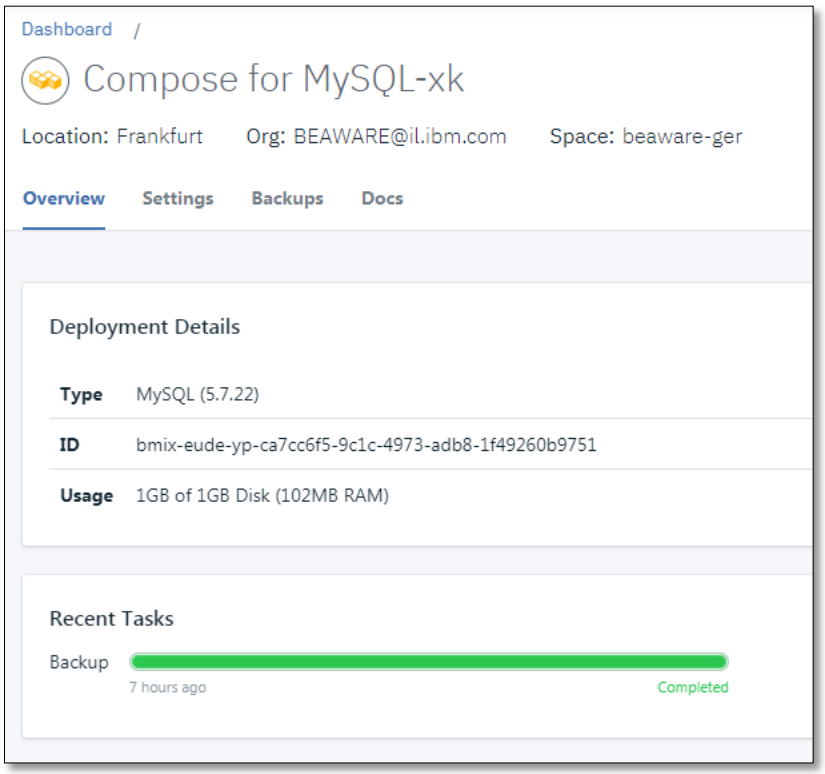


Figure 14: MySQL instance

3.1.5 Crisis classification

In this section, the evaluation process of the Crisis Classification component is carried out related with the 2nd prototype of the beAWARE system. The goal is to estimate the performance of each one of the Crisis Classification components, namely the *Early Warning* and the *Real-Time Monitoring and Risk Assessment* component, 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. It is worth to note that the execution time is carried out over the distinct processes within each component.

As the obtained data are quite different for each Use Case (flood, fire and heatwave) it is better to assess the performance of each component for each UC separately.

Crisis Classification evaluation for the Flood pilot

Briefly, the *Early Warning* component includes the following steps:

Step 1. Data Acquisition: includes the processes to grab stream of data (AMICO forecasts) from various prediction models and sources.

Step 2. Data Analysis: includes the processes:

a. to estimate the level of crisis based on water level forecasts and create the appropriate messages to forward to PSAP,
b. to search the provided risk maps and estimate the risk of flood in specific areas (polygons) near by the river section. Also, it creates the appropriate messages to forward to PSAP,
c. to estimate the crisis level in each river reach and the overall flood crisis level and create the appropriate messages to forward to PSAP.

In the flood use case, the data in pre-emergency phase is the output of the hydrological and hydraulic model, named AMICO, provided by AAWA. The AMICO provides hourly estimations of the river water level over specific river sections in forecasting period 55 hours ahead. From the total 304 river sections Early Warning component obtains forecasts and analyse the 60 most significant river sections, which have indicated by the experts (AAWA team). Moreover, these 60 river sections have been clustered to 6 groups. In each group, one river section is considered as *Critical*. The distribution of the river sections per group is illustrated in the following figure.

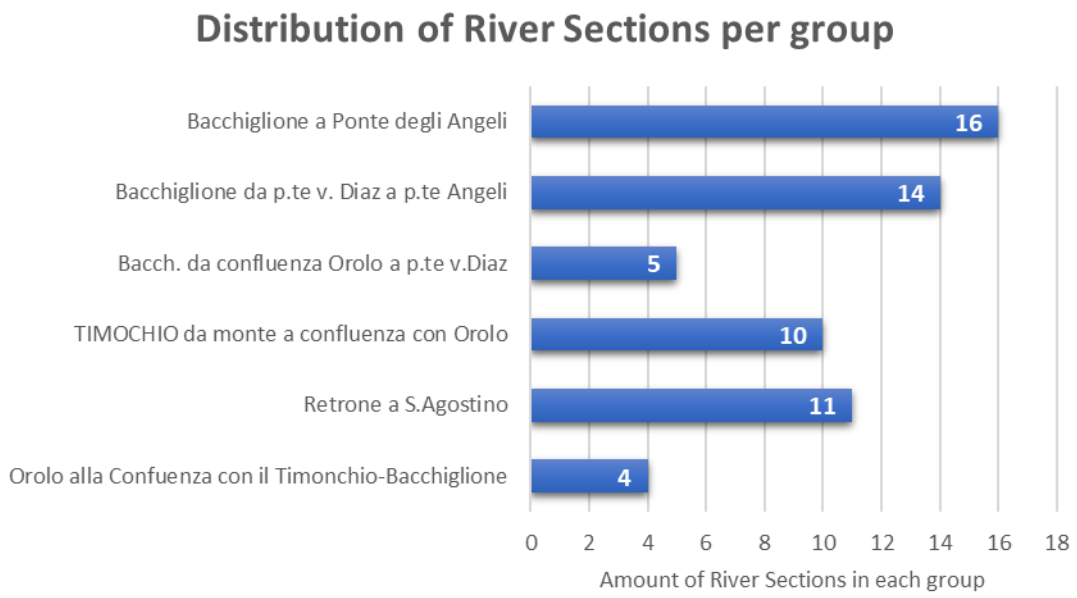


Figure 15: Distribution of river sections per group

This series of experiments focuses to evaluate the execution time in each step of the Early Warning component in the flood pilot. For this reason, we employ various type of real and simulated datasets. Specifically, the AMICO forecasts from the flood crisis event in the period between 31-10-2010 to 03-11-2010 is employed as real case dataset, named "Real Flood Data (2010)". The forecasts are stored to the FROST-Server and extracted from there via appropriate queries. Also, three simulated datasets are utilised which corresponds to different executions of AMICO algorithm with various initial conditions. These datasets are stored and retrieved from the local database. The number of forecasts that Early Warning component retrieves is 3300 estimations per AMICO run. The average total execution time after 5 iterations of the Early Warning algorithm is less than 3 minutes (156.82 ± 1.70 seconds) in the worst case (Table 1 and

Figure 16).

Table 1: Early Warning execution time per step

	Real Flood Data (2010)		Simulated Data "4March"		Simulated Data "5March"		Simulated Data "6March"	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Step 1	0.91	0.20	1.02	0.02	1.04	0.04	1.07	0.00
Step 2a	101.08	5.29	83.66	1.46	94.98	1.02	94.55	0.02
Step 2b	45.67	5.20	46.58	1.36	53.78	1.02	54.43	0.01
Step 2c	5.57	0.14	6.66	0.01	6.96	0.15	6.77	0.00
Total Time	153.24	10.52	137.93	2.81	156.76	1.98	156.82	1.70

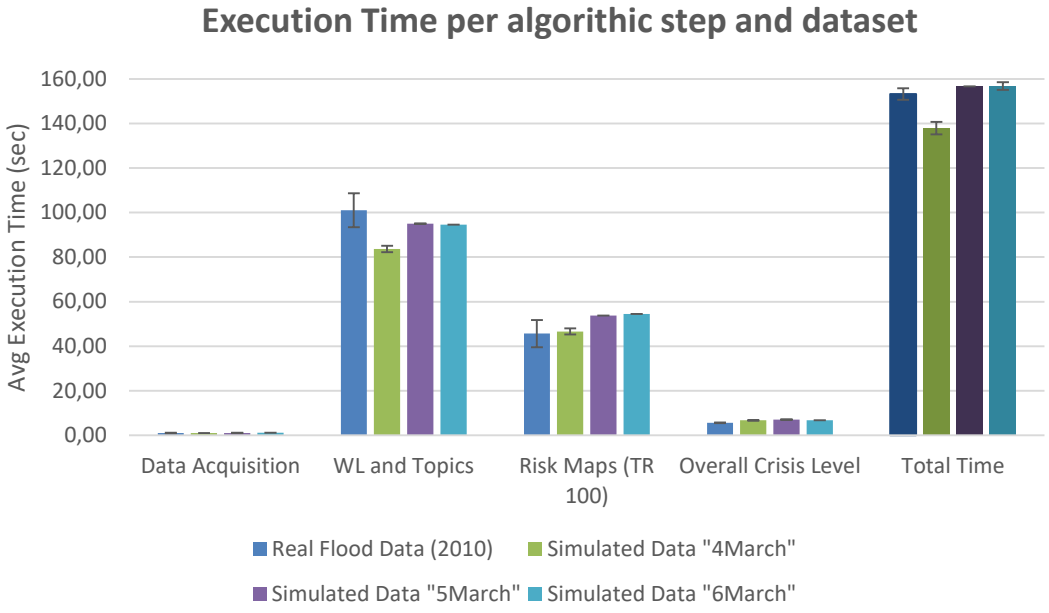


Figure 16: Early Warning execution time per step

It is worth to note that in the above experiment the return period of the risk maps is 100 years. In order to examine the effect of the return period in the execution of this component, a new set of experiments are carried out. As it is expected, there is a slightly deterioration in the time that fluctuates from 1.72 seconds in the “Simulated Data 4March” dataset to 9.90 seconds in “Real Flood Data (2010)” case (

Figure 17).

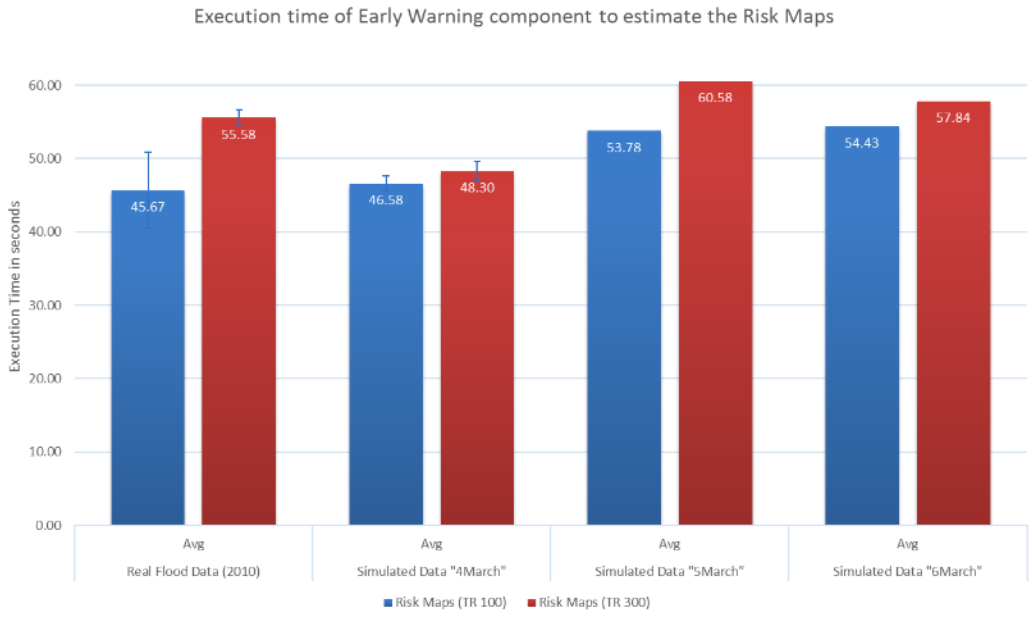


Figure 17: Execution time of Early Warning Component for risk maps estimation in return period 100 and 300 years

In the following table (Table 2) the number of metrics reports that the Early Warning component generates and forwards them to beAWARE dashboard and PSAP is presented. The metric reports for water level estimation are forward to the PSAP map and presents the river sections that exceeds one of the alarm thresholds in the forecasting period. The other two sets of metric reports (Metric Reports for Critical River Sections and Metric Reports for Overall Crisis Level) contain aggregated metrics to illustrate into the beAWARE dashboard. It is worth to note that the number of messaged for the risk maps as well as the number of unique polygons that they contain differ significantly in all the datasets between the return periods (TR100 and TR300).

Table 2: Number of metric reports which are generated by Early Warning component

	Real Flood Data (2010)		Simulated Data "4March"		Simulated Data "5March"		Simulated Data "6March"	
	TR 100	TR 300	TR 100	TR 300	TR 100	TR 300	TR 100	TR 300
Total Metrics Reports	61	61	57	57	68	68	65	65
Metric Reports for Water Level estimation	47	47	43	43	54	54	51	51
Metric Reports for Critical River Sections	6	6	6	6	6	6	6	6
Metric Reports for Overall Crisis Level	8	8	8	8	8	8	8	8
Messages for Risk Maps	38	42	35	41	42	49	41	47
Number of unique polygons	714	953	647	873	686	955	661	920

During the emergency phase, the *Real-Time Monitoring and Risk Assessment* component is activated aiming to track and inform authorities and decision makers regarding the evolution of the flood crisis event. Briefly the following steps are carried out:

Step 1. Data Acquisition: includes the processes to grab real-time observations at sensors which are located in specific weather stations. The measurements concern the status of current water level and the amount of precipitation.

Step 2. Data Analysis: includes the processes:

a. to estimate the level of crisis based on real-time observations,
b. to create the appropriate messages so as to update the map at PSAP regarding the status of observed water level and precipitation at specific weather stations and the visuals at the beAWARE dashboard,

Step 3. Forward the messages to PSAP/Dashboard.

Figure 18 presents the location and names of Weather Stations that *Real-Time Monitoring and Risk Assessment* component utilises.

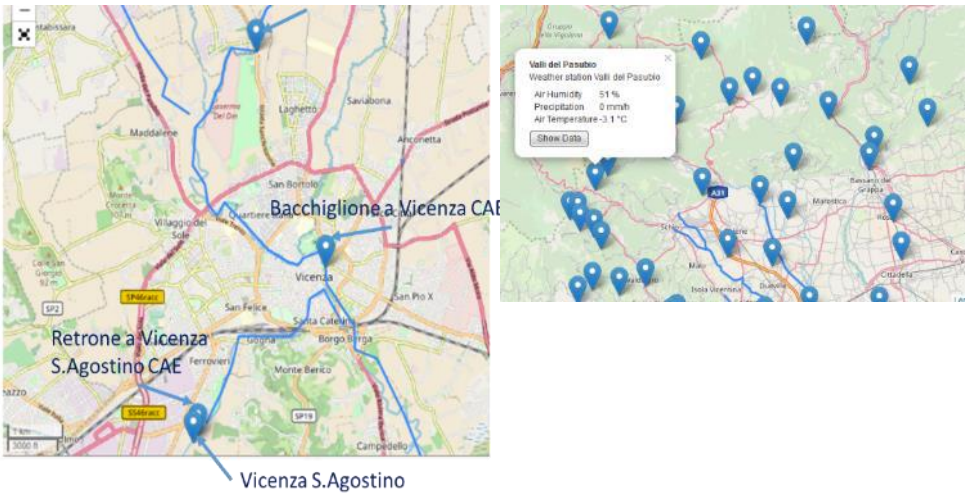


Figure 18: Weather stations in Vicenza region in which sensors for real-time observation of river water level and precipitation

Every hour the real-time observations are collected from those weather stations and analysing. The results of analysis are reported and an amount of messages proceeds to PSAP and dashboard. The number of generated messages depends on the number of observations that exceeds pre-defined thresholds. Using various datasets which are able to simulate different conditions, the *Real-Time Monitoring and Risk Assessment* component needs around 13.10 ± 2.08 seconds in average to generate around 10 messages for river water level and 11.7 messages for precipitation in those particular simulated datasets (Figure 19).

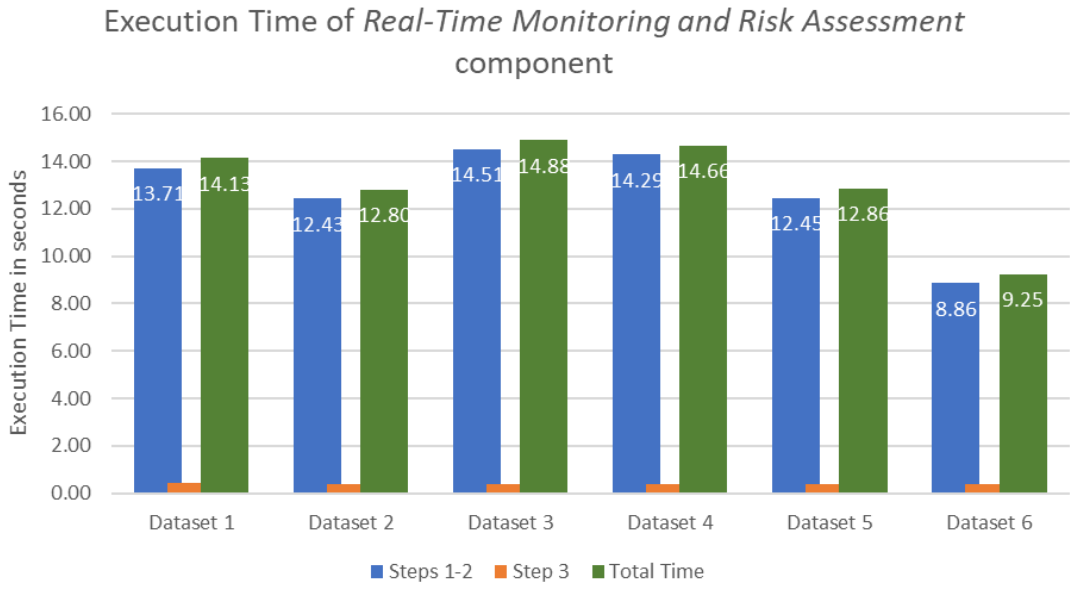


Figure 19: Execution time of *Real-Time Monitoring and Risk Assessment* component

Another functionality of *Real-Time Monitoring and Risk Assessment* component that it is worth to evaluate is the process to assess the risk of ongoing flood crisis event exploiting information from citizens' mobile application. The following algorithm use local information from citizens' perspective in order to estimate the current crisis risk based on the details provided by citizens regarding the people in danger, the contiguous buildings or other historical assets etc. Briefly the steps of the algorithm are the following:

Step 1. **Data Acquisition**: includes the processes to request the data related with the evolution of the flood crisis from citizens' mobile application via appropriate designed reports. Those data are retrieved from the beAWARE Knowledge Base Ontology.

Step 2. **Data Analysis:** estimates the risk assessment by the exploitation the receiving information from the citizens. It includes the calculations of hazard, exposure, vulnerability and finally the hydraulic risk and severity of each incident. If it is needed, the obtained information is enriched with data which are extracted from the GIS files presenting risk maps in various return time period. Those files have been stored in beAWARE's geoServer and are related with historical river water level observations, the exposure assets and their vulnerability in the Vicenza region as well as the severity level and risk estimations.

Step 3. Store the incident report and results of the analysis to the local database and **create** the appropriate messages to PSAP in order to update the status of each incident.

Step 4. Calculate the accumulated **Risk Assessment** relying on the severity of all obtained incident reports.

Step 5. Store the results to the local database and **create** the appropriate messages to Dashboard in order to update the corresponding plots.

In order to evaluate the Risk Assessment algorithm, we sent messages into the beAWARE system from various locations nearby the points of interest, such as hospital (red polygon in the following figure), other health care facilities (light brown polygons) and public buildings (brown polygons), places of Relief (green polygons) as shown in Figure 20.

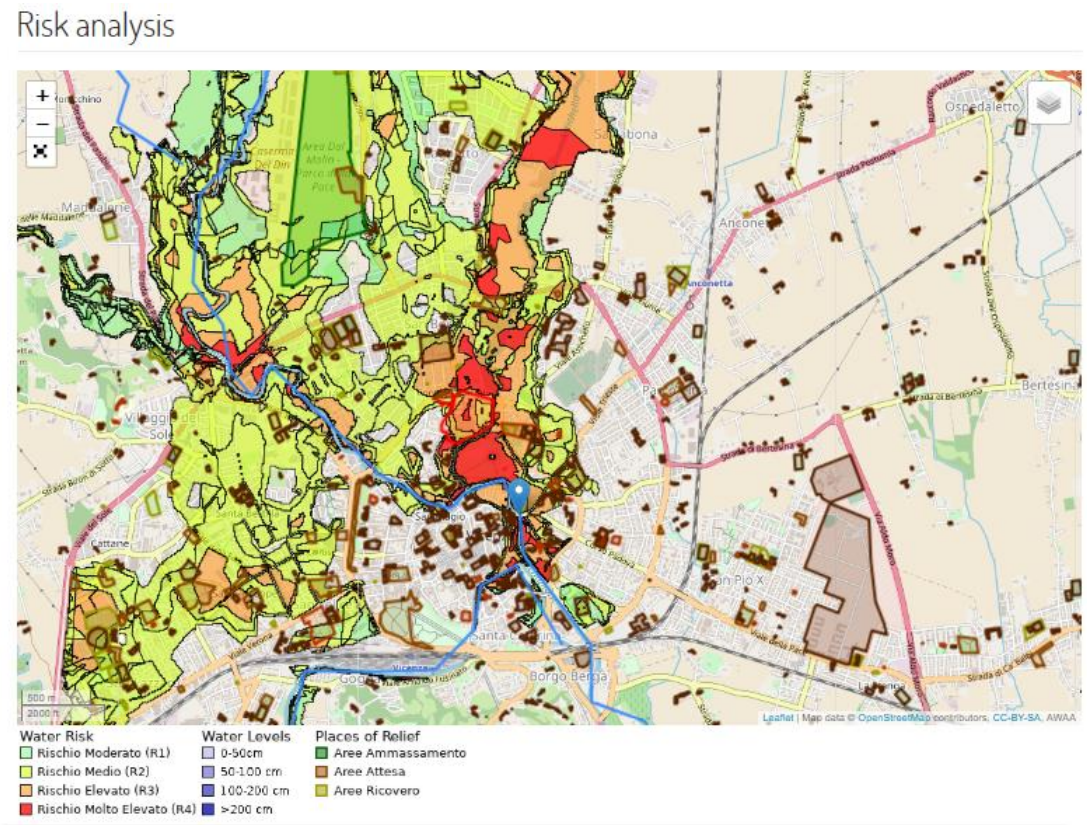


Figure 20: Points of Interest in Vicenza district [Source: https://beaware.server.de/servlet/is/696/]

The distribution of the 50 incoming incident messages on the map is presented in Figure 21. The majority of the incoming messages 36% (18 out of 50) were categorized as Severe by the Risk Assessment algorithm (Step 2), the other 26% (13 out of 50) were characterized as Extreme. Also, there exist 5 messages (10%) which were remarked as moderate by the algorithm and the rest (14, namely 28%) as minor severity.

Figure 21: Incoming incidents and their severity characterised by Risk Assessment algorithm

The most critical issue and major challenge in risk assessment algorithm is to manage to handle the incoming urgent messages from impacted citizens promptly and with highly accuracy. Thus, the execution time is a critical metric that we want to evaluate in relation with the number of incoming messages. For better estimation of the needed time to process the messages, each step of the above algorithm is measured. Particularly, the 2nd step is further elaborated into the following sub-steps:

- Step 2A - Extract data from GIS for Hazard field, if it is needed.
- Step 2B - Estimate Hazard value.
- Sep 2C - Calculate Exposure.
- Step 2D - Use GIS for exposure estimation.
- Step 2E - Calculate vulnerability.
- Step 2F - Calculate the Hydraulic Risk and Severity

As we can conclude from the following figure, the time-consuming processes are the extraction from beAWARE KB ontology the data coming from the citizen's mobile application (Step 1), which it lasts 0.33 ± 0.038 seconds in average and the steps that extract historical data from Risk Maps using the geoServer services (Step 2A and Step 2D).

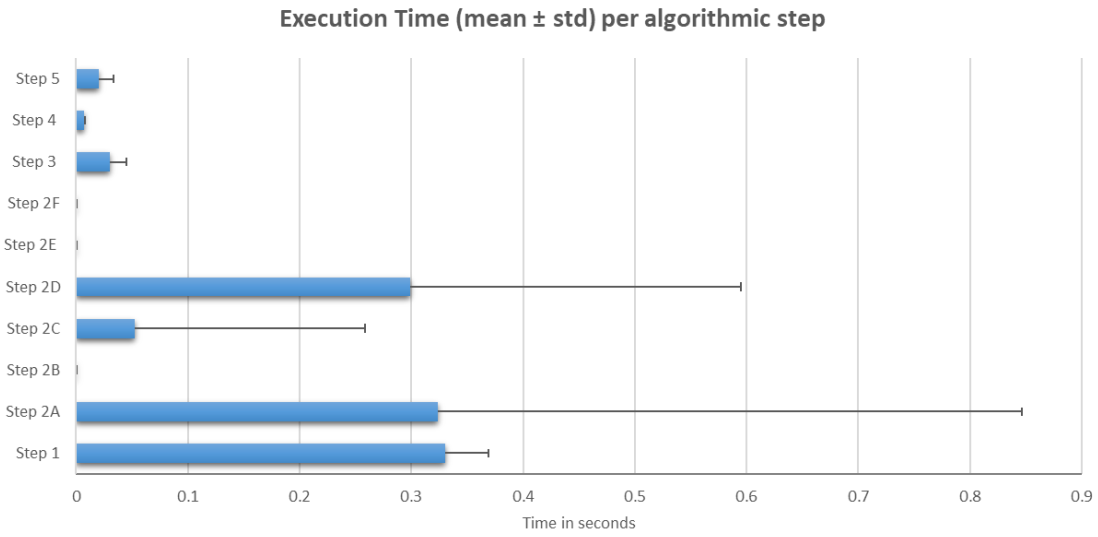


Figure 22: Average execution time per step of Risk Assessment algorithm

Indeed, as it is illustrated in

Figure 23, a significant overhead in the execution time takes place in the cases where the risk assessment algorithm needs to request the geoServer in order to extract historical data from the risk maps (green bars in

Figure 23). It is worth to note that from those 30 cases the 20 need to employ the geoServer services both to estimate the hazard value and to detect the exposure elements. However, the execution time does not exceed the 3.2 seconds.

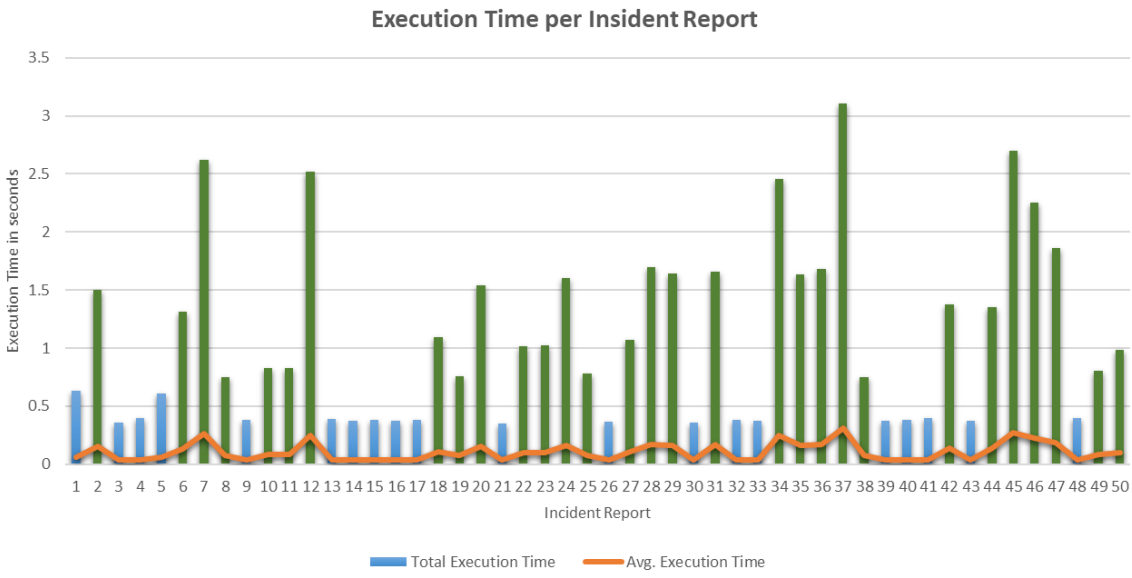


Figure 23: Execution Time (total and average) per Incident Report.

As regards the estimation of the accumulated Risk over all the incident reports which were obtained during the emergency flood crisis phase, we develop and evaluate

two approaches. The first one, which is named voting, the accumulated Risk of the Vicenza region estimated from the value of dominant category of the obtained incidents. The second one, the accumulated Risk generated by the weighted averaged mean of the severity of the obtained incidents.

The distribution of the risk assessment categories over the incoming incident reports over time is illustrated in the following figure. As we mentioned above, at the end, 13 incident reports were estimated as Extreme, 18 as Severe, 5 as Moderate and 14 as Minor.

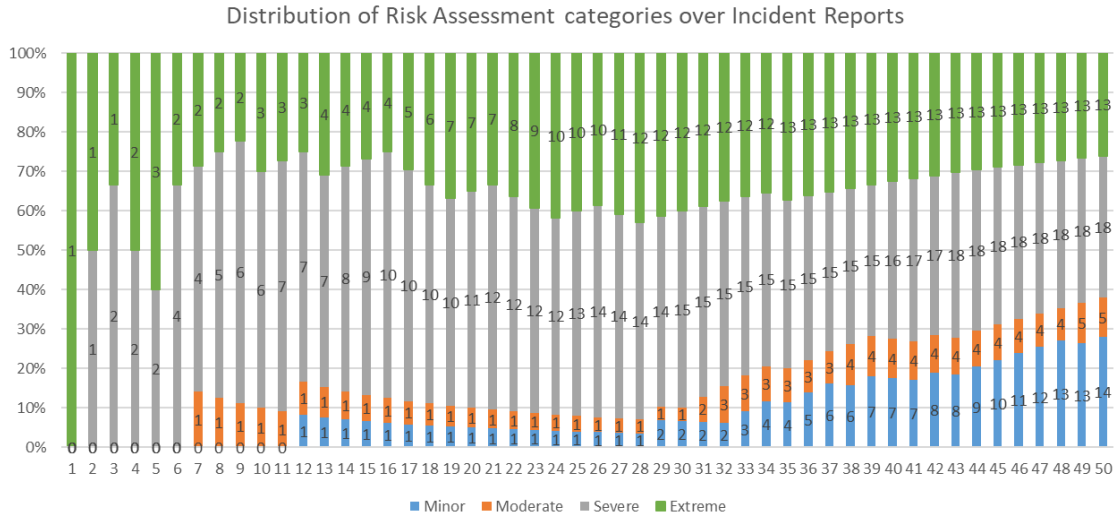


Figure 24: Distribution (in percentage) of Risk Assessment categories over incoming Incident Reports

Each time that a new incident report obtained from the Crisis Classification module, a Risk Assessment algorithm estimate and update the accumulated risk over the whole region using the Voting and Weighted Generalised Mean approaches. The

results (see

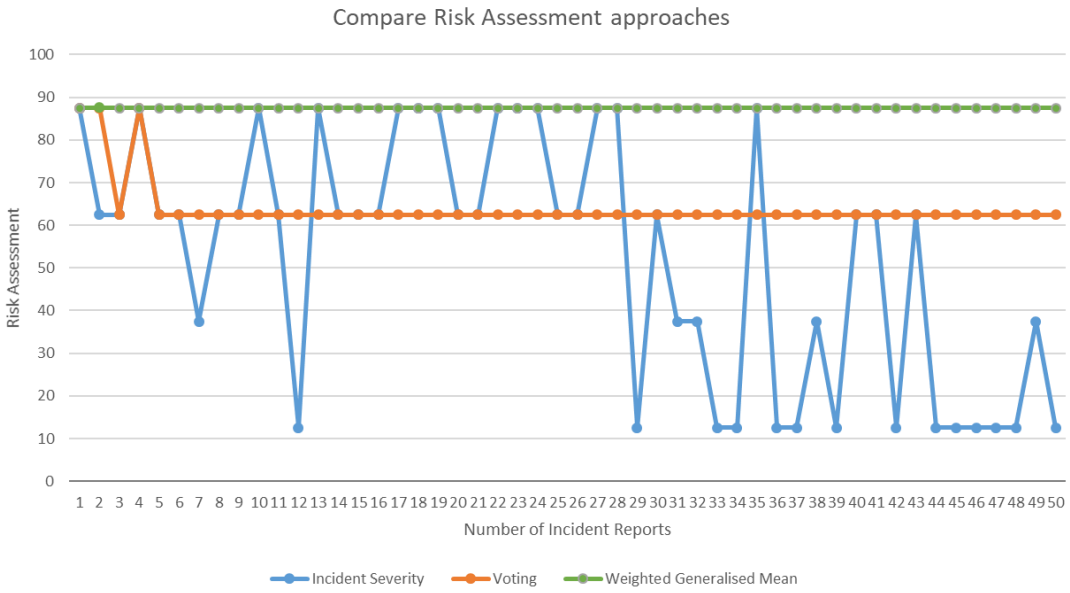


Figure 25) indicate that at the beginning when an extreme incident report arrives both methods estimate the overall crisis as 'Extreme'. Progressively, as the number of severe incidents increase, indicating that there exist a serious number of impacted citizens and the incoming information are valid, the Risk Assessment algorithm based on Weighted Generalised Mean approach keeps the accumulated Risk to 'Extreme'. On the other hand, the voting approach presents a decreasing trend in terms of the severity incoming incidents, while it initially estimates the risk of them as Extreme and then, as the number of incidents is increased, it reduces the total Risk to 'Severe'. In the following figure, with blue color is indicated the Severity of each Incident Report.

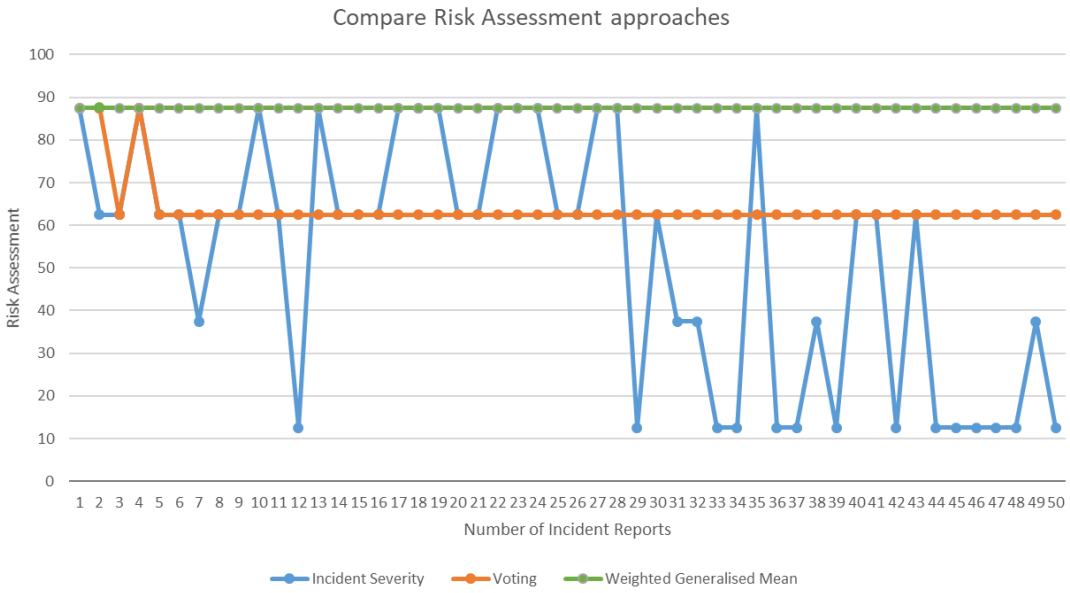


Figure 25: Compare the Risk Assessment approaches to estimate the accumulated Risk in the Vicenza region

3.1.6 **Text Analysis**

In this deliverable we present a quantitative evaluation of two components of the text analysis pipelines developed in the scope of beAWARE, namely WDS/EL and concept extraction. A third qualitative evaluation looks at the final outputs of the module. In all cases we use the set of 14 tweets used for the second written originally in Italian and then translated into English. Table 3 shows the English translations of the tweets after lexical normalization.

Table 3: English translations of tweets used in the 2nd pilot

Text 1	Water is close to embankment at Ponte degli Angeli
Text 2	Bacchiglione is going up, it's nearly up to the embankment. Better prepare sand packs
Text 3	Ponte Pusterla is full of logs and meanwhile the Bacchiglione is tall. flooding is coming
Text 4	Piazza Matteotti is flooded
Text 5	Piazza Matteotti is underwater! #flooding
Text 6	Piazza Matteotti is full of water!!!!!! a disaster!
Text 7	Piazza Matteotti is flooded, my basement is underwater!
Text 8	the afternoon continues here in Matteotti... between the flooding and the other!
Text 9	Better water situation here, but rain isn't stopping
Text 10	The road I should do to go to the office is close to Piazza Matteotti, that's why I'm at home
Text 11	Cars and dumpsters transported by the flow near Piazza Matteotti
Text 12	Surcharge of the drainage network in Piazza Matteotti
Text 13	Manhole covers in Piazza Matteotti are overflowing!!
Text 14	We are afraid. Levee near Ponte degli Angeli shows cracks and failures.

WSD/EL

The WD/EL component finds and disambiguates mentions of word senses and entities in BabelNet, a multilingual lexicographic database resulting from a mapping between multiple language versions of WordNet and Wikipedia. We have deployed a novel implementation where we treat the disambiguation task as a ranking of potential meanings. We adopt as criteria to estimate salience a similarity metric between pairs of meanings and a context-based metric for single meanings.

Similarity of meanings is estimated from the cosine distance between distributional vectors associated to meanings, or sense embeddings. We complement this similarity function with a context-based metric that we use to filter out unlikely candidate synsets and to introduce a bias in the ranking towards most likely candidates. Meaning bias is calculated from the average similarity between each

candidate meaning and a manually created set of reference meanings specific to the pilot. For the creation of the set we searched for BabelNet synsets that closely matched the top classes in the concept hierarchy of the beAWARE ontology, and also added the synset for the city of Vicenza to bias towards locations and topics closely related to the city. The resulting list of meanings is shown in Table 4.

We have experimented with various third-party sense embeddings for BabelNet. However, they tend to cover only a subset of the meanings in BabelNet, making it very difficult to compare uncovered meanings with those that do have vectors. For this reason, we have developed a similarity function that compares the glosses in BabelNet for pairs of meanings. More precisely, we use word-embeddings to calculate a BoW average for the whole text of the glosses after filtering out stop words. The resulting vectors are compared using cosine distance. This strategy and the specific resources used for the tests will be described in detail in the upcoming D3.4.

Table 4: Context meanings for beAWARE ontology classes relevant to the 2nd pilot

BabelNet sysnet	Ontology concept
bn:00002954n bn:00086542v bn:00088334v	flood
bn:00020598n bn:00083932v bn:00083969v bn:00084290v bn:00083966v	collapse
bn:00066032n bn:00092330v	rain
bn:00035294n bn:00059872n bn:00084056v	overflow
bn:00077888n	traffic
bn:00057017n	environment
bn:00008794n	infrastructure
bn:00079675n	vehicle
bn:14421293n	architectural structure
bn:12225318n	living being
bn:00134539n	Vicenza

For the purpose of evaluating the performance of this component, we manually annotated the set of tweets in English and Italian following the set of guidelines shown in Appendix 6.1 . According to the guidelines, we annotate both single words and multiple consecutive words with the ids of synsets in BabelNet that are a closest

match to their meaning in the text, allowing for multiple synsets if more than one is found adequate.

Precision and recall are then calculated by comparing the number of synset annotations produced by the system with the set manually annotated synsets. As baselines, we adopt a random strategy and BabelNet First Sense (BFS). The latter, picks the first sense returned by BabelNet, which in most cases this corresponds to the most frequent WordNet sense. We compare three versions of our component, one that disambiguates only according to the bias function (Bias), another that ranks candidates using both the bias and similarity functions (Rank), and a third one that ranks nominal expressions only and uses the BFS baseline for the rest (Rank+BFS). Table 6 and Table 5 show the results of our evaluation using precision (P), recall (R) and F-score (F1) metrics for English and Italian tweets respectively.

Table 5: Evaluation results for the Italian texts

	Nouns			Verbs			Adjectives			Adverbs			All		
	48			13			11			11			80		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Random	0.31	0.34	0.32	0.29	0.21	0.24	0.39	0.37	0.38	0.67	0.36	0.47	0.33	0.32	0.33
BFS	0.56	0.61	0.58	**0.56**	**0.39**	**0.46**	**1.0**	**0.91**	**0.96**	**1.0**	**0.55**	**0.71**	0.65	0.63	0.64
Bias	**0.78**	**0.82**	**0.80**	0.3	0.24	0.28	0.9	0.82	0.86	0.84	0.46	0.59	0.76	0.72	0.74
Rank	**0.78**	**0.82**	**0.80**	0.23	0.16	0.19	0.9	0.82	0.86	0.84	0.46	0.59	0.74	0.69	0.71
Rank+BFS	-	-	-	-	-	-	-	-	-	-	-	-	**0.80**	**0.75**	**0.78**

Table 6: Evaluation results for the English texts

	Nouns			Verbs			Adjectives			Adverbs			All		
	51			17			10			11			84		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Random	0.37	0.32	0.34	0.17	0.24	0.20	0.67	0.60	0.64	0.34	0.28	0.30	0.30	0.30	0.30
BFS	0.32	0.28	0.30	**0.25**	**0.36**	**0.30**	**1.0**	**0.9**	**0.95**	**0.56**	**0.46**	**0.50**	0.40	0.41	0.40
Bias	0.68	0.57	0.62	0.17	0.24	0.20	**1.0**	**0.9**	**0.95**	0.45	0.37	0.40	0.55	0.55	0.55
Rank	**0.70**	**0.58**	**0.64**	0.21	0.30	0.25	**1.0**	**0.9**	**0.95**	0.45	0.37	0.40	0.56	0.56	0.56
Rank+BFS	-	-	-	-	-	-	-	-	-	-	-	-	**0.59**	**0.60**	**0.60**

From the results we can see that our Bias and Rank components perform better than the baselines for nominal expressions. This improvement is aided by the fact that our strategy can detect multiword expressions, while BFS is limited to single words. When applied to other grammatical categories, however, we struggle to match the performance of BFS. For this reason, we have also added the Rank+BFS system that combines ranking for nominal expressions and falls back to BFS for other POS. This system gets the best overall results.

This evaluation will be repeated for the 3rd and last beAWARE prototype, for which we intend to extend the gold standard to cover all the project pilots and languages. We will also use more than one annotator so as to report inter-annotator agreement figures.

Concept extraction

Concept Extraction module plays an important role in the text analysis pipeline and affects retokenization and relation extraction modules which aim at assisting the creations of connected semantic and ontological representations, therefore its performance influences on the final output of the prototype. The most important task is to define correct spans for concepts that should contain words representing together a single unit of knowledge so that its parts being taken separately do not reflect the meaning and might obstruct correct interpretation and reasoning. Consider the following sentences: a) “I'm surprised no-one's called the *fire brigade*.” b) “If needed, additional forces can be mobilized from the National Guard Reserve to assist the local *fire rescue team*.” c) “We can confirm *fire services* are attending a *fire* at our Monsoon Forest habitat.” In the first two sentences, the word “fire” is a part of concepts “fire brigade” and “fire rescue team”. In case of a mistaken split of these concepts, the concept “fire” appears as an individual element that may lead to assigning a potentially wrong relation between “rescue team”/“brigade” and “fire” and consequently to a false detection of an emergency event. At the same time, small confidence in the correctness of spans may also lead to missing an event of interest as, e.g., it can happen with the sentence c) where the concept “fire” might be only associated with the false concept “services”. Thus, having high values of both precision and recall is crucial in this domain.

The developed method of extracting the concepts is based on the analysis of statistical and linguistic features of sequences of tokens. The Google N-gram dataset[3] is used to obtain the statistics on the usage of word combinations. Possible part-of-speech chains for matching complex noun phrases as candidates for concepts have been designed. The algorithm consists in the following steps: 1) defining part-of-speech tags for a given text; 2) selecting parts of noun phrases comprising exactly two terms (i.e., “meaningful” words); 3) assessing the significance of each selected part depending on its position within a list of close collocations (differ only by one term) ordered by frequencies; 4) combining intersected significant parts into concepts and leaving the remainder as separate concepts if they form noun phrases

[3] https://books.google.com/ngrams/

by themselves; 5) applying NER module to detect additional out-of-vocabulary multi-word expressions; 6) eliminating detected concepts that have an overlap with named entities; 7) compiling an output list of non-overlapped and non-embedded concepts including named entities as the result to be passed to the next module in the pipeline.

In order to evaluate the module, the sets of tweets from the second pilot were manually annotated in compliance with the following guidelines: 1) annotate only noun phrases as concepts; 2) all nouns should be included into annotation; one noun should appear in the only concept; 3) noun phrase should be treated as a concept if it represents a single piece of knowledge and is closer to semantically undividable unit rather than to compound phrase in a given context; if there are several embedded concepts, the one with the largest span should be annotated; 4) rarely occurred or novel multi-word noun phrase that potentially might become a concept should be annotated as a single concept only if they form a proper name; 5) annotate hashtags as solid concepts; do not parse them into separate words. The obtained annotations were used as ground truth for evaluating the results of the automatic concept extraction

Table 7: Performance indicators for concept extraction

	Precision	Recall	F_1-score
English	0.887	0.712	0.79
Italian	0.7	0.712	0.706

The values of key measures for English and Italian are presented in Table 7. The obtained results show that the module performs with relatively high precision and recall and might be already used within an emergency event domain. Taking into account the higher value of precision for English, it should be noticed that there is a room for improvement of the method for Italian. To determine possible improvements, a qualitative evaluation required will be conducted and reported in future versions of this deliverable.

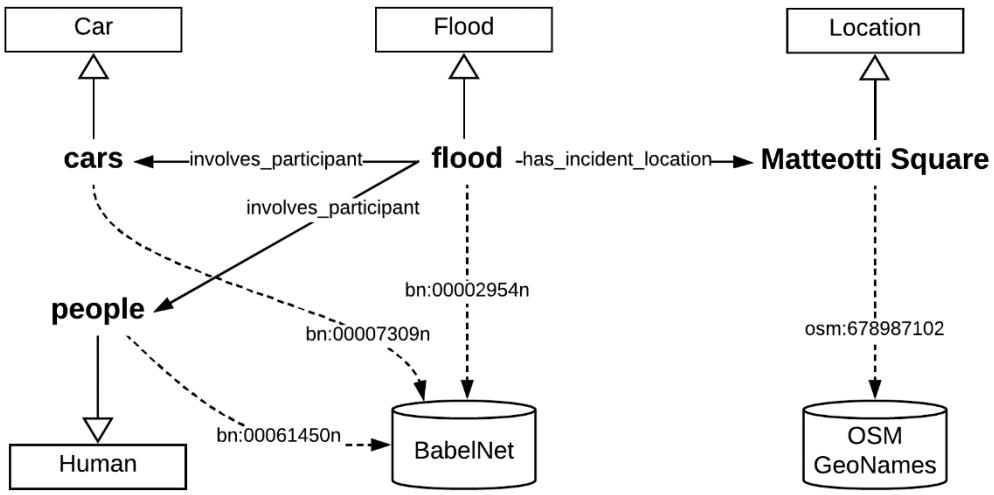
Text analysis

As mentioned in Section 2.3.6 our text analysis pipeline produces a linguistic representation -semantic graph- that is used as an intermediate representation to be mapped to the project ontology. The final output is an n-ary relation, where the relation instantiates one of the incident types modeled in the ontology, and the arguments are instances of either the classes modelling impacted objects in the ontology or of the class *Location*. The relation and arguments are connected by the ontology *involves_participant* and *has_incident_location* properties. Instances may be associated by the id of a BabelNet synset. In addition, coordinates and references

to geographical databases produced by the geolocation module may be associated to arguments indicating locations.

An example of the output of the text analysis module is given in Table 8 for the sentence “Flood is dragging cars and people in Mateotti Square”. An incident of type **Flood** has been extracted that has two impacted objects as participants, one of type *Car* and another of type *Human*. Both the incident and the participants have been linked to BabelNet. “Mateotti Square” has been detected as a *Location* and linked to a geographical database.

Table 8: example of output from text analysis module



In our qualitative evaluation using the 2nd pilot tweets we focus on the extracted relations, or more precisely, on the extracted incidents, their participants and locations. We exclude from this evaluation the links to BabelNet -evaluated in Section 3.1.6 – and the links produced by geolocation – to be evaluated in upcoming deliverables. We also restrict our evaluation to the information about incidents that can be directly modelled using the ontology classes *Incident*, *Vulnerable Object* and *Location*, and the properties *involves_participant* and *has_incident_location*. In our example above, for instance, we would the impact type indicated by the word “dragging”.

Considering all the above, we have manually analyzed the texts shown in Table 3. The expected analyses for each text are listed in Table 9. Please note that since the output of the text analysis module is expressed in terms of a non-linguistic ontology, the same output is expected for an Italian tweet and its English translation. Each output cell specifies one or more ontological classes describing incident types and, for each, a list of classes indicating types of vulnerable objects participating in the

event. Those participants that also correspond to locations are assigned an additional class *Location*.

Table 9 also contains the actual output from the text analysis module for both Italian and English tweets. Comparing these results to the gold analysis shows that, in general, the module can correctly detect the incident being communicated and its impacted objects. Detection of the main incident only failed in texts 10 and 13. The former does not make any explicit mention of any incident. Inferring from the text that some kind of event is taking place in Piazza Matteotti is beyond the capabilities of our pipeline. In the second text the system fails to recognize "exploding"/"scoppiando" as an overflow event. Our strategy for mapping words and BabelNet synsets doesn't consider mentions of these verbs as potential indications of *Overflow* events, as they often indicate very different types of incidents. This could be improved, however, by checking if the impacted objects include any suitable impacted objects, as is the case of manhole covers.

Table 9: Expected and actual outputs of the text analysis module

	Expected output	**English output**	**Italian output**
Text 1	FLOOD EMBANKMENT BRIDGE, LOCATION	FLOOD EMBANKMENT BRIDGE	FLOOD EMBANKMENT BRIDGE
Text 2	OVERFLOW EMBANKMENT RIVER, LOCATION	OVERFLOW EMBANKMENT RIVER	OVERFLOW EMBANKMENT RIVER
Text 3	OVERFLOW PONTE PUSTERLA RIVER, LOCATION	OVERFLOW BRIDGE RIVER	OVERFLOW BRIDGE RIVER
Text 4	FLOOD SQUARE, LOCATION	FLOOD SQUARE	FLOOD SQUARE
Text 5	FLOOD SQUARE, LOCATION	FLOOD SQUARE	FLOOD SQUARE
Text 6	FLOOD SQUARE, LOCATION	FLOOD SQUARE	FLOOD SQUARE
Text 7	FLOOD CELLAR SQUARE, LOCATION	FLOOD BUILDING SQUARE	FLOOD BUILDING SQUARE
Text 8	FLOOD SQUARE, LOCATION	FLOOD SQUARE	FLOOD SQUARE
Text 9	FLOOD PRECIPITATION	FLOOD	FLOOD PRECIPITATION
Text 10	INCIDENT SQUARE, LOCATION	-	-
Text 11	FLOOD CAR GARBAGE COLLECTION SQUARE, LOCATION	FLOOD CAR GARBAGE COLLECTION SQUARE	FLOOD CAR SQUARE
Text 12	OVERFLOW SEWER SQUARE, LOCATION	OVERFLOW SEWER SQUARE	OVERFLOW SEWER SQUARE

Text 13	OVERFLOW SEWER SQUARE, LOCATION	-	-
Text 14	CRACK LEVEE BRIDGE, LOCATION	CRACK LEVEE BRIDGE	CRACK LEVEE BRIDGE

A significant issue is that none of the locations in the texts were recognized as such, e.g. “Ponte degli Angelli” is recognized as a *Bridge* but not as a *Location*. This is due to problems in the performance of the NER component.

Some errors arise only in one of the languages. This is the case of a *Precipitation* event not being detected in the English version of Text 9, due to limitations in the dictionaries used to map words and BabelNet synsets to the ontology classes. Similarly, ”cassonetti” is not detected as a *Garbage Collection* asset, while “dumpsters” is. The module may also fail to find a more specific class and instead resort to a more general one, as happens with “basement”/”cantina”, which are mapped to *Building* instead of the more appropriate *Cellar*.

While the set of tweets used for this evaluation do not include any text that could lead to false positives, our relation extraction strategy may associate objects as participants of the wrong incidents. Future evaluations will have to cover these situations.

3.1.7 **Automatic Speech Recognition**

Evaluation of the ASR module after the 1st Pilot in Thessaloniki (Greece) was focused on the Greek language model. Adaptation methodology and transcription results for the Greek model were presented in D7.5 and D1.3. During the preparation of the 2nd Pilot, the Italian model was likewise evaluated, using case specific Italian audio recordings. The Italian model used in beAWARE is based on an open-source CMU model[4] that has been adapted to speech recordings provided by CERTH and AAWA. For the evaluation, a set of 6 flood-related sentences, was dictated by 3 subjects, resulting in a set of 18 sentences. Audio files were subjected to speech recognition and transcriptions were automatically compared to the original, manually annotated sentences, by using 5prealpha release of Pocketshinx[5]. The measure used for

[4] https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/Italian/

[5] https://github.com/cmusphinx/pocketsphinx

evaluation was percent Word Error Rate (%WER), which is defined as: %WER= 100*(S+D+I)/N, where 'S' is the number of word substitutions, 'D' is the number of deletions, 'I' is the number of insertions and 'N' is the number of words in the reference text. Additionally, percent Word Accuracy is used, which is defined as %WAcc = 100 - %WER. The Italian model produced a WER=21.2% and a corresponding %Wacc=78.8%. The result transcriptions are satisfying, considering that Italian is still an under-resourced language. However, there is still space for recognition improvement, considering that the corresponding %WER and %Wacc for Greek language was 18.2% and 81.8% respectively. The main modifications that have been planned until the 3rd prototype is the expansion of the Italian dictionary, since the available open-source dictionary is relatively limited and the utilization of advanced noise reduction algorithms.

The integrated call center was also evaluated, with respect to its timing performance. In particular, emergency calls received by the call center are being stored on an FTP server. Subsequently, a python script is set to periodically check the server for new recordings, every 6 seconds. The recordings are then uploaded to beAWARE storage server and a message is sent to the message bus, as a 021 topic, in order to inform the Media Hub and subsequently the ASR component. After evaluating the call center solution with several testing calls, we estimated t1, which is the initial period from the end of the call until the query to the server and t2, which is the time between the query and the creation of the 021 topic. As expected, t1 spanned from 0 to 6 seconds, with average value around 3 seconds. However, this can be reduced by decreasing the time interval between requests. Additionally, the average value of t2 was around 1.5 seconds.

3.1.8 **Visual analysis**

In this section a general evaluation is presented first, focused on the operation and the timing performance of the VISUAL ANALYSIS components during the 1st pilot in Thessaloniki and the 2nd pilot in Vicenza.

Table 10: Visual Analysis operations during the pilots.

	Number of requests		Avg file size (MB)		Avg duration (sec)
	Images	Videos	Images	Videos	Videos
1st pilot	115	20	1.47	17.87	9.7
2nd pilot	101	6	0.72	18.05	9

We examine several aspects regarding the operation of the VISUAL ANALYSIS components, such as the number of analysis requests that were received during the pilots and the mean size of media files. Table 10 sums up the operations that the VISUAL ANALYSIS components were involved in during the pilots. Since the average media file size of the typical analysis request has not changed dramatically between the two pilots, it is safe to assume that the VISUAL ANALYSIS components were tested in similar conditions in both pilots. This is a very important conclusion which gives us the option to make comparisons between performances on both pilots.

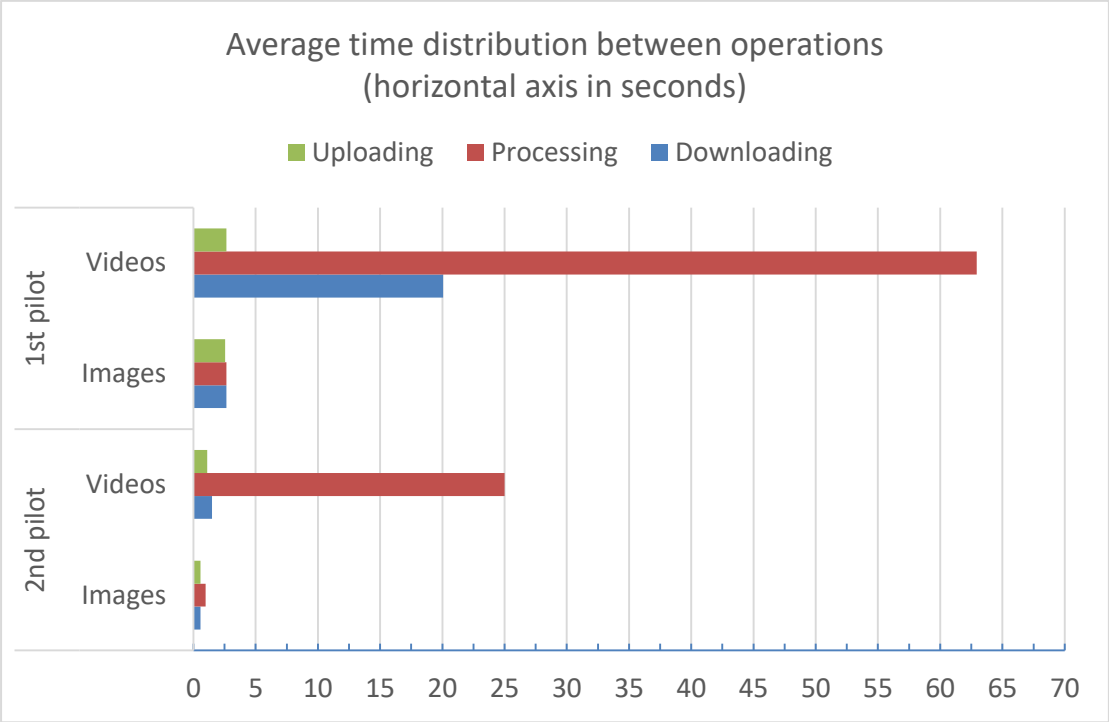


Figure 26: Timing performance for the VISUAL ANALYSIS components during the first two pilots.

The processing of each analysis request is comprised of several operations for which we separately measured the average duration: (a) downloading the media files from the beAWARE storage, (b) the actual processing of the media files and (c) uploading the analysis results to the storage. The average time it took for each operation to process a single media file is shown in Figure 26. It is clear that the components were more time-efficient in the 2nd pilot. Notably, the components required less than half the time in the 2nd pilot to process a media file on average. This is mainly due to the improved version of the Object Detector that was integrated after the 1st pilot which requires less computations. An additional efficiency bottleneck in the 1st pilot, was the integrated Face Detection algorithm that was operated along with the generic Object Detection, so as to more accurately count humans inside places of relief.

Next, the technical evaluation follows for each module of the VISUAL ANALYSIS components based on reports from previous deliverables, relevant publications and data exchanged during the pilots.

Emergency Classification (EmC) / Validation Mechanism

This module is responsible for the distinction between images or videos that show a flood or fire event and those that contain irrelevant content or media not proper for analysis, e.g. the accreditation documents that were exchanged between teams during the 2nd pilot. Note that there are also special cases of analysis requests where a specific flag overrides the validation mechanism. One such case was the traffic analysis requests on the Angeli Bridge in Vicenza that were handled during the 2nd pilot. In these videos, taken from a static camera, it is not the presence of a flooded region that makes the content relevant, but the result of the water level monitoring component. In the case of water level threshold overtopping it was considered useful to also monitor the traffic conditions on the bridge.

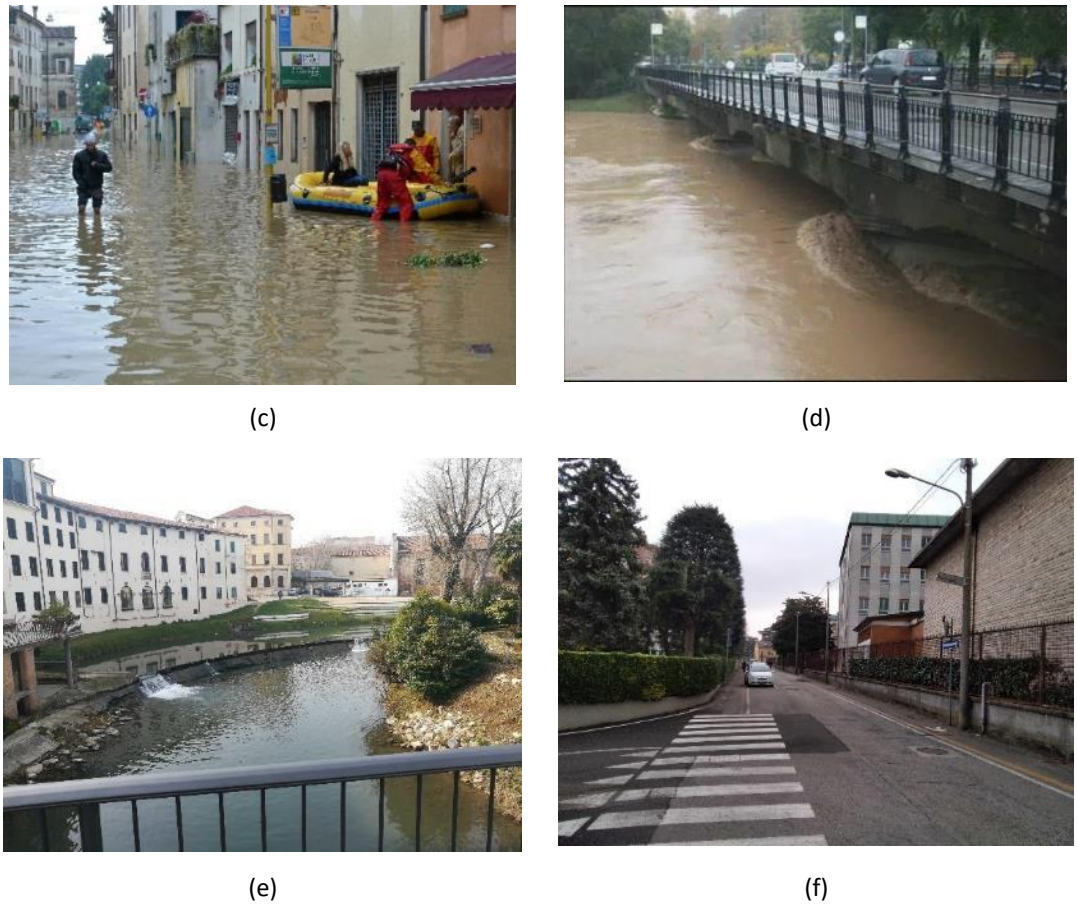
We use the *Classification Accuracy* metric in order to evaluate the emergency classification module. Our component has been evaluated in the MediaEval's Multimedia Satellite Task[6,7], both the 2017 and 2018 edition. According to D3.3 an overall increase in Classification Accuracy of 1.77% is reported comparing to other methods evaluated on the same dataset.

(a)

(b)

[6] Bischke, Benjamin, et al. "The Multimedia Satellite Task at MediaEval 2017." MediaEval. 2017.

[7] Moumtzidou, Anastasia, et al. "A multimodal approach in estimating road passability through a flooded area using social media and satellite images." Proceedings of the MediaEval 2018 Workshop, Sophia-Antipolis, France. 2018.



(c) (d)

(e) (f)

Figure 27: Qualitative evaluation of EmC for the 2nd pilot.

In the heatwave scenario the EmC was not operational, since all the functionality for this particular pilot was related mostly to traffic monitoring and face counting in places of relief and not the detection of a crisis event from images or videos.

For 2nd pilot the EmC was used for the validation of multimedia content by detecting flood events from images and videos. Table 11 shows the performance of EmC during the 2nd pilot. As the table shows, no relevant items were missed from the system. There were only two errors: two non-flood images that were misclassified as flood. Therefore, for the 2nd pilot the classification accuracy reached a satisfying score of 98% correct predictions.

Table 11: EmC performance during the 2nd pilot.

		Prediction	
		Flood	**Not Flood**
Truth	**Flood (Relevant)**	23	0
	Not Flood (Irrelevant)	2	105

In Figure 27 the top row shows the two misclassified errors, the middle row shows true flood images that were correctly detected (True Positives), and the third row shows non-flood images that were discarded by the EmC (True Negatives).

Object and Face detection and Tracking

As for the Object Detection and Tracking (ObD), the *Precision – Recall* graph and the *mAP* metric will be presented to evaluate its performance. This module is the core of two monitoring functions: the capacity of places of relief, and the traffic on the streets of an impacted region. The first function was tested exclusively for the heatwave scenario in the 1st pilot and the second function was evaluated in the 1st and 2nd pilot. The evaluation is performed on the images collected during the pilots. Those images were manually annotated with ground truth bounding boxes of vehicles and pedestrians so as to make a comparison with the detected bounding boxes. More specifically, 954 rectangles were manually annotated in 81 images.

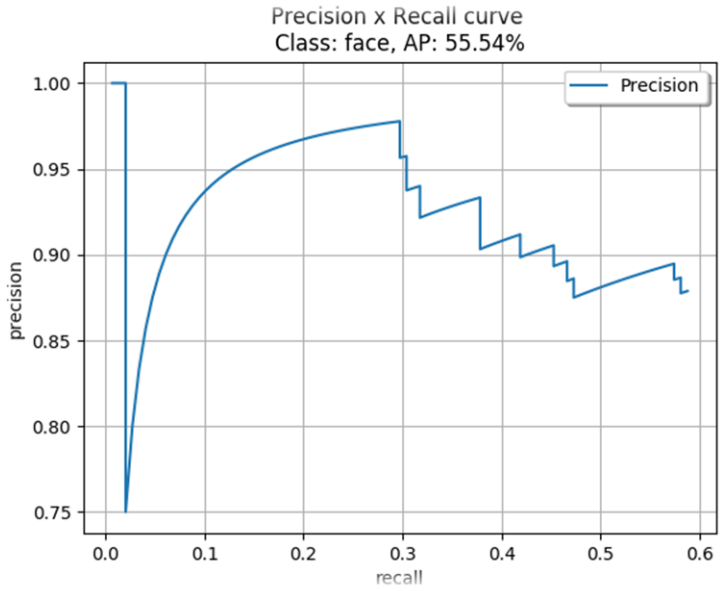
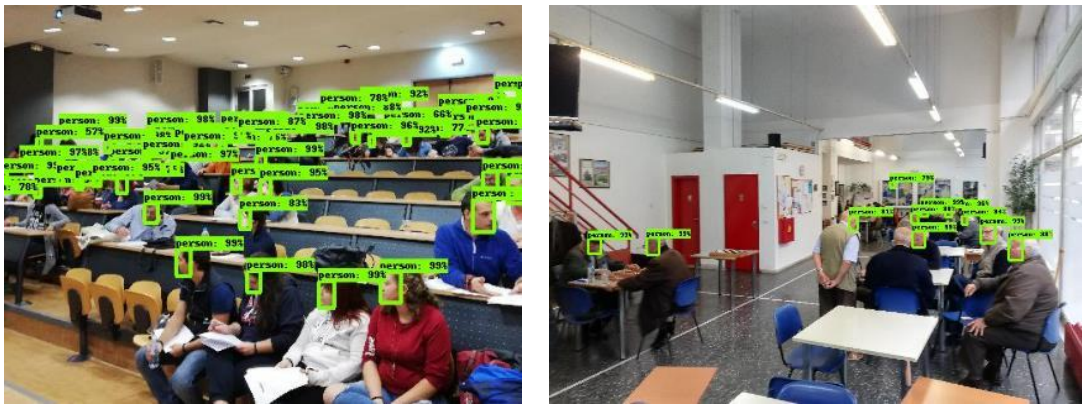


Figure 28: PR curve for people (face) detection during the 1st pilot.



(a) (b)

Figure 29: Qualitative results for people (face) detection during the 1st pilot.

Face detection was used in the 1st pilot to detect people inside places of relief. The overall performance of this module is summarized in Figure 28 which shows the Precision - Recall curve for class 'face'. This graph shows that even for higher values of Recall the Precision is maintained close to 90%, indicating that quality detections have been acquired in that operational range. The performance drops however for Recall values above 60%. This is possibly due to the fact that many people were positioned with their face not shown to the camera and as such the detector could not retrieve all possible ground truth boxes. This conclusion leaves some room for improvement in our approach for future versions of this module. The qualitative results shown in Figure 29 confirm that hypothesis.

For traffic monitoring, we are concerned with the detection of more classes, e.g. five vehicle classes (car, bicycle, motorcycle, bus, truck) and one class for pedestrians. Note that here our generic object detector is developed to recognize the pedestrians from their full human figures and not only by their faces. Table 12 shows the detection performance for each class and each pilot. Note that for the 2nd pilot very few flood images contained vehicles or pedestrians visible, so the results are based on very few ground truth instances and may not be indicative of the performance. This is mainly due to the fact that the flood media files were not captured live on the actual location but were simulated and were limited in number.

Table 12: Object Detection performance during the 1st and 2nd pilots.

		Class						mAP%
		Car	Bicycle	Motorcycle	Bus	Truck	Person	
Average Precision	1st Pilot	60.73%	10.15%	48.23%	50.70%	39.74%	59.21%	44.79%
	2nd Pilot	73.34%	-	66.67%	-	-	47.74%	62.58%

Spatio-Temporal Dynamic Texture Localization

The localization of dynamic textures, such as water and fire, is a challenging task in computer vision domain. In VISUAL ANALYSIS the role of this module is to detect as many flood or fire pixels in the images/videos as possible, in order to later calculate the overlap of the impacted region with bounding boxes of people and vehicles. The overlap percentage is considered as an indication of how close to the danger are the detected targets. In the evaluation of this module we will examine the *Overlap Precision and Recall* metrics.

For the evaluation of the fire localization algorithm a report on D3.3 states that our method, evaluated in the BowFire database[8], achieves 77% Recall score, which is 9% above other baseline techniques. However, Precision scored lower compared to other techniques. A higher Recall score means that we managed to detect most of the ground truth fire pixels but at the expense of higher number of False Positives as Precision suggests. However, when evaluating by F1-Score which is a metric that combines Precision and Recall performance, our method is still well above baseline techniques.

For flood pixel localization, a water texture segmentation algorithm was used. Reports on D3.3 and the relevant publication[9] mention that both Precision and Recall, when evaluating on the VideoWater database[10], were at 92% which means that we got predictions with low False Positive and False Negative rates.

The flood texture localization module was integrated and begun operating for the 2nd prototype and as such it was tested along with the other modules in the 2nd pilot. Figure 30 shows some qualitative examples of the module's performance in flood images. A good amount of flood pixels is shown to be detected (in blue tint) accurately, but there are also some False Negatives. Note that the lack of annotated data for flood regions in images led us to train the models using an existing database made for water segmentation. That means that the module may possibly be stronger in detecting clear blue water regions as opposed to muddy waters of a flooded region. This can explain the False Negatives shown in the examples.

[8] Chino, Daniel YT, et al. "Bowfire: detection of fire in still images by integrating pixel color and texture analysis." 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images. IEEE, 2015.

[9] Giannakeris, Panagiotis, et al. "People and vehicles in danger-A fire and flood detection system in social media." 2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP). IEEE, 2018.

[10] P. Mettes, R. T. Tan, and R. C. Veltkamp, "Water detection through spatio-temporal invariant descriptors," Computer Vision and Image Understanding, vol. 154, pp. 182–191, 2017.

(a) (b) (c) (d)

Figure 30: Qualitative evaluation for flood localization during the 2nd pilot.

Drones Analysis

As described in D7.5, in order to take advantage of the integration of drones in beAWARE and maximize their efficiency, a new analysis module has been added to analyse drone footage, namely Drones Analysis. This module receives sequences of images from the Drones Platform, sent at 1 fps, groups them in batches of 10 seconds and performs object detection and tracking, in order to track people and vehicles in danger. In case of a positive detection, it creates alerts (through dedicated bus messages), containing the type of objects, their position and the analyzed sequence as a video. The results are communicated to PSAP and to Drones Platform. Specifically, drones platform is able, by using location information, to navigate the drone back to the target.

While results of the full detection model (containing people and vehicles) were presented in D1.3, for demonstration purposes, during the 2nd Pilot, a dummy was used, in order to simulate a person in danger. Since the dummy does not exactly resemble a human being, in order to avoid misclassifications or adaptation of the existing model to the dummy, a separate model was trained on footage of the dummy. During the Flood Pilot, in order to demonstrate the whole functionality and communication between the involved components, an autonomous drone flight was

performed on the district of Retrone river in Vicenza. During the drones session, a sequence of 141 images was generated and transmitted by the Drones Platform, during the phase that the drone was scanning the predefined area in order to detect the dummy. This sequence was later annotated in order to create ground-truth images. Instances of the dummy were found in 9 images. During the Pilot, all images were analyzed by the Drones Analysis module. For quantitative evaluation of the detection performance, the overlap between ground-truth and detected bounding boxes of the dummy was taken into account, by estimating the Intersection over Union (IoU) for every pair of bounding boxes. IoU is given by the overlapping area between the predicted bounding box and the ground-truth bounding box divided by the area of union between them. By setting a threshold on the IoU (e.g. 50%) we can tell if a detection is valid (True Positive) or not (False Positive). By comparing ground-truth and detected objects and by setting a IoU threshold of 50%, it was noticed that 9 out of 9 dummy instances were correctly detected (TP=9), without introducing any FP or FN (FP=0, FN=0). Additionally, with the same IoU threshold, the Average Precision was 100.00%, which, in this case, is the same as the mean Average Precision, since there is only one class (dummy). The Precision-Recall Curve is shown in Figure 31, which shows that all objects were correctly classified. It should be noted that the number of annotated boxes is small in order to have more conclusive and representative results (this is why the curve is stable in 100%, without even small fluctuations), however, the most important outcome is that all instances of the dummy were successfully detected. Figure 32 shows an example of a correctly detected object.

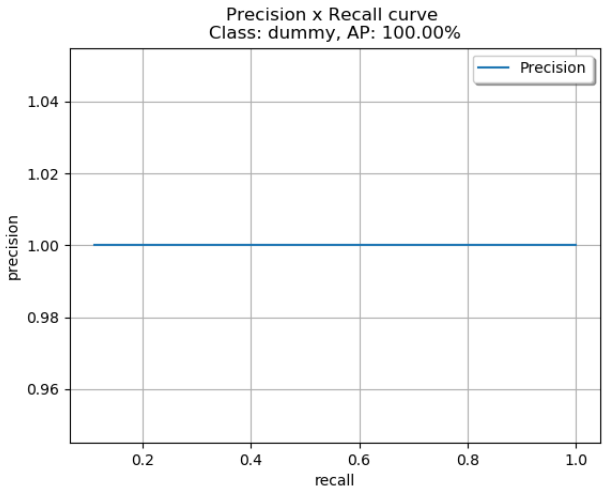


Figure 31: Precision-Recall Curve for the detection of the ‘dummy’ instance by the Drones Analysis.

Figure 32: An instance of a correctly detected object (dummy).

The following figures contain evaluation results for the timing performance of Drones Analysis module. It should be noted that Drones Analysis analyzes each batch of images sequentially, because parallel processing of many batches could create an extreme demand for GPU memory and could possibly cause memory errors. Consequently, the analysis of each batch has to wait the completion of the previous batches. Additionally, some extra delays are caused by the initial 10-second waiting period until all images arrive, the sorting of possibly unsorted arrivals, the storing of the analyzed images on a video and the uploading of the results to the storage server. Figure 33 presents the pure analysis time needed for the analysis of each batch, i.e. the time from the moment the batch has been collected and is ready for analysis until the end of the analysis. It can be noticed that analysis time is relatively stable, since the context is similar for all batches. The average time taken is 9.25 seconds. However, when it comes to the time needed for the full handling of a batch, i.e. the time needed from the completion of the analysis of the previous batch until the completion of the current batch, this period is influenced by the sequential processing of consecutive batches. As it can be seen in Figure 34, the first batches need more time to be processed (average time 46.57 seconds for the first 6 batches) compared to the last ones (average time 36.6 seconds for the last 5 batches). This is because, until the middle of the session, images are still arriving at the analysis module, introducing an additional delay due to image accumulation. Consequently, each batch of images is processed every 36-46 seconds in average, while they are being collected every 10 seconds. This means that each batch is analyzed with a delay of 26-36 in average. The cumulative result can be seen in Figure 35, which

depicts the time passed from the arrival of the first image of each batch until the batch has been processed. Additionally, Figure 36 depicts shows the time passed from the initiation of the image transmission process until the moment each batch is analyzed. In general, for the analysis of 12 batches of images, of total duration of 120 seconds (2mins), the whole communication flow took around 500 seconds (8.3mins). This is sufficient for short drone sessions, however, in larger sessions, delays could increase significantly. Until the final prototype, the code is planned to be optimized, with the respect to computational speed, by speeding up each analysis step, where possible, and by examining the possibility of multithreading. Additionally, Drones Analysis was evaluated while running on a windows server inside an Anaconda environment[11], which in many cases has been proven to deteriorate speed performance. The deployment on a Linux server without the use of Anaconda environment will be examined.

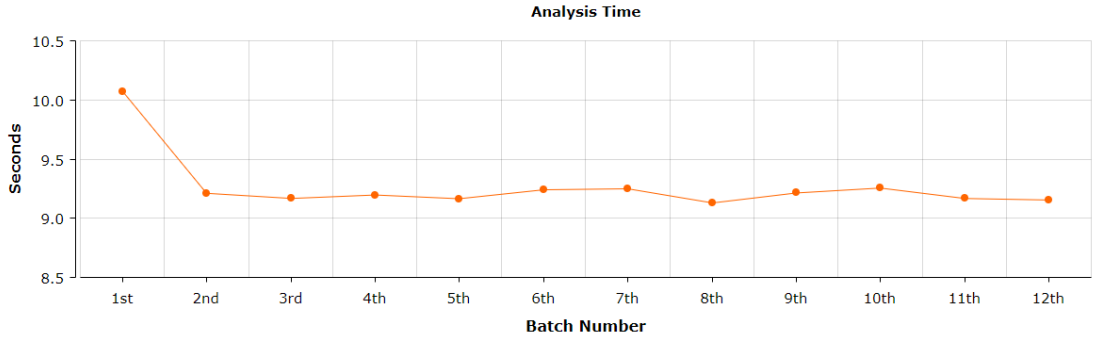


Figure 33: The pure analysis time for the analysis of each batch of images (time since the batch has been collected and is ready for analysis until the end of analysis).

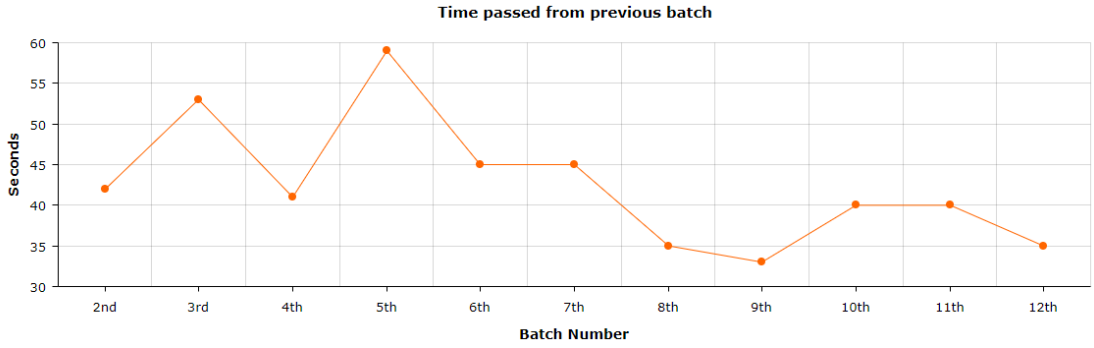


Figure 34: The time that has passed between the completion of the previous batch until the completion of the current batch.

[11] https://docs.anaconda.com/anaconda/navigator/tutorials/manage-environments/

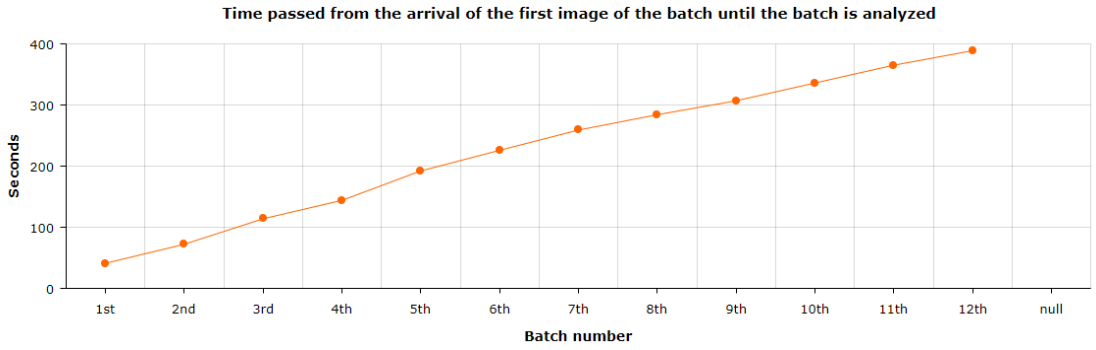


Figure 35: The time interval from the arrival of the first image of each batch until the batch is analyzed.

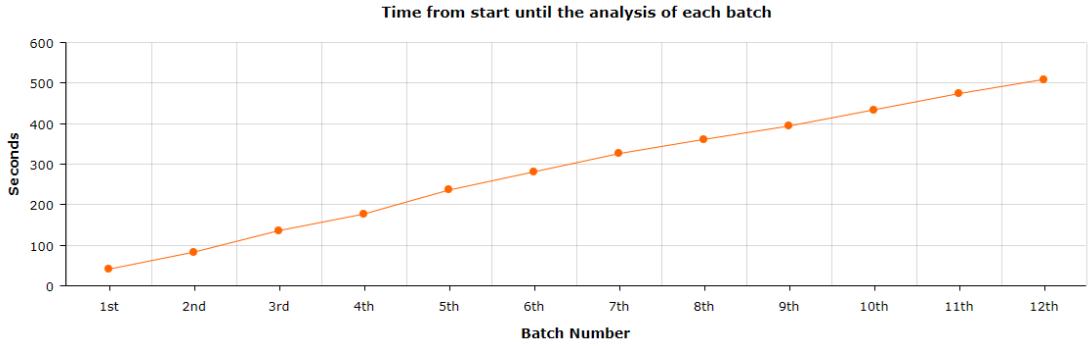


Figure 36: The time passed from the start of image transmission until the analysis of each batch.

Visual River Sensing

Another recently added visual analysis module is the Visual River Sensing (VRS), which performs visual analysis on videos from static surveillance cameras installed by the river, in order to monitor the water level and create alerts, in case some predefined thresholds are exceeded. This module has been calibrated for a surveillance camera installed in Bacchiglione river in the center of Vicenza (Angeli Bridge) and can easily be adjusted to other cameras. VRS streams video-frames directly from the IP of the camera and creates a short video chunk in order to be processed. An example captured frame can be seen in Figure 37. The Figure depicts Angeli bridge, a part of the Bacchiglione river and an old rod (marked inside a red box), placed on the bank of the river, that was used for measuring the water level, before the installation of water level sensors. Apart from water level estimation, the video chunk is also forwarded to Traffic Analysis component, in order to obtain a better overview of the flood event.

The water level estimation module uses an edge detection algorithm in order to detect the marker (rod), which is of known length. After detecting the marker, the algorithm estimates the distance in pixels between the highest detected points and

the lowest detected points of the marker (that should mark the surface of the water). This distance corresponds to the length, in pixels, of the visible part of the marker, which is translated in real length in meters, by using calibration data. Subsequently, the length of the visible part is corresponded to water level. If the water level exceeds some predefined thresholds, three different types of alerts are generated respectively: 'Moderate', 'Severe ', 'Extreme'. The thresholds for the specific camera have been defined by AWAA to 3.0m, 4.6m and 5.4m respectively. Currently, water level is estimated by using only the first frame of the video, however, in the final prototype, the algorithm will use average values from multiple frames, in order to become more robust.

Figure 37: Captured frame for the static camera in Bacchiglione river (Angeli Bridge). The marker has been marked with a red box.

For the evaluation of the module, an annotated series of video captures has been used from 2016, containing 39 videos, spanning from 29/02/2016 09:19am to 01/03/2016 07:57am, when there was a flooding event. The annotations are water level measurements from a sensor installed in the middle of Angeli Bridge. It should be noted, that there is a difference of 0,12m between the sensor measurements and the marker, thus an offset of 0.12m is added after the estimation of water level by VRS. Another issue is the video resolution of the training dataset. The video resolution of the annotated dataset is relatively low (640x340), which results in a resolution of the estimation of around 0.04m. Although, this is sufficient enough for the needs of the task, this resolution could be further decreased during the real-time estimation, since the current maximum resolution of the camera is 1920x1080. The following charts (Figure 38 and Figure 39) present evaluation results of the water

level estimation on the evaluation dataset. More detailed results can be found in the Appendix 6.2 . From the evaluation results, it was concluded that during the day, the detection of the marker was very accurate, resulting to a low percent error, due to good lightning conditions. In particular, the average percent error from 09:19:51 to 18:02:27 was 2.62%, with only one outlier. However, during the night, the selected parametrization of the detector failed to detect the marker, resulting in a blank detection image, due to totally different lightning conditions. To solve this issue, the algorithm was modified in order to use a different parametrization (that performed well during the night), in case no pixels are detected, which leads to a high detection accuracy even during the night. The main remaining issues, are some inaccurate detections during dusk (e.g. 19:19:38) and dawn (e.g. 06:12:16, 06:53:46), due to particularities in the lightning conditions. Contrary to videos captured at night, the problem in these specific cases is that, even though the detection is not accurate, however some pixels are being detected, so the algorithm does not change parametrization. In order to solve these types of errors, in the final prototype the detector is expected to become more robust by investigating different edge detection parametrizations, that could lead to more global results or, if this is not possible, to take into account the timestamp of the video in order to adjust the detector accordingly. Additionally, the detection will be enhanced by using average values of multiple frames, instead of just using a single frame, as in current implementation. Finally, the use of a higher video resolution will be explored, in order to further improve accuracy, by an efficient compromise between resolution and transmission rates during streaming from the static camera. Figure 40 and Figure 41 contain some examples of accurate and inaccurate detections of the marker respectively.

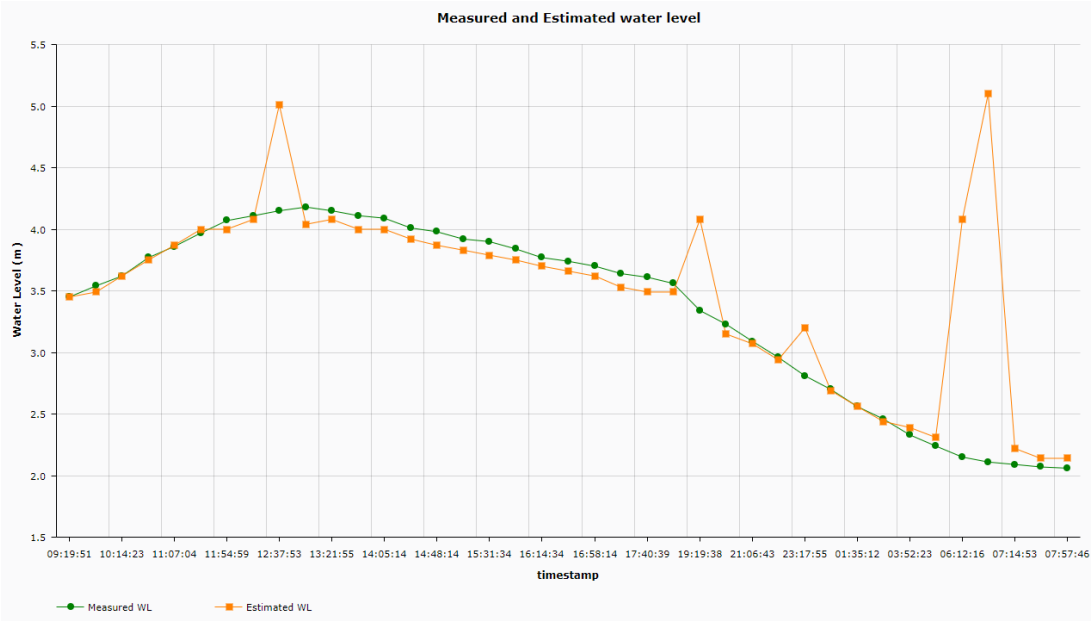


Figure 38: Comparison of water level values measured from sensor and estimated by VRS.

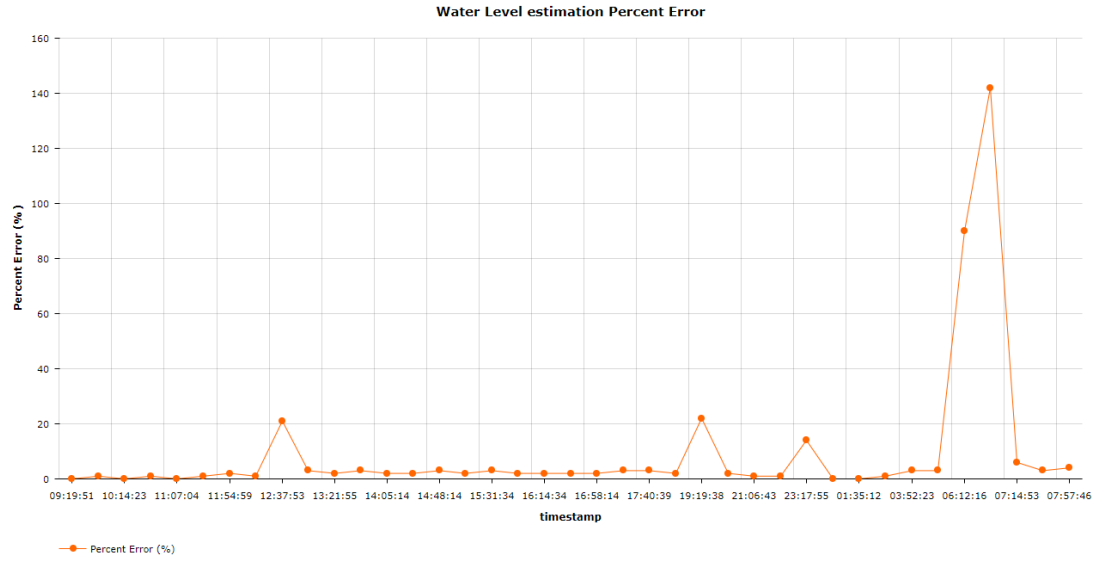
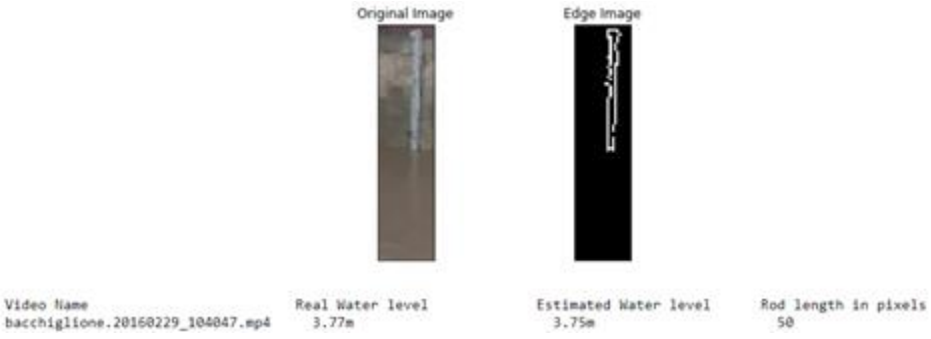
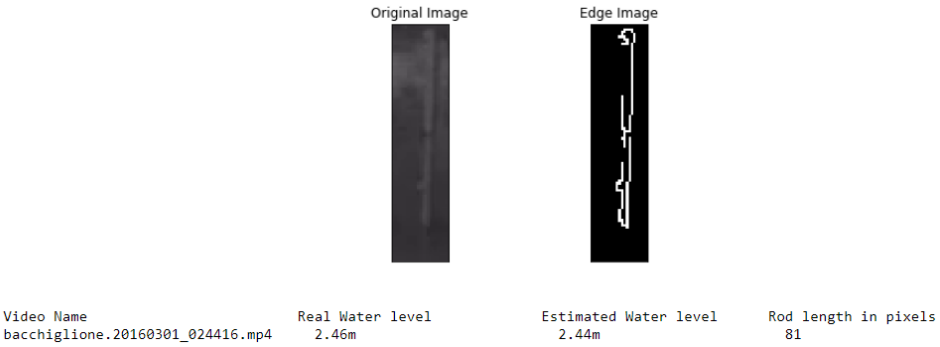


Figure 39: Percent Error between measured and estimated water level values.



(a)



(b)

Figure 40: Examples of accurate estimations of water level during: (a) the day (09:47:36-29/02/2016) and during (b) the night (02:44:16-01/03/2016). Left images contain the cropped image of the rod and right images the corresponding edge detection result.

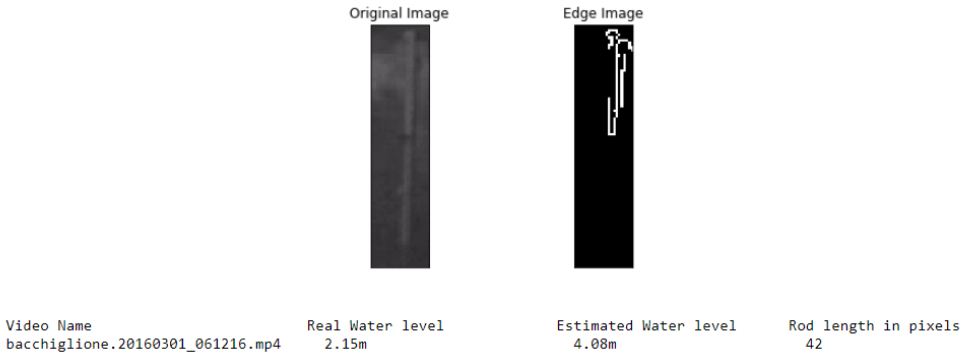


Figure 41: Example of inaccurate estimation of water level during dawn (06:12:16-01/03/2016)

VRS was also evaluated regarding time efficiency. VRS module is set to periodically check water level every 20mins. The streaming part is set to last 10secs. After streaming, the video-file that is created is forwarded to the analysis module. In order to evaluate the time efficiency of the analysis module, analysis time was measured on 10 consecutive runs and the average time was 3.28 seconds.

3.1.9 **beAWARE Knowledge Base**

The central point for the beAWARE Knowledge Base (KB) is the beAWARE ontology. Therefore, the evaluation focuses on the ontology. The beAWARE ontology has been published by Kontopoulos et al. (Kontopoulos)at the ISCRAM conference. The paper contains a detailed description and evaluation of the beAWARE ontology. Therefore, we just summarize the evaluation results in the deliverable. More details can be found in the mentioned publication.

The ontology is served through the knowledge base. The implementation of the KB forces the syntactical correctness. The KB also includes a reasoner for the ontology. No separate check for the ontology with an external reasoner (like stated in the

performance indicator description) was needed, since the ontology was continuously verified by the KB. For assessing the ontology quality the OOPS! – OntOlogy Pitfall Scanner[12] was used. The pitfall scanner allows to search in the ontology for common errors and unconventional structures. This will point out if best practices are violated. No critical issues were found when testing the beAWARE ontology and an important issue has already been fixed previously. The ontology structure was evaluated by OntoMetrics[13]. For the results can be found in Table 13. For a detailed discussion about the meaning of each measurement, we refer to the publication.

Table 13 Ontology metrics produced by the OntoMetrics tool

Base Metrics	Class count	38
	Object property count	37
	Data property count	22
	SubClassOf axioms count	21
	Disjoint classes axioms count	2
	Inverse object properties axioms count	18
	Transitive object property axioms count	2
	Symmetric object property axioms count	1
	DL expressivity	$SI^{(D)}$
Schema Metrics	Attribute richness	0.578947
	Inheritance richness	0.657895
	Relationship richness	0.609375
	Axiom/class ratio	10.184211
	Class/relation ratio	0.59375

The knowledge base is the central component, storing the semantic data. The execution of the pilot showed that the knowledge base was able to answer all the competency questions (in-depth described in D4.2 Semantic representation and reasoning) needed during the pilot use case.

[12] http://oops.linkeddata.es/

[13] https://ontometrics.informatik.uni-rostock.de/

beAWARE Knowledge Base Service

As presented in 2.3.9 , a quantitative evaluation approach has been applied to measure the temporal efficiency of the KBS. In particular, all the Kafka bus communications of the beAWARE pilot in Vicenza were reproduced and a thorough dataset of execution durations, in association with the reported incidents volume, was generated. The analysis of this knowledge is essential to appraise the performance of the KBS-WG duo. A presentation of the resulted outputs is demonstrated next.

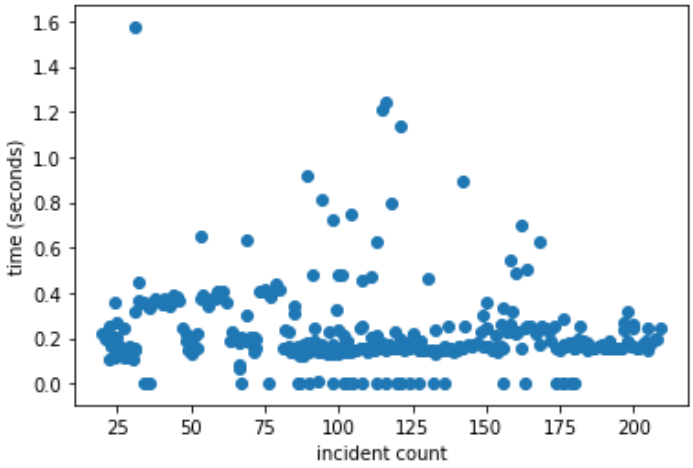


Figure 42: Duration of semantic fusion (population to ontology) of incoming knowledge, in conjunction with the number of submitted incident reports.

Figure 42 presents the execution time of the ontology population algorithms for all Kafka messages addressed to the KBS. The majority of timings demonstrate values bellow 0.4 seconds throughout the whole incident report count axis, with certain few exceptions. Such deviations are to be expected, due to the different nature of incoming messages. For instance, a video analysis communicated to the KBS usually contains a large number of detected incidents and vulnerable objects - proportionate to the video duration - hence increasing the population's time consumption.

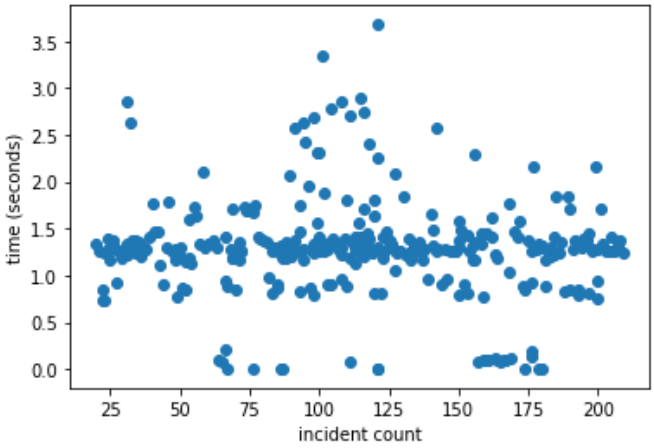
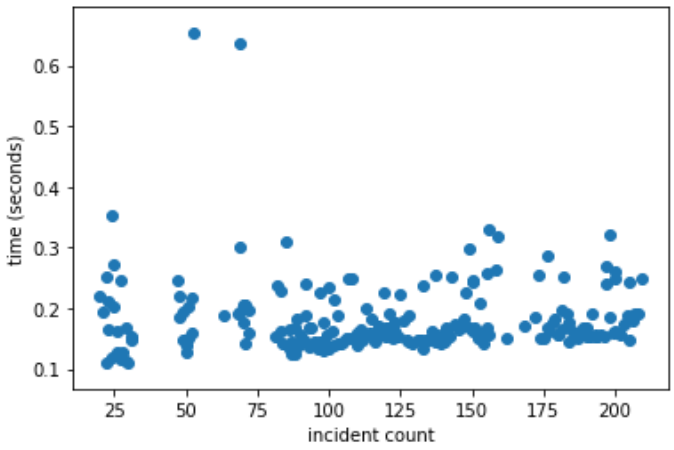


Figure 43: Duration of semantic reasoning on the populated knowledge, related to the number of submitted incident reports.

Following the semantic fusion, the semantic reasoning process is applied to the populated data. This mechanism intends to correlate new with existing knowledge, create associations between entities, calculate incident severity levels and communicate its findings to other components (e.g. PSAP). Since it requires the revision of all stored data on each repetition, it is expected to present inflation in execution times while a crisis progresses. However, for the studied dataset, which is a sufficient simulation of a one-day crisis, such behaviour is not detected.

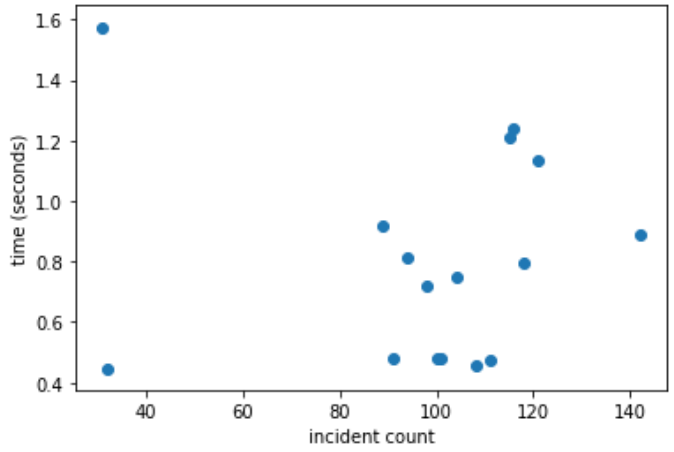
As in Figure 42, few Y-axis deviations occurred in Figure 43, based on the increased/decreased complexity in the handling of specific messages. Within this scope, it was considered useful to study the performance of KBS-WG per distinct Kafka message types. Therefore, some results are demonstrated bellow.




```
Count:207
Time range:0.11  0.654
Incident range:0.0  189.0
Average time:0.184
```

Figure 44: Duration of TOP021_INCIDENT REPORT message handling (semantic fusion and semantic reasoning).

Upon the reception of a TOP021_INCIDENT_REPORT message, a set of instances is populated to the ontology, representing an incident report, its location and the attached media items. Then, the semantic reasoner applies a spatial clustering algorithm and informs the PSAP with the new information. The complexity of these actions is relatively low, a fact also portrayed in the execution times of Figure 44 .




```
Count:16
Time range:0.448  1.575
Incident range:11.0  122.0
Average time:0.805
```

Figure 45: Duration of TOP018_IMAGE_ANALYZED message handling (semantic fusion and semantic reasoning).

Inspecting messages received from the image analysis component (Figure 45), processing times present circumstantial rises to higher values and do not portray a

pattern. This is a rational outcome, since each image analysis may contain a variable number of incident and object detections.

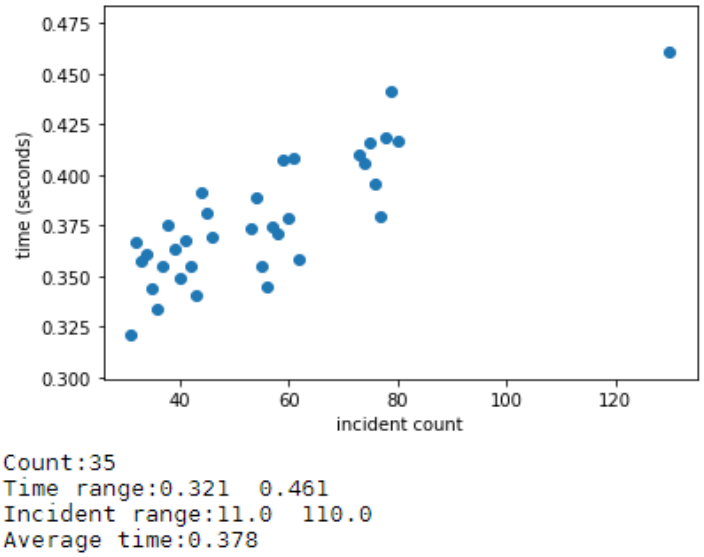


Figure 46: Duration of TOP006_INCIDENT_REPORT_CRCL message handling (semantic fusion and semantic reasoning).

Similar to incident reports originating from users (see Figure 44), Figure 46 shows handling times of incident reports, this time from the Crisis Classification component. Here, the Y-axis value range is quite narrow, revealing an insignificant increase trend while moving towards higher incident counts.

To conclude, the performance of the KBS and WG is subject to quantitative, time-based evaluations. By exploiting the logged communications from the beAWARE pilot in Vicenza, we were able to apply such an evaluation with real-life data and assess the capabilities of these components. Both aggregated and topic-specific outcomes affirm anticipated and reasonable behaviours. It is expected that the processing times should inflate with greater volumes of incoming data, however the available dataset was not sufficient to expose the conditions for such an event.

3.1.10 **Multilingual Report Generator**

Since there are no reference datasets for Natural Language Generation in the crisis management domain or in other languages than English, we report here a quantitative evaluation on general domain data (English), and a qualitative evaluation on the beAWARE domain (English and Italian).

For the quantitative evaluation, we use the BLEU metric on English data. BLEU is an n-gram-based comparison score obtained by comparing a predicted output,

produced by our generator, with the expected one: single words, bigrams (sequences of two words), trigrams and quadrigrams in both outputs are compared and the similarity between them is calculated. As dataset, we use the whole evaluation section of the dependency version of the Penn Treebank (Johansson & Nugues, 2007), converted to predicate-argument structures, using the semantic analyser described in (Mille, Carlini, Latorre, & Wanner, 2017). The converted semantic structures are then sent to the beAWARE generator (FORGe), replacing the linearization module by an off-the-shelf linearisation tool used for previous evaluations, so that the results are fully comparable.[14] As a result, the improvement of the score is due almost exclusively to the sentence structuring grammars (Sem-DSynt and DSynt-SSynt), which are the core grammars of the beAWARE generation pipeline. The BLEU score increased 4.31 points since the beginning of the project, from 35.53 to 39.84, which represents an increase of 12.1%, better than the highest expectation from D1.2 (10%).

For the qualitative evaluation in the beAWARE domain, we selected 10 reports with different levels of complexity generated by the system in English and Italian. For each report, human assessors were presented the automatically generated report, and a manually written report. For each report, human assessors were asked to give ratings according to three evaluation criteria: Intelligibility, Fluency/Naturalness, and Accuracy. Each one of these three criteria was rated on a 1 to 5 scale, with 5 being the best rate and 1 the worst, according to the following instructions:

INTELLIGIBILITY

Evaluate the rate of grammatical errors (syntactic constructions, word agreements, basic word order) and bad word choices.
5 - [No mistakes]: the meaning of the text is clear, and there are no questions. Grammar and word choices are all correct. 4 - [Minor mistakes]: the meaning of the text is clear, but there are some problems in grammar or word choices. 3 - [Some mistakes but understandable]: the basic thrust of the text is clear, but the evaluator is not sure of some parts because of grammar or word choice problems. 2 - [Basically wrong]: the text contains many grammatical or lexical problems, and the evaluator can only guess at the meaning after careful study, if at all.

[14] Note that the beAWARE generator is designed primarily to perform well on the crisis management domain, in which the variety of sentences is not comparable to that of general domain text. The linearisation (word ordering) module of beAWARE is not yet fully operational on general domain texts, so using a statistical linearisation tool also allows for achieving a large coverage during the word ordering task, needed for this evaluation.

1 - [All wrong]: the text cannot be understood at all. No amount of effort will produce any meaning.

FLUENCY/NATURALNESS

Sometimes everything is grammatical, but considering each sentence individually, should it be split or merged with another one? Should a (part of a) sentence be moved to another place in the text?
5 - [No rewriting]: the text is very easy to read; no rewriting is needed whatsoever. 4 - [Minor rewriting needed]: the text is easy to read although some sentences could be split, merged together or moved to make it better. 3 - [Some rewriting needed]: the text is easy to read but clearly unnatural. 2 - [Heavy rewriting needed]: some part of the text is difficult to read, because of poor sentence delimitation or ordering. 1 - [Complete rewriting needed]: the general style and ordering of the sentences prevents an easy reading of the text: sentences are clearly too long or too short, or clearly out of place.

ACCURACY

Evaluate if the content of the text is faithful to the contents to be verbalized. Each text is followed by a content plan (a list of triples, see below) which specifies the contents for each text. You must rely on this content plan to rate accuracy. The content plans are very straightforward and there are no tricks: for instance, you do not need to assess things like 'there is an Agent in the content plan, but it looks like more like an Undergoer in the text'. Focus simply on if the contents mentioned in the plan are in the sentences.
5 - [No content missing/added]: the content is faithfully conveyed in the output text; no information is missing, and no useless information is added. 4 - [Few minor content missing/added]: most of the content faithfully conveyed in the output text; only few information of minor relevance does not appear or is added in the final text. 3 - [Some content missing/added]: some minor content is missing or has been added, or the evaluator is not sure whether this is relevant information or not. 2 - [Some important content missing/added]: the content is not adequately conveyed to the output text. Some important information is missing or has been clearly unnecessarily added. 1 - [Lots of important content missing/added]: the content is not conveyed at all to the output text. Too much information is missing, or the text has very little relation with the original content.

The input structures for generation are triple sets, from 2 triples for the simplest report, up to 12 triples for the most complex one. A sample set of sentences to

evaluate looks like the following, where (1) is the human-produced report, and (2) is the report generated by the beAWARE platform:

TEXT 6 (P2a-6):

(1) There is a flood. Many cars and many people are affected. The river Bacchiglione flowed over its banks and the levee collapsed at Leonardo bridge. Matteotti square went underwater and the sewers are flooded.
Intelligibility: Fluency/Naturalness: Accuracy:

(2) A flood is reported. Many cars and many people are impacted. The Bacchiglione has overflowed. The levee, cracked at Angeli bridge, has collapsed in Leonardo bridge. The sewers and Matteotti square are flooded.
Intelligibility: Fluency/Naturalness: Accuracy:

Content Plan of Text 6 for Rating Accuracy:

Undergoer(to_report, flood)
Undergoer(to_impact$_1$, car)
Undergoer(to_impact$_2$, people)
Quantity(car, many)
Quantity(people, many)
Agent(to_overflow, Bacchiglione)
Agent(to_collapse, levee)
Location(to_collapse, Leonardo Bridge)
Undergoer(to_crack, levee)
Location(to_crack, Angeli Bridge)
Undergoer(to_flood$_1$, sewer)
Undergoer(to_flood$_2$, Mateotti square)

A small document describing the content plans was provided to the evaluators, so that they were able to understand the contents and apply correctly the Intelligibility criterion. 8 evaluators assessed the 10 English reports, and 4 evaluators assessed the 10 Italian reports according to the three criteria. The results of the evaluation are provided in Table 14.

Table 14: Results of the human evaluation for the beAWARE report generation

		System	Human
English	Intelligibility	4,35	**4,64**
	Fluency/Naturalness	4,06	**4,45**
	Accuracy	**4,78**	4,26
Italian	Intelligibility	4,15	**4,43**
	Fluency/Naturalness	3,9	**4,4**
	Accuracy	**4,65**	4,58

Table 14 shows that the results are consistent across languages: the average Intelligibility and Fluency/Naturalness is higher in human-produced reports, whereas the automatically-generated reports are in general more accurate. Nonetheless, the beAWARE system achieves high scores for the Intelligibility criteria, an average 0,3 points lower than the human reports. The Accuracy of the system is very high, since the totality of the contents is always generated, whereas humans tend to favour Fluency/Naturalness when they write, at the expense of reflecting exactly the contents of the inputs. In the last phase of the project, we will focus on improving the Intelligibility and Fluency/Naturalness of the system-generated outputs, with the objective of getting closer to the human-generated ones in terms of their quality, without losing Accuracy.

3.1.11 **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 to the drones' platform dashboard, including the route of the current stage and imagery transmitted by instruments on the drone.

The first part of the demonstration (first execution block) consisted of a scan of a pre-defined area. The demonstration started with the drone going up to the designated flight height of 15 meters, and flying to the starting point of the scanning of the area. The route of the scanning phase was calculated for the drone to cover the designated area

- Overall area length was 130 meters and width 68 meters.
- Distance between scan lines: 14 meters
- Scan speed: 3 meters per second
- Camera gimbal pitch: 45 degrees towards the ground.
- Scan altitude: 15 meters

During the drone operation images were captured and sent in real-time to the beAWARE platform:

- Frequency of captured images was 1 frame per second
- Image resolution: 1280x720 pixels
- Compression: jpeg
- Total amount of images captured: 612
- Total images size per flight: 78.4 MB (126 KB average image size)

The images were sent in real-tine to the beAWARE platform by storing the image in the platform Object Store and sending a message on a pre-defined topic on the message bus, publishing the existence of a new image, including the corresponding metadata. Published images are consumed by the image analysis component. The main capability demonstrated by the image analysis component in this case is the identification of a person in danger (represented by a mannequin lying on the ground) in the imagery coming from the drones. When the image analysis component identifies a person in danger it sends a corresponding notification message through the message bus. In this demonstration the drones platform picks up these messages to be used at a later stage in the demonstration. Images that were reported by the analysis component to be relevant are displayed as well in the dashboard.

Once the scan of the area is completed, the next execution block is loaded, which contains the inspection of several pre-defined points of interest: (Pipes of a pump, Pump, Gate).

In all the points of interest the drone reaches the designated point and lowers its altitude to 10 meters, to send more detailed images. In one of the points of interest, in order to go down from 15 meters to 10 meters in a safe way, avoiding obstacles (power cables), the drone goes down in an L-shape maneuver, (lowering at a safe point from 15 to 10 meters, and then advancing horizontally toward the filming point at an altitude of 10 meters.

At the last stage the drone flies to take a closer look at the person in danger identified in real-time by the image analysis component. As mentioned above the drone platform picks up the information provided by the image analysis component, including the corresponding coordinates. This part demonstrated the dynamic nature of the autonomous flight components, in which not the entire route needs to be available and calculated in advance, but rather the route can be calculated while the drone is in the air based on real-time analysis of current events. In addition, this part demonstrates the bi-directional interaction between the drones platform and

the beAWARE platform. The drones platform registered as a subscriber to a specific message bus topic to which the image analysis component published its finding. The message recited by the drones platform included the location of the “person in danger”, that information in turn was used to dynamically create a new execution block, which was sent to the drone, in which it was instructed to fly back to the position which was indicated by the received message.

To conclude the session the drone flies back home and lands at the point of departure.

3.1.12 **Public Safety Answering Point**

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 .

To evaluate the number of requirements implemented we provide a list of requirements that are relevant to the PSAP design. A reference to URs and the way we addressed them can be found in “D6.5 Advanced Visualization and Interaction for Enhanced Situational Awareness - State-of-the-Art” deliverable.

The following table lists all those relevant user requirements and how they are implemented in the PSAP.

Table 15: User Requirements implemented in the PSAP

UR#	Requirement name	Requirement description	Implied/Expected Interaction Modality[15]
UR_101	Type of visualization	Display information to authorities in a web-gis platform (citizen and first responders reports)	Event Map
UR_103	Flood warnings	Provide authorities/citizens with warnings on river levels overtopping some predefined alert thresholds, based on forecast results	Alert/Warning Display
UR_107	Localize video, audio and images	Provide authorities with the ability to localize videos, audio and images sent by citizens from their mobile phones	Event Map integrated with media viewer
UR_108	Localize task status	Provide authorities with the ability to localize first responders reports regarding the status of their assigned tasks	Event map integrated with report event information
UR_109	Localize tweets	Provide authorities with the ability to localize Twitter messages concerning a flood event	Event map integrated with social media reports
UR_112	Detect element at risk from reports	Provide authorities with the ability to detect the number of elements at risk and the degree of emergency from text sent by the mobile app and by social media	Risk assessment metrics
UR_117	Manage assignments in case of new emergencies	Provide authorities with the ability to manage first responder assignments	Task management interface
UR_118	River overtopping	Provide authorities/citizens with the ability to know if the river level is overtopping predefined alert thresholds	Alert/Warning Display
UR_120	Map of rescue teams and task evaluation	Display to authorities the position of first responder teams in all the municipality and provide the ability to evaluate in real time the execution of the assigned tasks	Event map with informative icon semantics
UR_128	Evaluation of the level of risk	Provide authorities with the ability to evaluate the forecasted level of risks (based on all the available dataset)	Risk prediction metrics

[15] Added in V2.0

UR#	Requirement name	Requirement description	Implied/Expected Interaction Modality[15]
UR_131	Traffic warnings	Provide authorities with the ability to send warnings to citizens in order to avoid a certain area that is jammed with traffic	Public Alert/Warning Editor-Generator
UR_213	Recommendations	Sending recommendations to citizens.	Public Information Editor-Generator
UR_214	Warnings	Sending warnings of pre-emergency alerts to citizens by authorities	Public Alert/Warning Editor-Generator
UR_215	Evacuation orders	Ordering evacuations of citizens at risk.	Public Instruction Editor-Generator
UR_302	Automatic warning	beAWARE system to generate and provide the authorities with an automatic warning when an imminent heatwave phenomenon is forecasted	Event Map Emergency metrics
UR_303	Risk assessment for a forest fire	Provide the authorities with a risk assessment regarding the probability of a forest fire to occur during or in the upcoming period after a heatwave. The relevant authorities will have an assessment of a fire risk based on the weather forecast during a heatwave and especially during the following days	Risk prediction metrics
UR_306	Number of people affected	Provide the authorities an estimation of the people that might be affected from the phenomenon and in which areas	Emergency Statistical metrics
UR_309	False Alarms	Provide to the authorities a procedure to confirm necessity of rescue teams so they are not sent needlessly to one place instead of somewhere else where they are needed more urgently, therefore the ability to handle false alarms.	Task management
UR_310	City-wide overview of the event	Provide the authorities to have a city-wide overview of the event – allow decision making authorities an overall view of all incidents handled at any point in time/ see where all rescue teams are located in real-time to allow them to make informed decisions regarding who to send where… etc	Informative Summary/ emergency overview display
UR_313	First responders status	Provide to the authorities the current status and location of all first responders when they are performing their tasks	Workforce monitoring interface

UR#	Requirement name	Requirement description	Implied/Expected Interaction Modality[15]
UR_314	Assign tasks to first responders	Allow authorities to assign additional tasks to those first responders who are available or even instruct those who are able to assist other responders	Workforce monitoring interface integrated with task management interface
UR_316	Capacity of relief places	Provide to the authorities the current state of the available capacity of all relief places provided to the public	Emergency Statistical metrics
UR_318	Trapped citizens	Allow authorities to know if there are people trapped (e.g. in an elevator) and display where	Map-based Incident display integrated with alerting capability
UR_319	Trapped elders at home	Allow authorities to know if there are elder people trapped in houses without an A/C and display where	Map-based Incident display integrated with alerting capability
UR_320	Hospital availability	Show to the authorities the current availability of the hospitals.	Emergency Statistical metrics
UR_332	Localize tweets	Provide authorities with the ability to localize Twitter messages	Event map integrated with social media reports
UR_334	Manage assignments in case of new emergencies	Provide authorities with the ability to manage first responder assignments	Task management interface
UR_335	Map of rescue teams and task evaluation	Display to authorities the position of first responder teams in all the municipality and provide the ability to evaluate in real time the execution of the assigned tasks with a global visualization of the activities performed	Workforce monitoring interface integrated progress assessment
UR_337	Location of vehicles and personnel involved	Allow authorities/first responders to visualize position of vehicles and teams on the incident site	Map-based Workforce monitoring

PSAP is constantly evolve within the development cycle of the beAWARE project based on evaluation criteria such as relevance, usability, effectiveness. Nevertheless, in the following we present a technical evaluation study that reflects the performance and the technical functioning of the tool based on the incident and metric reports that were sent during the 2nd pilot.

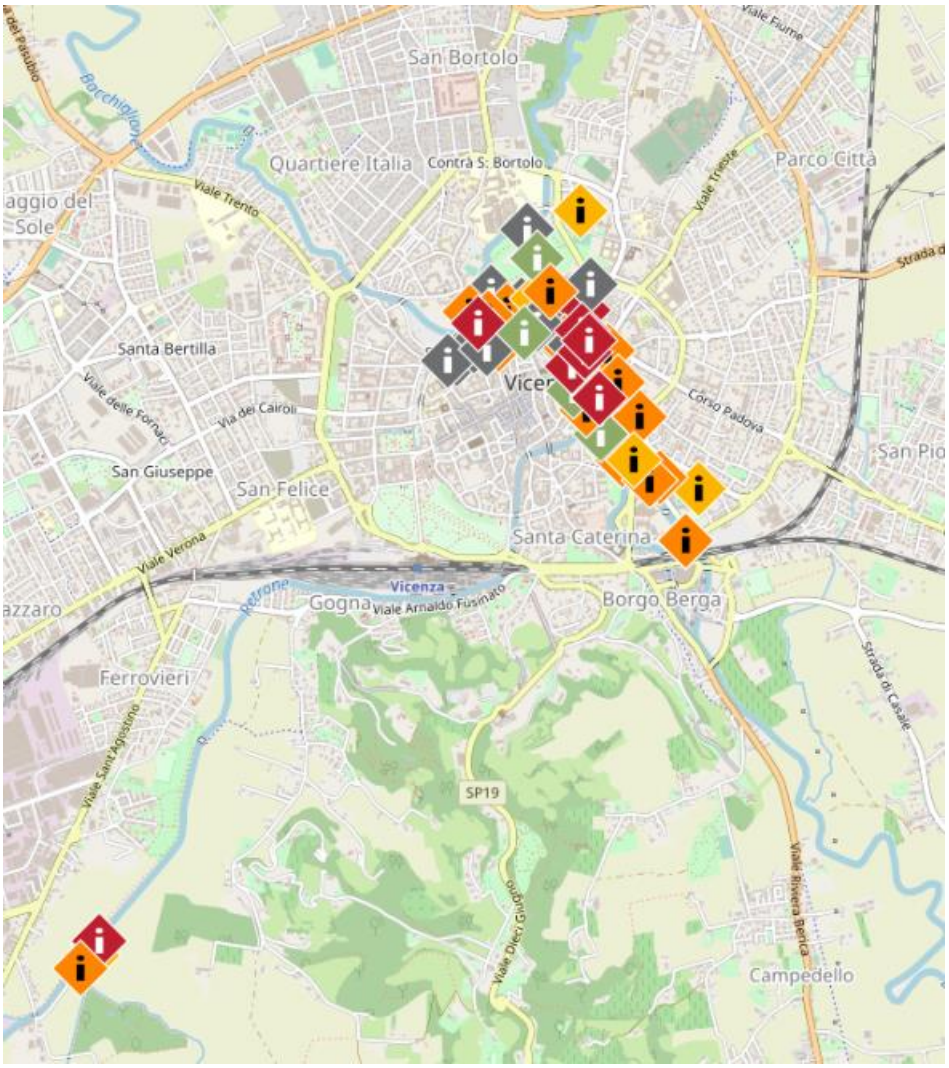


Figure 47: Incident Reports visualised on the Incident Map.

Messages in the PSAP arrive either as metric reports transmitted by the crisis classification module (see 3.1.5) or as incident reports sent by the KBS (see 3.1.9).

Metric reports are shown both on the Incident Map and the Dashboard. These reports are related to weather, sensor and risk data that are processed by the Crisis Classification module, translated accordingly, to metrics for the Dashboard or reports with attributes to be shown on the Incident Map.

Incident reports are shown on the Incident Map. These reports are generated by the KBS based on incidents reported by the users via the mobile application or automatically fetched by the beAWARE system through the call center, the VRS or SMA or they are automatically generated by the Crisis Classification module based on sensor indications.

In the pre-emergency phase, the PSAP visualise the following data:

Component	Description	Type of chart
Dashboard	Forecasted water level from specific river reaches in the flood pilot	Line plot
	Forecasted weather data like temperature, precipitation, etc	Line plot
	Aggregated data like forecasted crisis level per river reach or whole region of interest	Gauge plot
	Distribution of the forecasted crisis level over the locations (e.g. river sections in flood pilot)	Traffic light plot
	Predicted Discomfort Index and Remaining Hours to Heatwave per location	Bar plot
Map	Incident of forecasted water level for those river sections that exceed the 1st alarm threshold	Colored incident on the map
	Polygons obtained by the Risk Maps	Colored Polygons

In the emergency phase, the PSAP visualise the following data:

Component	Description	Type of chart
Dashboard	Observed water level from specific weather stations in the flood pilot	Line plot
	Observed weather data like precipitation, etc from specific locations	Line plot
	Aggregated data like observed crisis level over the whole region of interest	Gauge plot
	Estimation of the ongoing crisis event (Risk Assessment) measured by the estimation of incident reports	Gauge plot
	Observed Discomfort Index Remaining Hours to Heatwave per location	Bar plot
Map	Incident of observed metric (water level and precipitation) for specific weather stations that exceed the 1st alarm threshold	Colored incident on the map

All messages that were sent during the Vicenza Pilot, were reproduced and the average visualisation time per subset and per phase was calculated. The results are gathered in the following tables.

Table 16: Visualisation Time per subset.

Set name	Phase	Component	Quantity	Duration/Time (sec)
Low forecasted measurements	Pre-Emergency	Map	9	3.4 sec
Medium forecasted measurements	Pre-Emergency	Map	9	3.4 sec
High forecasted measurements	Pre-Emergency	Map	12	4.91 sec

Very High forecasted measurements	Pre-Emergency	Map	30	9 sec
Incident	Pre-Emergency	Map	500	145.2 sec
Risk maps (polygons)	Pre-Emergency	Map	41	17.62 sec
Aggregated forecasted measurements for Line plots	Pre-Emergency	Dashboard	6	5.2 sec
Aggregated forecasted measurements for Gauge plots	Pre-Emergency	Dashboard	7	3.7 sec
Aggregated forecasted measurements for Pie charts	Pre-Emergency	Dashboard	6	2.74 sec
Aggregated forecasted measurements for Traffic Light plot	Pre-Emergency	Dashboard	1	1.64 sec

Table 17: Visualisation time per phase.

Set name	Phase	Component	Quantity	Duration/Time (sec)
Real-time measurements - 24 hours before and current measurements (for Line plots)	Emergency	Dashboard	30	14.48 sec
Aggregated real-time measurements for Gauge plots	Emergency	Dashboard	6	2.75 sec
Current low measurements	Emergency	Map	1	2.15 sec
Current medium measurements	Emergency	Map	9	3.4 sec
Current high measurements	Emergency	Map	7	3.22 sec
Current very high measurements	Emergency	Map	9	3.4 sec

Figure 48 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 (Vicenza 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. In general, the PSAP component performed well, with all the real time information to be timely displayed. A shortcoming that was revealed during the pilot is that map's performance drops significantly in panning, zooming and rendering speed when a large number of risk map polygons are drawn on the riskMap layer.

The usability evaluation was conducted based on the feedback of the participants in the second pilot. This feedback was gathered during the debriefing session directly after the pilot and in questionnaires filled out by the participants. The data gathered were used to rate the PSAP for usability and users' satisfaction. Since the evaluation is based on the users' responses the results can be found in the deliverable D2.6 "Evaluation report of the 2nd prototype".

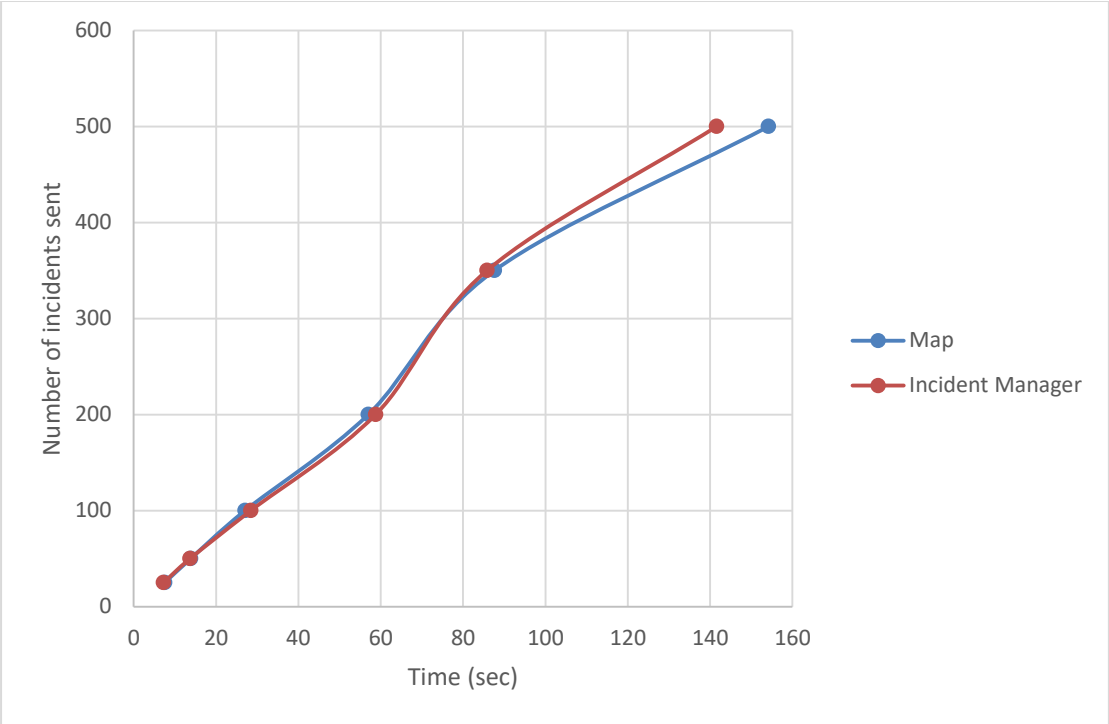


Figure 48: Average visualisation speed when varying the number of incidents received on the PSAP

3.1.13 **Mobile Application**

To evaluate the number of requirements implemented by the mobile app, we will first provide a list of requirements that are relevant for the mobile app, e.g. because they concern the mobile app directly or need specific data from the app. Those requirements with a detailed description can be found in D2.10. The following table (Table 18) lists all those relevant user requirements and how they are implemented in the mobile app.

Table 18: User Requirements as implemented features in the Mobile Application

UR#	Requirement name/description	Implementation in mobile app
UR_103	Flood warnings	Alert mechanism
UR_104 UR_327	Send/receive emergency reports	Incident report mechanism
UR_105 UR_219 UR_314 UR_335	Send task reports Coordination and communication between different resources	Task management mechanism
UR_107 UR_330	Localize video, audio and images	Incident report mechanism
UR_108 UR_331	Localize task status	Incident report / task management mechanism
UR_110 UR_221 UR_333	Localize calls	To realize phone calls, there are two mechanisms implemented: • Audio recording

		functionality in mobile application • Call center functionality in the beAWARE platform
UR_111	Detect flooded elements from video	Incident report mechanism: video recording
UR_112	Detect element at risk from reports	Incident report mechanism: categorization scheme
UR_116	Warning people approaching flood areas	Alert mechanism: limited by location and radius of alert
UR_117 UR_119 UR_325	Manage assignments/tasks of first responders	Task management mechanism
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
UR_133	Send water level estimation from mobile app	Incident report mechanism: categorization scheme
UR_134	Send specific type of incident reports	Incident report mechanism: categorization scheme
UR_135 UR_227	Specific mobile app for first responder and citizen	Login for first responders to enable all features
UR_201	Detection of people and goods in danger	Incident report mechanism
UR_206	Specific weather data	Weather data is not directly provided to citizen. Anyhow this information can be passed to citizens/first responders by alert or task management mechanism.
UR_210	Mobile application	Incident report mechanism
UR_211	Location of personnel involved	Team management mechanism
UR_224	Automatic translation from a foreigner applicant through mobile app	Multilingual support
UR_313	First responders status	Team management mechanism
UR_340 UR_342	Internal sharing of information	Incident report / Task management mechanism

The table shows that all defined user requirements have been addressed in the mobile app implementation, at least at a basic level.

The evaluation of the usability is done in a qualitative way. It is based on the feedback of the people involved during the execution of the second pilot. This feedback was gathered during the debriefing session directly after the pilot and in questionnaires filled out during the debriefing. Since the evaluation is based on the users' responses the results can be found in the deliverable D2.6 "Evaluation report of the 2nd prototype", which will be finished together with this technical evaluation report.

4 Conclusions

The second beAWARE prototype has successfully integrated a number of new modules, as formulated in the DoA. Furthermore, the second prototype has shown many improvements to existing functionalities based on the findings from the evaluation of the first prototype as well as the recommendation from the mid-term review (July 2018). The focus of the work was on improving established services, developing and performance optimisation. Significant improvements in platforms' modules were made in comparison to the First Prototype, by advancing platform's technologies

With regard to the second 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:

The technical infrastructure which consists of the platform backbone performed as expected, providing the necessary throughput and uptime. The infrastructure deployed on the cloud sustained the combined throughput from all platform components, and was available and responsive during the entire pilot.

The SMA module ran smoothly throughout the pilot, offering a three-step validation of incoming data (crawled tweets) and anonymization of sensitive information (user profiles), while it maintained a low process time. The significance of the newly introduced feature of fake tweets detection has been proven by the evaluation. On the other hand, the evaluation of the SMC module is still pending due to the lack of an annotated dataset, but is scheduled to be addressed in the next prototype. Future work also includes the establishment of communication between MTA and SMC, in order to exploit the locations extracted by MTA. In this way locations will not have to be predefined anymore, allowing the system to be more generic.

Crisis Classification component sufficiently and timely process the obtained data (forecasts and real-time observations), providing assessments for the severity of the upcoming or ongoing crisis event. The evaluation of the new integrated functionalities, namely the risk maps, the novel risk assessment algorithm, which rely on the exploitation of the local information coming from the citizens' and first responders' mobile application via the appropriate incident reports, achieve reliable performance and meet the end-users (authorities) requirements as they have set for the flood pilot. However, further work is planned for the third iteration of the project including: (i) the elaboration of the risk assessment algorithm by exploiting the data from other heterogeneous resources (social media, images, videos etc.), (ii) clustering the incident reports based on their temporal/spatial characteristics and

then the estimation the severity of the crisis for each cluster, (iii) further analysis the socio-economic data, exposure and vulnerability metrics gained from risk maps and coupled with the aim of ex-ante risk assessments.

ASR component produced sufficient transcription results, but it will further benefit from the inclusion of advanced denoising techniques that address fragmented input as well as the addition of automatic language identification.

Drones platform and the related analysis module performed well by successfully detecting the target objects. The overall time needed for the analysis of drone footage and communication with the drones platform was adequate, since the drone transmitted limited amount of information (an image per second). Nevertheless, in future flights, the use of video files may increase delays. Until the final prototype, the code is planned to be optimized, with respect to computational speed, by speeding up each analysis step, where possible, and by examining the possibility of multithreading, or multi-processing using additional HW and the deployment on different environments. The drones platform shall evolve towards sending video files in real-time.

In general, the visual analysis components performed as expected. The crucial task of emergency classification was carried out well. Regarding the object detection and tracking tasks the components performed smoothly. Moreover, the extension of the detection and tracking functionality, as it is planned on the agenda, to include animal detection and people in wheelchairs will further strengthen its effectiveness in the disaster risk reduction domain. Localization algorithms will be further improved by retraining the algorithms following additional data acquisition plans, with the focus on the fire texture localization domain while preparing for the final pilot.

The evaluation of the KB - KBS duo concluded that its performance was more than adequate during the simulation of a real-life crisis scenario. The durations of incoming message processing presented a normalized behavior, revealing no efficiency issues for the scale of this scenario. However, a performance decline should be anticipated for a much greater number of incident reports. This could be tackled, provided the appropriate dataset, via the assignment of more resources to the software and the application of targeted code optimizations.

Text analysis performed well with the limited set of messages that were part of the pilot. However, many of its internal components are not yet used by the overall system, mostly due to being added very recently to the module. Thus, the results of geolocation were not considered by the crisis classification module, and neither were the results of WSD/EL used by the report generation module. The new wrap-up

summary functionality is still largely experimental and due to its many limitations it was not properly demonstrated during the second pilot. One of the main goals regarding this module is to improve this functionality so as to make it more central in the 3rd pilot.

The mobile application worked as expected and not serious technical problems were reported by the users during the pilot (see D2.6). The interference with other apps (which, in special cases, caused some problems during the first pilot) were solved and did not re-occur. All sent incident reports were timely delivered to the beAWARE platform. The position and the status of the first responder teams were continuously reported to the authorities. This allowed them a good overview of the available forces all the time. First responders timely received the tasks together with the position and description, so their mission was always clear to them. Based on the user feedback, improvements of the user interface, regarding overview and accessibility will be done for the 3rd prototype. Improvements in the synchronization of the mobile app with the platform are planned to further minimize the time in the mobile app interaction, even if the number of participants increases.

In the current version of the PSAP Map Visualization was improved and several new features were added providing informative data to the user. Nevertheless, in some cases, it was noted that the information provided was not very comprehensive, especially in cases where icons of incidents or tasks were overlapped.

To ensure the best possible user experience, we intend to add more visual information to the user on the map in a more detailed form of information about the critical data of events on the map such as status and more. The data we display in a grid table below the map will be more informative and a set of rules will be defined to trigger notifications to the user in case of tasks that did not finish at the specified time, severity of task/incident changed and more.

In addition, we intend to show the radius of the population being alerted on the map with a circle and a specific colour indicating the severity of the alerts.

Finally, experts on visual design will be consulted and every effort will be made to incorporate more user-friendly features, icons and navigation tools.

Still, the more critical feedback was very constructive and led to very helpful and important insights on how improve during the last development cycle. The outcomes of the 2nd prototype evaluation presented in this document will be the reference point to address the technical development of the platform towards the final version.

5 Bibliography

Haerder, T., & Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM computing surveys (CSUR) 15.4*, pp. 287-317.

Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007).*

Kontopoulos, E. M. (n.d.). Ontology-based Representation of Crisis Management Procedures for Climate Events. *1st International Workshop on Intelligent Crisis Management Technologies for Climate Events (ICMT 2018), colocated with the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2018).* Rochester NY, USA.

Mille, S., Carlini, R., Latorre, I., & Wanner, L. (2017). UPF at EPE 2017: Transduction-based Deep Analysis. *Shared Task on Extrinsic Parser Evaluation (EPE 2017).*, (pp. 80-88).

6 Appendix

6.1 Appendix A: Annotation guidelines for the creation of the WSD/EL dataset

1. Only annotate text spans consisting of a single or multiple consecutive words, e.g. do not annotate "Blue iced juice" as "blue juice", even if there is a synset for it.
2. If a suitable meaning is found for a multiword, do not annotate any meanings for parts of the multiword, e.g. if a synset is found for "emergency break", do not annotate "emergency" nor "break".
3. Do not annotate parts of a multiword NE even if no synset is found for it, e.g. do not annotate "Ponte" nor "Angeli" even if there is no synset for "Ponte Angeli".
4. Given a text span s and a set of synsets M returned by BabelNet after looking up s:
    - As a rule of thumb, annotate s with the any suitable meanings in M.
    - Multiple synsets are allowed if they are all judged to be correct
    - Words in s can be replaced by lemmas if necessary, e.g. look up "euro" instead of "euros".
    - If a NE synset is not in M but can be easily and unequivocally inferred from the context, annotate s with it, e.g "cruise liner" may be inferred to refer to the specific ship "Costa Concordia" in addition to the generic concept "cruise liner".
5. Do not attempt to annotate complex metaphors or inferred concepts not in M, e.g. do not try to guess if "the best feeling" refers to a specific emotion such as love, do not annotate "losing side" with "loser"
6. Words with multiple POS interpretations -> annotate correct meanings belonging to any of possible POS, e.g annotate verbal and adjectival synsets for words like "found", "convicted", annotate adjectival and adverbial synsets for "fast"
7. Annotate meanings for quantities, units, currencies, time periods, and percentages, e.g in "10% of 100€" annotate "10", "%", "100" and "€"
8. Annotate the following types of words only:
    - Nouns: Yes
    - Main verbs: Yes
    - Adverbs: Yes
    - Adjectives: Yes
    - Determiners and pronouns:
        - Quantifier: Yes (few, fewer, little, many, much, more, most, some, any)
        - Number: Yes
        - Article: No (a/an, the)
        - Demonstrative: No (this, that, these, those)
        - Possessive: No (my, your, his, her, its, our, their, x's)

    - Relative and interrogative: No (who, whom, whose, which, what, that)
    - Personal: No (I, me, she, they)
    - Reflexive and reciprocal: No (himself, each other)
    - Indefinite: No (one, other, another, no one, anybody, nothing, everything, something, someone, whatever, whoever, none, all, both, either, such, each, etc.)
  - Auxiliary verbs:
    - Copula: No
    - Modal: No
    - Tense: No
    - Aspect: No
  - Idioms, temporal expressions: No (Just as, In the light of)
  - Conjunctions: No
  - Non-governed prepositions: No (in 2005)
9. Annotate using the format synset_id0-"annotated words" or synset_id0|synset_id1|...-"annotated words" if multiple correct meanings are found

6.2 Appendix B: Evaluations Results of Water Level estimation through VRS

Table 19: Evaluations Results of Water Level estimation through VRS

Video timestamp	Measured Water Level	Estimated Water Level	Percent Error (%)
09:19:51	3.45	3.45	0
09:47:36	3.54	3.49	1
10:14:23	3.62	3.62	0
10:40:47	3.77	3.75	1
11:07:04	3.86	3.87	0
11:33:24	3.97	4	1
11:54:59	4.07	4	2
12:16:13	4.11	4.08	1
12:37:53	4.15	5.01	21
13:00:15	4.18	4.04	3
13:21:55	4.15	4.08	2
13:43:41	4.11	4	3
14:05:14	4.09	4	2
14:26:49	4.01	3.92	2
14:48:14	3.98	3.87	3
15:10:07	3.92	3.83	2
15:31:34	3.9	3.79	3
15:52:59	3.84	3.75	2
16:14:34	3.77	3.7	2

16:36:29	3.74	3.66	2
16:58:14	3.7	3.62	2
17:19:25	3.64	3.53	3
17:40:39	3.61	3.49	3
18:02:27	3.56	3.49	2
19:19:38	3.34	4.08	22
20:08:58	3.23	3.15	2
21:06:43	3.09	3.07	1
22:10:27	2.96	2.94	1
23:17:55	2.81	3.2	14
00:27:07	2.7	2.69	0
01:35:12	2.56	2.56	0
02:44:16	2.46	2.44	1
03:52:23	2.33	2.39	3
05:01:46	2.24	2.31	3
06:12:16	2.15	4.08	90
06:53:46	2.11	5.1	142
07:14:53	2.09	2.22	6
07:36:17	2.07	2.14	3
07:57:46	2.06	2.14	4
Overall Average			**9.10**