Evaluating FLOODIS: Mobile Sensing for a Flood Emergency Service in the Cloud

Claudio Rossi*, Alfredo Favenza*, Francesco Scullino*, Vito Macchia, Giovanni Luca Spoto, Fabrizio Dominci
*Microsoft Innovation Center / Istituto Superiore Mario Boella
Torino Italy
{surname}@ismb.it

Abstract—**Despite the growing development of space-based systems aimed at monitoring and studying natural hazards, floods continue to harm humankind worldwide, causing enormous human and economic losses. In a view of improving the timeliness of existing flood emergency systems, we propose FLOODIS: a novel service that exploits existing space technologies together with mobile sensing and state-of-the-art cloud computing platforms in order to provide a fast, flexible and scalable flood emergency system. We evaluate the performance of our service oriented cloud-based architecture with a real prototype, achieving real-time performances at scale.**

Keywords-component; cloud, performance evaluation, flood, emergency, mobile, service, 3-tier architecture

I. Introduction

The periodical occurrence of flood events in urban and suburban areas represents nowadays a big issue for humankind. Flood events, especially in -but not limited to- European countries caused in recent years many human and economic losses, with heavy impacts in densely populated areas. Flood consequences are exacerbated by urban sprawl and climate change, especially in areas close to creeks and rivers. This situation often translates in flashing floods that are very difficult to predict in advance and hard to manage with respect to population alerting, operational planning, and flood evolution prediction. According to [1], between 1980 and 2008 there have been around 2800 floods worldwide, with almost 280k people killed and about 400 billion dollars of economic impact.

These facts point to the need of improving the timeliness and the intelligence of flood emergency systems to help public administrations manage flood events. Indeed, real-time advanced monitoring and forecasting services are becoming mandatory to better address and mitigate crises arising before, during and after heavy flooding.

Nowadays, on field flood emergency systems principally depend on the collection and dissemination of Earth Observations (EO): a technology that features remote sensing, image processing and spatial analysis to provide geographic information from satellite imagery. In terms of crisis management and monitoring of natural disasters, EO is one of the leading technologies, and it allows capturing important measurements of the hazard, both during and after the event, thus providing various analysis opportunities. The European Copernicus Programme for Earth Observation, "Copernicus EMS" [2] (formerly known as GIO-EMS, GMES Initial Operations Emergency Management Service) provides geographical mapping services of natural disasters, with accurate geospatial information derived from satellite observations and complemented with in situ or open data sources, whenever available. Copernicus satellites support a wide range of applications and they can collect many measurables for different natural disasters, also providing detailed Digital Elevation Models (DEM) [3] of the earth. Indeed, Copernicus EMS represents a very powerful technology for land monitoring. However, the real potential of this system is currently limited by the few number of Copernicus operational satellites, counting five in orbit satellites (Sentinel 1-5). Overall, they can provide actual data of a specific area, together with the related delineation maps, with an average frequency in the order of hours. Due to both administrative operations required by the activation procedures and technological setups, some days can pass between the occurrence of a flood and its first mapping. This delay limits the responsiveness of operations during fast occurring emergency events, like flash floods, posing a big issue in providing fast and reliable flood information in near real-time. We stress that due to lacking of real-time flood emergency systems, prompt alerting and actuation of evacuation procedures represents a huge challenge for Civil Protections (CPs) and Disaster Management Centers (DMCs), who struggle to avoid causalities despite of the huge efforts of their emergency teams.

To overcome the aforementioned limitations we propose FLOODIS: a novel service that provides a faster, flexible and scalable flood emergency system. We leverage on cloud solutions [4] and on mobile sensing by letting users report on-ground flood status through their mobile devices (smartphones, tablets). A user Report contains a short description, a photo, and an indication of the water level[1]. We integrate these geolocated Reports with EOs in order to provide a Decision Support System (DSS) for public administrations, targeting early flood notification for citizens as well as for emergency teams. We also enhance flood extent map updates and forecasts by including user Reports, both form citizens and professionals operating in the field, in a novel flood forecast model. Indeed, we couple user-generated Reports with data from existing

[1] The water level can be binary (to indicate the presence of water) or an actual water level estimation, according to the user profile

emergency systems in a view of creating a comprehensive flood emergency system.

Furthermore, to increase the accuracy of the flood forecast model, we validate and augment the position of user-generated Reports by implementing a cloud-based service on top of the EGNOS Data Access Service (EDAS) [5], which provides differential corrections for GPS, achieving both position integrity [6] – in the form of the so-called protection levels – as well as increased accuracy.

In short, FLOODIS aims to achieve the following goals:

- Provide a flood alert and information service for emergency response teams and affected citizens, leveraging on existing space assets;
- Increase social awareness and citizen involvement about flood emergencies with mobile sensing, i.e., allowing users to provide real-time flood Reports;
- Increase the accuracy and the timeliness of flood extent and forecast maps by integrating user-generated information into a novel flood forecast model;
- Ultimately, reduce the social impact of floods in terms of human and economic losses.

FLOODIS can be defined as an end-to-end system focused at the integration and convergence of space technologies, i.e., Copernicus EMS, European Global Navigation Satellite Systems (EGNSS), European Flood Awareness System (EFAS) [7] with ICT technologies and novel paradigms, i.e. cloud computing and mobile sensing.

The contribution of this paper are:

- The design of the FLOODIS system and the implementation a FLOODIS prototype featuring a Service Oriented Architecture (SOA);
- A performance evaluation of the FLOODIS cloud-based deployment using real mobile devices and a commercial PaaS cloud service.

Our results demonstrate that the proposed architecture is able to handle in real-time big concurrent user-generated content, enabling the management of critical and highly dynamic emergency scenarios such as flood events.

II. Space Assets

FLOODIS combine both Earth Observation (EO) and GNSS (Galileo/EGNOS) technologies to provide an enhanced flood emergency service. Specifically, we use Copernicus EMS flood map products, the EGNOS Signal in Space (SIS), and the EDAS service. The latter is used to get EGNOS data in real time through Internet, obtaining EGNOS benefits without relying on the SIS that may suffer from poor availability in dense urban environments (urban canyons). In the following subsections, we give a brief overview of both Copernicus EMS and of EGNOS/EDAS in a view of allowing the reader to understand the importance of these providers in the FLOODIS system.

A. Copernicus EMS

Copernicus EMS mapping products provide accurate geospatial information derived from satellite sensing to all actors involved in the management of natural disasters, man-made emergency situations, and humanitarian crises. There are many emergency domains, each one containing different product types, including mapping, monitoring, and high resolution impact assessment. In case of floods, meteorological satellites can be used to get weather forecasts, while other satellites can provide soil moisture mapping and detailed DEM of areas at risk. These information can be useful for flood simulations and forecasts. Moreover, satellite images can show the extent of the inundation and its evolution. In the aftermath, satellites images support the creation of risk maps showing the flood damages, with a proper classification of the inundated lands. As soon as an emergency event is activated, Copernicus EMS triggers the production of maps to support emergency management. The Copernicus EMS service can be used by authorized, associated and general public users. Authorized and associated users can trigger activations, whereas public users are not authorized to trigger the service, but they can be informed of an activation request through the web portal. Note that FLOODIS is seen as a public user and hence it is timely informed about flood activations. Depending on the level of urgency, each disaster category contains two different service modes: Rush and Non-Rush mode.

The Rush mode service consists of on-demand and fast (within few days) provision of geospatial information in order to support emergency management activities immediately after an emergency event. This service type involves a timely acquisition, processing and analysis of satellite imagery together with other geospatial raster and vector data from third party data, whenever available. The outputs are standardized following a set of parameters that can be chosen by the user when requesting the service. Overall, the list of activations are sorted according to the disaster type and structured according to the activation code, which is made by the location, the date and the country of a specific event. For each event it is also possible to create a GeoRSS feed to get informed about updates. FLOODIS automatically subscribes to any activation related to floods in order to get updates in real-time. The Rush mode service provides three different map products:

- Reference maps – provide general overview about the affected area of interest;
- Delineation maps – show full flood extent for a specific time period. They are provided at different scale, one or more detailed views plus an overview view;
- Grading maps – show magnitude and damage grade caused by flood on infrastructure.

The Non-Rush mode service consists of the on-demand provision of geospatial information whenever the urgency of such information is low. Hence, this service type supports emergency management activities not related to immediate needs. This service addresses prevention, preparedness, disaster risk reduction or recovery phases. Note that in the latter case the product may be delivered in weeks/months. Given the wide variability of situations, the user may request non-rush mode products in two ways:

- Choosing from a pre-defined set of detailed topographic features (in particular regarding infrastructures) and disaster risk information (hazard, exposure, risk). This allows having a standard base structure;
- Describing in free text the desired product type. This allows users to include a wide range of optional information layers, depending on their needs.

The maps of the Non-rush mode service can be divided in the following categories:

- Reference maps – provide general overview about the affected area of interest;
- Pre-disaster situation maps – thematic map before the flood. It may help regional planning and identification of damage;
- Post-disaster situation maps – thematic map after the flood. It may help regional planning and identification of damage, and supports change detection analysis and vulnerability assessment.

The delivery of Non-Rush mode products can last up to 8 weeks. We summarize in Table 1 Copernicus service models and main map type produced. Figure 1 and Figure 2 shows a detail and overview delineation map in Rush mode, respectively. Figure 3 and Figure 4 show a reference and post-disaster map in Non-Rush mode, respectively.

Within FLOODIS, we are mainly interested in the Delineation maps, which provide an assessment of the flood event extent (and of its evolution, if requested). The basic data is a polygon that identifies the boundary of the flood extension at a given point in time. Remote Sensing and GIS techniques can be used very effectively to determine current flooding and to separate flood from reference water levels.

B. *EGNOS/EDAS*

The European Geostationary Navigation Overlay Service (EGNOS) is the first pan-European satellite navigation system. It augments the satellite navigation systems, making them suitable for safety critical applications. EGNOS provides corrections and integrity to GPS signal over a broad area centered in Europe. It is composed by two segments: the space segment and the ground segment. The space segment comprises 3 geostationary (GEO) satellites broadcasting corrections and integration information for GPS satellites in L1 frequency.

The ground segment comprises a network of different stations and facilities. They collect measurements from GPS satellites and process them. Based on that, several sets of corrections are transmitted to EGNOS satellites, which are broadcasted to users. These corrections refer to: clock corrections for each GPS satellite in view, ephemeris corrections, ionospheric errors. The EGNOS system can also warn the users in case anomalies in GPS data, which are detected in a very short timeframe.

Table 1: Copernicus EMS service modes and map

Type of disaster	FLOODS					
Service mode	Rush mode			Non-Rush mode		
Map products	Reference	Delineation	Grading	Reference	Pre-disaster	Post-disaster

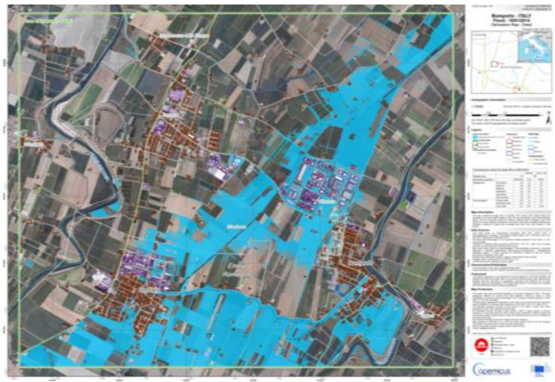


Figure 1: Detailed delineation map with current flood extent in the Bomporto region (Rush mode)

Figure 2: Overview delineation map with current flood extent in the Bomporto region (Rush mode)

Figure 3. Non-rush mode service of a reference map of Salzburg

Figure 4. Non-rush mode post-disaster map (detail)

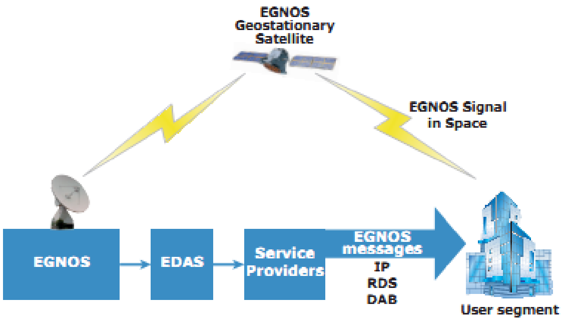


Figure 5. EDAS high level architecture

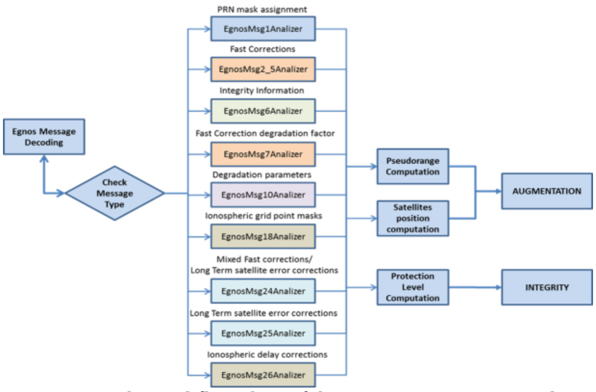
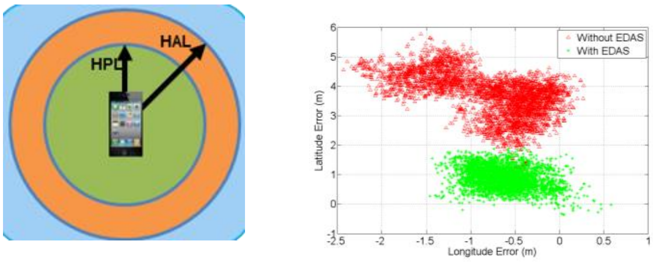


Figure 6. High Level flow chart of the EGNOS Message Decoding Component



(a) (b)

Figure 7: EGNOS/EDAS Horizontal Protection Level

Users can benefit from the EGNOS system by receiving the signals broadcasted by the GEO satellites, also called the Signal-In Space (SIS). However, due to obstructions caused by tall buildings, the user may not be able to receive the SIS, which happens very frequently in dense urban scenario. EDAS [5] provides an opportunity to deliver EGNOS data by ground-based network, including the cellular network. The high level architecture of EGNOS/EDAS is shown in Figure 5, which sketches the different delivery scheme of the system.

The EDAS service can deliver two main types of data in real time:

- Augmentation information (including integrity), as normally received by users via the EGNOS geostationary satellites;
- Raw data collected by the EGNOS monitoring reference network.

The EDAS service type is selected by the user as first step. Next, upon reception of the EDAS data, content parsing is the first operation that needs to be performed. The parsing is demanded to a component called the EGNOS Message Decoding Component, which also decodes the messages and extract all EGNOS messages based on the 6-bit message type identifier. Due to the limited size of the type identifier, only 64 types of messages are possible.

We report in Figure 6 the flow chart of the operations performed by the EGNOS Message Decoding Component, from the decoding phase to the final output, while we sketch in Figure 7a the Horizontal Protection Level (HPL) computed through EDAS, and the Horizontal Alert Limit (HAL). The HAL is the maximum limit for the HPL, after which the position obtained cannot be considered valid for navigation services. The HAL was initially conceived for plane navigation applications. Figure 7b shows the position errors with and without EDAS, which we obtain collecting one sample every second for several hours at a single known position. On average, the EDAS augmentation algorithm reduces by 0.18m and 2.78m the longitude and latitude error, respectively. Since the HAL is not standardized for land applications and given the results obtained in past GPS/EGNOS performance evaluations, we set the HAL=10m.

III. System Architecture and Implementation

A. *System Architecture*

A flood emergency service can be subject to usage peaks at certain times, i.e., during the flood event, while being completely inactive otherwise. To cope with usage bursts, we implement a cloud-based Service Oriented Architecture (SOA) breaking down FLOODIS in different sub-systems in a view of allowing each one of them to be deployed independently, automatically scaling according to the usage. The main components of FLOODIS are:

- A GEO Gateway that interfaces with all existing European emergency systems like Copernicus EMS and EFAS [7] in order to provide both flood extent and flood forecast maps. It implements a novel flood forecast model that is able to include user-generated content into, enhancing the accuracy of flood forecasts;
- A multi-platform Mobile Application (MA) that (i) sends real-time geolocalized flood user Reports, (ii) receives terrain maps together with additional layers for: street, terrain, flood extent, flood forecast, and nearby user Reports, (iii) receives flood related alerts from the authorities via FLOODIS;

- An augmentation service based on EDAS that receives GNSS raw data from the mobile application and computes an augmented and validated GNSS position, featuring increased accuracy and integrity;
- A Service Layer (SL) that acts as the FLOODIS centralized service provider, implementing (i) web services to receive/provide geolocalized flood Reports from/to the mobile application; (ii) a service that provides all flood Reports of a given area back to the GEO Gateway (for flood forecast updates), (iii) a web site that acts as the Decision Support System (DSS) for CPs and Disaster Management Centers (DMCs), enabling to visualize all flood information collected by FLOODIS and to send flood alerts to mobile users in geographical proximity.

In addition to the information shown by the MA, the DSS running on the SL enables to see historical flood extent map and the current position of civil protection agent in the field. Figure 8 shows the architectural scheme of FLOODIS, highlighting the four sub-systems and their interactions.

B. *FLOODIS Implementation*

1) GEO Gateway

The core functionality of the GEO Gateway is to ingest, process, store and distribute flood related map layers. Flood delineation and flood forecast layers are the main inputs to FLOODIS and therefore to the GEO Gateway. Delineation maps in the form of ESRI shapefiles are downloaded from the Copernicus EMS website together with metadata acquired from the EMS GeoRSS feed. Within FLOODIS, this process is triggered when a flood event is activated in Copernicus, which happens rarely, i.e., in the unlikely event of a flood. Subsequently, GIS processing is conducted to transform the shapefiles and store the information in a spatial database, that we implement using a PostgreSQL with the PostGIS extension. The provision of the flood delineation maps to the final users is then achieved by using the open-source map server software GeoServer, that implements Open Geospatial Consortium (OGC) web services. Within FLOODIS we select the web map tiled service (WMTS) as it provides for caching. Thus, the tiles that are requested are cached, greatly reducing the load on the spatial database. Both, flood delineation and flood forecast maps are delivered with WMTS, and they are requested by the FLOODIS Service Layer as well by the MAs. The use of OGC standards enables the Geo Gateway flood layers to be requested by any WMTS compatible client besides FLOODIS, which demonstrates the flexibility of Service Oriented Architectures. We report in Figure 9a the architecture of the Geo Gateway, in which we specify all the system sub-components, the main standards adopted and information flows.

Clearly, the bottleneck of the GEO Gateway is the GeoServer, as it may need to serve a big amount of clients in case of a massive flood event. In order to guarantee performance and scalability we deploy the GeoServer with a Microsoft Azure [8] Cloud Service, that can scale according to the clients demand.

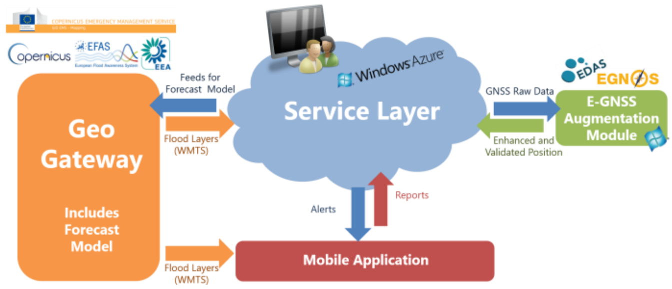


Figure 8: FLOODIS architecture

2) Flood Forecast Service

We realize that it is not possible to always depend on EO data alone and therefore we decide to look for practical alternatives that could provide flood disaster managers and responders with vital flood related information based on non-EO sources. The goal of the FLOODIS Flood Forecast Backend (F3B) service from the beginning was to use as much as possible already available data and information and for this reason many of its capabilities have been developed as Copernicus downstream services using Copernicus reference data. The inputs include post-processed river flow data from EFAS [7] Sensor Observation Service (SOS), the EU-DEM [3], CORINE [9], Open Street Map (OSM), as well as user supplied field data about flood extent location. The service also integrates flood modeling based on the LISFLOOD-FP [10,11] model as well as simple planar models (0D solutions). The latter consist of a linear interpolation of the water level to derive water surface and then inundated area. 0D solutions are usually used as a lower bound to evaluate more advanced solutions [12]. The outputs produced are current and forecast vector layers of model flood extent that are pushed to the users for interpretation and application for now-casting and decision support.

3) Service Layer and Augmentation Module

In order to implement the Service Layer (SL), we use the well-known 3-tier programming pattern. This pattern divides the program into three logically and architecturally different layers: Data Access (DA), the Business Logic (BL) and the Presentation (P). The DA deals with the interaction with the database, the BL provides for common algorithms and processing, while P implements the User Interface together with the client-side scripts and validation rules. In Figure 9b we report the 3-tier architecture implemented, together with the main technologies used. We adopt NHibernate for the Object Relation Mapper (ORM), and SQL Azure as database. We deploy the service layer using the Platform as a Service (PaaS) approach, using one Microsoft Azure Cloud service for the Service Layer, one Azure SQL database (Basic deploy) for user Reports textual information and positions, and one Azure Storage for photos[2] and logs. We implement all web services using the REST architecture, the JSON data format, and the .NET WEB API 2.0 framework. The SL host also a web-site for user registration and all OAuth2.0 services related to user login from mobile devices. To implement both registration, login, and user authorization, we rely on the .NET Identity

[2] each report can contain up to one photo

framework 2.1.0. The web site DSS hosted by the SL is implemented using the Model View Controller (MVC) pattern, HTML5/CSS3, and the .NET framework 4.5.1.

We separate the Augmentation Module (AM) from the SL because not all user-devices can provide GNSS raw data. In such case, the user Reports are sent with standard positioning, i.e., latitude plus longitude, without triggering the augmentation procedure. We apply the same 3-tier patter also for the augmentation module, changing the data access to work with an Azure Table Storage [13], which is a NOSQL database. In this case, the data is easily mapped into a key-value structure that enables us to exploit the greater speed of NOSQL solutions.

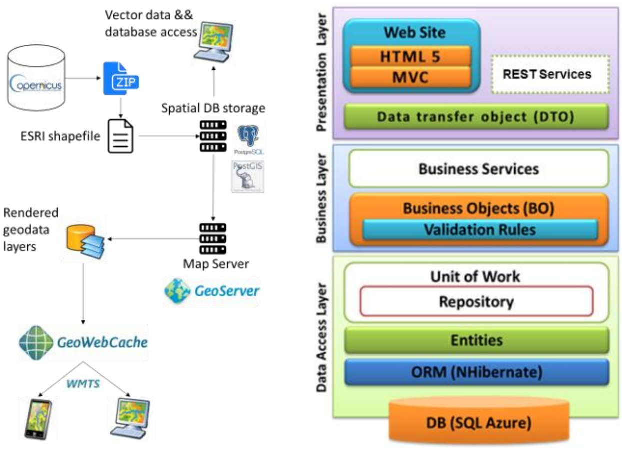
We implement the EGNOS/EDAS correction algorithm as outlined in Section II.B, following the RTCM specification [14] as described in our preliminary work [15].

Our implementation allows the receiver to perform the correction, achieving the so-called position augmentation, compute the position integrity, or both. However, generally there are two modes of operation:

- Correction-only mode, in which EGNOS data are used to improve the accuracy of pseudoranges and satellites positions. Thus, the GNSS receiver will compute a more accurate final Position Velocity Time (PVT) vector;
- Corrections-plus-integrity mode, the system uses the information from EGNOS also to calculate Horizontal and Vertical Protection Levels.

Figure 10 shows the main building block of our implementation, together with the information flow from the initial service request to the final PVT and PLs computation. Note that the algorithm needs position raw data from the GNSS receiver.

We implement both SL and AM using .NET framework version 4.5.



(a) (b)

Figure 9: Geo Gateway architecture (a) and Service Layer architecture (b)

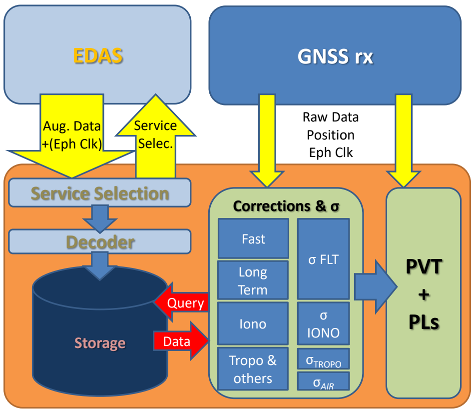
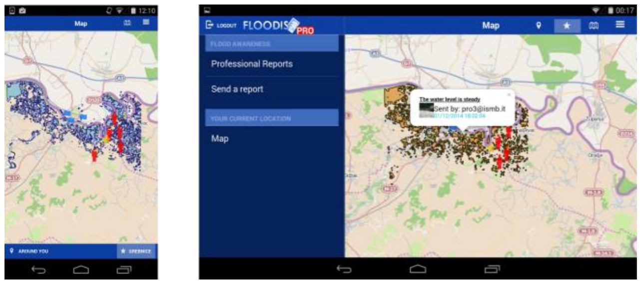


Figure 10: Mid-level block diagram describing augmentation algorithm structure and data flow



(a) (b)

Figure 11: FLOODIS mobile application running on a smartphone (a) and on a tablet (b)

C. *Mobile Application*

The Mobile Application (MA) is the only component that does not need to scale, because it runs on end-user devices. However, it needs to achieve wide adoption in order to maximize the mobile sensing benefit resulting from a massive in-field data collection from users. To maximize adoption, we implement the mobile application using Cordova: a platform for building hybrid mobile applications using HTML, CSS and JavaScript. In this way, we can build the FLOODIS MA for all major mobile systems, including Windows, Windows Phone, Android and iOS. We use the Leaflet library for what concern GEO Gateway flood map representation, which are delivered using the WMTS standard.

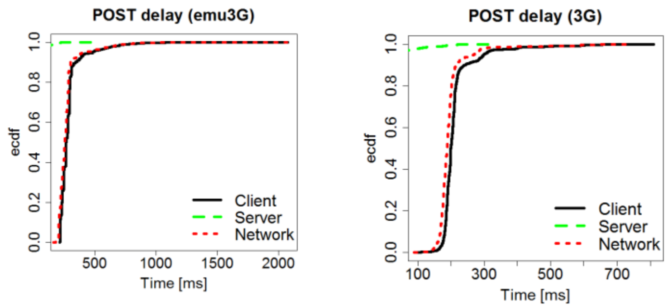
GNSS chipset drivers of mass-market smartphone and tablets do not expose position raw data to the operating system. In a view of augmenting the GNSS position, we use a commercial external GNSS receiver equipped with a SIRF chipset, which is able to provide position raw data in SIRF binary format. We implement the procedure to query the SIRF chipset together with the parsing of the binary data in order to extract all the raw data we need for augmentation. For further details on position raw data please refer to [16, 17].

The MA provides different capabilities according to the user: logged-in users can contribute with flood Reports, while all other users can see only flood extent maps, if any. CP users can also see other CP users nearby as well as flood forecast maps.

Figure 11 contains two screenshots of the FLOODIS MA, showing a map (street view) with additional layers: both a flood extent map as well as user generated Reports and civil protection user positions. The latter are simulated while the flood extent map is a real delineation map obtained by the Geo Gateway. Note that we use OSM as our map provider.

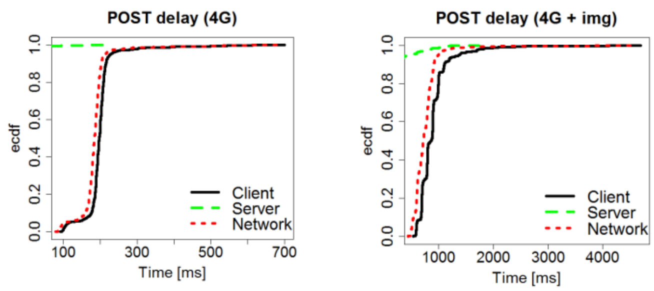
IV. PERFORMANCE EVALUATION

From the FLOODIS architecture, two bottlenecks can be identified. The first one is the GEO Server included in the Geo Gateway, because it has to provide flood extent and forecast maps both to the SL (for visualization through the DSS web site) and to the MAs. While web site users are limited to employees of CPs and DMCs, MA users grow with adoption and usage, potentially reaching a very big volume, eventually posing performance challenges. The same consideration holds for the SL, which has to ingest user Reports. Since we are focused on mobile-sensing, which is one of the main novelties in flood management services, we present the performance evaluation of our testbed from the user perspective, using the real MA and the Azure cloud deployment of the SL described in Subsection II.B. Note that the flood forecast algorithm is not a bottleneck, because it is computed in background every few hours if there is at least one ongoing flood in Europe. Throughout all the evaluation, we use as SL the Cloud deployment explained in Subsection II.B.3, enabling the Azure autoscale function up to a maximum of 5 Small instance [14].



(a) (b)

Figure 12: ecdf for delay contributions for the POST Report service without image, used by the Chrome device emulator with 750kbps 3G (a) and with Android Nexus 7 device with a real 3G connection (b).



(a) (b)

Figure 13: ecdf for delay contributions for the POST Report service without (a) and with (b) attached image, called by and Android Nexus 7 device connected with 4G.

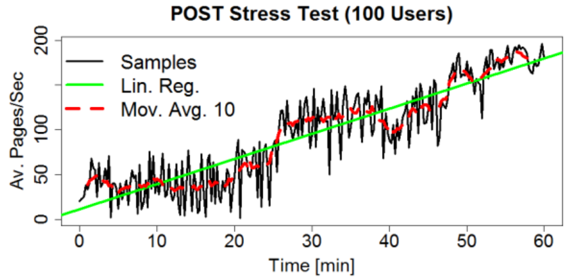


Figure 14: Average Pages per Second achieved with a stress test of the POST Report (with image) service with 100 parallel users. The Azure Cloud Service is deployed with 1 up to 5 Small instances, with autoscaling in steps of 2, and with Azure SQL database (Basic).

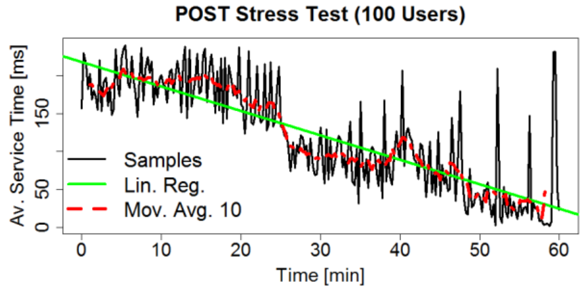


Figure 15: Average Service Time achieved with a stress test of the POST Report (with image) service with 100 parallel users. The Azure Cloud Service is deployed with 1 up to 5 Small instances, with autoscaling in steps of 2, and with Azure SQL database (Basic).

We select Small Azure instances and a Basic database because we want to minimize the operational costs, to provide FLOODIS for PAs at a minimum economic burden. We test the POST Report service, having a HTTP request payload of 230B and 590B without and with an image, respectively. The HTTP response payload is negligible as it contains only the HTTP response code. We pre-load our database with 500k user Reports.

As performance indicators, we select different delay contributions. The client delay, defined as the delay between the HTTP request and the response measured at the device (client). The network delay, defined as the sum of the network delay of the HTTP request plus the network delay of the HTTP response; and the server delay, which is the delay between the reception of the HTTP request and the dispatch of the HTTP response at the Cloud.

First, we measure the aforementioned delay contributions with one client and without attaching an image to the Report, varying the client type and the network connection. Figure 12a shows the results obtained with the MA deployed as web application, and run using the Chrome device emulator set with a Nexus7 device and a 3G connection throttled at 750kbps. With this setup, we obtain a median delay of 267ms, 255ms, and 11ms for client, network, and server, respectively. Clearly, in this case the delay is dominated by the network. We perform the same test using a real Android Nexus 7 device connected with 3G, obtaining slightly better results, namely a median delay of 199ms, 188ms, and 11ms for client, network, and server, respectively (Figure 12b). This improvement is

978-1-61284-732-0/11/$26.00 ©2010 IEEE

motivated by the greater speed of the 3G connection[3]. We enable the 4G connection on the Nexus device and we perform the POST Report with and without image, obtaining the results shown in Figure 13. Despite the greater speed of the 4G connection, without picture the median delay contributions are very similar to the 3G case, specifically 197ms, 184ms, and 11ms, for client, network, and server, respectively. This is due to the little payload (230B) that due to the channel allocation scheme of the 4G network cannot benefit for the greater capacity[4]. Including in each Report an image of size 340KB, the median delay increases of almost one order of magnitude, specifically we obtain 869ms, 721ms, 127ms, for client, network, and server, respectively. Given our Azure deployment, the aforementioned tests, summarized in should be considered as a best case, because there is only one device interacting with the Cloud.

In order to understand the performance of our Cloud deployment at scale, we use the Azure Stress Test Tool provided by Visual Studio Online. We simulate 100 parallel users – which is a very high number considering our application - submitting a flood Report with image for 1 hour, letting each user continuously fires a Report one at a time. We plot in Figure 14 and Figure 15 the average pages (requests) per second and the average service time in function of time, respectively. It is clearly visible that the Azure Cloud Service performs two scaling operations around minute 25 and minute 40, increasing the instances to 3 and 5, respectively. We note that the scaling time between two configurations is around 20min, despite the Azure setting at 5min, and that performances are less consistent with the number of active instances. However, the average performances are good for our application, confirming the validity of our deployment for a commercial deployment.

V. CONCLUSIONS AND FUTURE WORKS

We designed, implemented and evaluated FLOODIS: a novel flood emergency service exploiting mobile sensing and cloud architecture to improve management and operations in case of floods. We will demonstrate the FLOODIS prototype during in-field trials that will be held together with our stakeholders: the Civil Protection of the Veneto Region in Italy and the Albania country. Their feedback will serve to further refine and improve the FLOODIS service for Civil Protection Agencies. Furthermore, we will realize a full business case for FLOODIS, with the aim of providing a more precise quantification of the FLOODIS roll out in Europe and beyond.

Table 2: Median delay contribution of single device tests

Connection type	*Median Delay [ms]*		
	Client	*Network*	*Server*
3G @750kbps	267	255	11
3G	199	188	11
4G	197	184	11
4G with image	869	721	127

[3] 3G at 5.9/2.2 Mbps averaged on 10 downlink/uplink speed test before starting the evaluation

[4] 4G at 12.4/6.2 Mbps averaged on 10 downlink/uplink speed test before starting the evaluation

REFERENCES

[1] Flood – data and statistics, http://www.preventionweb.net/english/hazards/statistics/?hid=62

[2] Copernicus, http://ec.europa.eu/enterprise/policies/space/copernicus/index_en.htm

[3] The Digital Elevation Model over Europe from the GMES RDA project (EU-DEM), http://www.eea.europa.eu/data-and-maps/data/eu-dem

[4] NIST, “The NIST Definition of Cloud Computing,” September 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[5] EC, “Data Access Service (EDAS) Service Definition Document (v2.0)”, 2013.

[6] Position Integrity, http://www.navipedia.net/index.php/Integrity

[7] EFAS, https://www.efas.eu/

[8] Microsoft, “Microsoft Azure Cloud Computing Platform,” http://www.microsoftazure.com.

[9] Corine Land Cover Classes, http://land.copernicus.eu/pan-european/corine-land-cover/clc-2012/view

[10] J. M. Van Der Knijffa, J. Younisa, A. P. J. De Rooa “LISFLOOD: a GIS - based distributed model for river basin scale water balance and flood simulation”, International Journal of Geographical Information Science, vol. 24, issue 2, 2010

[11] Bates, P.D . and De Roo, A.P.J.,. A simple raster-based model for floodplain inundation. Journal of Hydrology, no. 236, pg.54-77, year 2000.

[12] G. Di Baldassarre, “Floods in a Changing Climate: Inundation Modelling”, Cambridge University Press, 2002

[13] B. Calder, J. Wang, A. Ogus, and et al. Windows azure storage: a highly available cloud storage service with strong consistency. In SOSP '11, pages 143-157, 2011.

[14] RTCA DO-229D, 2006.

[15] A. Favenza, C. Rossi, M. Pasin, F.Dominici “A Cloud-based Approach to GNSS Augmentation for Navigation Services”, ACM/IEEE UCC, 2014

[16] GPS Interface Control Document ICD-GPS-200, Revision C. U.S. Air Force, 2013

[17] B. W. Parkinson, J. J. Spilker Jr., Global Positioning System: Theory and Applications, vol. 1. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1996.

978-1-61284-732-0/11/$26.00 ©2010 IEEE