Comput Syst Sci & Eng (2020) 1: 5–12

International Journal of
Computer Systems
Science & Engineering

An Internet of Things Platform for Air Station Remote Sensing and Smart Monitoring

David Corral-Plaza[1], Juan Boubeta-Puig[1], Guadalupe Ortiz[1], and Alfonso Garcia-de-Prado[2] *

[1] *Department of Computer Science and Engineering, University of Cadiz, Avda. de la Universidad de Cádiz nº 10, 11519, Puerto Real, Cádiz, Spain*
[2] *Department of Computer Technology and Architecture, University of Cadiz, Avda. de la Universidad de Cádiz nº 10, 11519, Puerto Real, Cádiz, Spain*

Air pollution is currently receiving more attention by international governments and organizations. Nevertheless, current systems for air quality monitoring lack essential requirements which are key in order to be effective concerning users' access to the information and efficient regarding real-time monitoring and notification. This paper presents an Internet of Things platform for air station remote sensing and smart monitoring that combines Big Data and Cloud Computing paradigms to process and correlate air pollutant concentrations coming from multiple remote stations, as well as to trigger automatic and personalized alerts when a health risk for their particular context is detected. This platform has been tested by analyzing the results of observing Andalusian, South of Spain, sensor network during a long period of time. The results show that this novel solution can help to reduce the impact of air pollution on human health since citizens are alerted in real time.

Keywords: Air quality, Environmental Monitoring, Remote sensing, Internet of Things.

1. INTRODUCTION

Year after year, the world is increasingly more conscious and worrying about air pollution and how it can affect citizens' lives. Among other consequences, air pollution can seriously affect their health, worsening certain illnesses or even causing death to specific risk groups [1]. On the one hand, a study made by WHO's International Agency for Research on Cancer (IARC) concluded that outdoor air pollution is carcinogenic to humans, especially associated with lung cancer [2]. On the other hand, the European Union estimated that pollution caused more than 400,000 premature deaths and preventable diseases in 2010, including respiratory conditions such as exacerbated cardiovascular problems and asthma [3]. Moreover, health impacts caused by air pollution involve external costs in order to repair a given situation or to avoid welfare losses [4]. For these reasons, the society is becoming more interested in this topic, paying extra attention to air quality. Indeed, air quality monitoring is a fundamental issue to be tackled by the whole society in general, and particularly by a representative sample of citizens such as children and elders, people doing physical exercise outside and those who have some types of lung disease.

Due to this worldwide concern, several systems for air quality monitoring have been created over the last years. However, most of them present limitations about providing recommended air quality notifications based on citizens' particular context [5]–[7]. Moreover, these systems provide the air quality-related information without focusing on the personal circumstances of each user.

The main current problem is that monitoring the environment is not enough; we need to guarantee that citizens can easily be made aware of this information, which must be kept up-to-date in real time. Furthermore, if we want the risk groups to be aware of their particular dangers and how they should act according to the different risk levels, we should provide them with the right-now information and vehemently encourage them to act upon it. However, this kind of personalized services can become complex in Internet of Things (IoT) applications [8] because of the large

* {david.corral, juan.boubeta, guadalupe.ortiz, alfonso.garciadeprado}@uca.es

amounts of data required to be analyzed in order to perform a suitable customization of such services.

Therefore, we can state that in order for air monitoring and notification systems to be effective concerning the use citizens make of such information, these systems have to satisfy the following requirements: (1) air quality information and alerts have to be updated in real time; (2) the information has to be actively provided to citizens in a clear and user-friendly way; (3) the information provided to users, in particular to risk groups, needs to be adapted to their specific characteristics and (4) the system should also take into account the type of activity the user is going to carry out and adapt notifications accordingly.

In order to tackle such challenges, this paper proposes an IoT platform for remote sensing and smart monitoring of air stations. This platform is based on FIWARE [9], a cloud platform recently promoted by the European Commission [10]. In the last years, the interest in IoT systems has grown significantly and therefore it is being developed by many companies [11]. The main advantages provided by FIWARE are as follows. FIWARE provides novel software components available through Application Programming Interfaces (APIs), known as Generic Enablers (GEs), that allow us to reduce the time needed for developing the whole solution, as well as increasing the scalability, modularity and flexibility of the solution [12]. This platform is based on a RESTful architecture, what permits the architecture components to talk to each other easily and in a decoupled way [13]. FIWARE is completely open source, i.e. its architecture, GEs, and services ecosystem are open to whoever wants to research on them. This platform also provides us with capabilities of Infrastructure as a Service (IaaS) based on OpenStack [9].

The FIWARE-based smart platform that we propose in this work has the ability to process and correlate big data coming from multiple air information sources and to notify users in real time when a health risk for their particular context is detected. In addition, a mobile application benefits from the platform since it allows users to register and receive air quality alerts depending on their situational context.

Thus, the main contributions of this paper are (1) the definition of a flexible and scalable platform designed for IoT real-time sensor data monitoring based on recognized standards as FIWARE and (2) a full implementation of the platform for air quality monitoring with the aim of preventing air-quality related citizen health risks through context-aware personalized real-time notification, which satisfies all the requirements previously stated.

The rest of the paper is organized as follows. Section 2 presents the background work. Section 3 describes the proposal through an application scenario. Section 4 presents our platform for air quality monitoring and notification. Section 5 explains the experimental validation conducted and the obtained results. Section 6 describes the related work. Finally, Section 7 highlights the conclusions of the proposal and the future work lines.

2. BACKGROUND

In this section, the relevant subject matters for the scope of this paper are introduced.

2.1 Complex Event Processing

Complex Event Processing (CEP) [14] is a cutting-edge technology that allow us to correlate and analyze huge amounts of data in real time. These data are processed as events, representing a change of state in the real world such as an increase in temperature, humidity or air quality.

The main aim of CEP is to promptly detect situations of interest by using event patterns, which describe the conditions to be satisfied. These situations depend on the application domain.

In particular, CEP can be of great utility in the scope of IoT, since a large amount of IoT sensors from which we need to make prompt decision requires a technology capable of performing streaming analysis on the data coming from the IoT sensors in real time and this is exactly what CEP provides us with. In the scope of air quality monitoring, we can detect in real time pollutant concentrations dangerous for the health of the citizen in general or a specific risk group in particular, preventing unwanted damages.

There are multiple CEP software solutions to analyze the data. In our FIWARE-based platform, we use Cepheus CEP, whose core works with Esper [15]. Esper is an open source CEP engine which allows us to process up to 500 000 events per second in a simple desktop machine. One of the main advantages of using Esper, against other alternatives, is the capability of adding new configurations in runtime, so it is not necessary to stop the whole platform to be able to add new situations of interest.

The situations of interest (event patterns) to be detected are implemented in Esper using the Event Processing Language (EPL), a SQL-standard language with extensions such as sequence and temporal operators. EPL provides the SELECT, FROM, WHERE, GROUP BY, HAVING and ORDER BY clauses well-known by SQL developers. In addition, EPL statements can specify data windows, based on certain conditions such as length and time. When a pattern is satisfied, it generates a new complex event automatically. The complex events can be used to feedback the CEP engine in order to detect new complex events based on the already detected ones. Additionally, a priority can be set to these patterns, what is very useful in order to define pattern hierarchies.

2.2 Situational Context

Situational context is defined as the characteristics of the user's surrounding environment and the user itself [16].

We consider situational context as the characteristics that define a person and her environment, as well as the interaction between them, at a given moment. Among the different elements that can define a person's situational context, we highlight: age, sex and diseases; weather and location; and the type of activity that the person is doing [17].

All situational contexts are different between them since each person is unique and her way of interacting with its surrounding environment is unique too.

For this reason, air quality does not affect all people equally, e.g., a bad air quality affects a person with asthma more than a person who does not have asthma. This way, we consider the situational context essential to customize the air quality notifications to be sent to people breathing under difficult circumstances in a specific moment.

3. APPLICATION SCENARIO

In this section, we describe the application scenario chosen to demonstrate the usefulness of our proposed IoT platform.

3.1 Air Sensor Network

The air sensor network for air quality surveillance and control used to test our platform is described as follows. This network is distributed throughout the Andalusian territory which has 87 268 Km2 (33 694 square miles) and an estimated population of 8.4 million people.

This network is composed by 91 stations, 81 for air quality and 10 for meteorological monitoring. Most of these stations belongs to the Andalusian government meanwhile the rest of them are owned by other public and private companies, but all of them are ruled and managed by the Andalusian government. The location of these monitoring stations is not random; they have been located in strategic places in order to be able to take representative measures of different scenarios, from green and healthy areas to busy avenues.

Each station has several sensors, depending on what it is important to monitor in the area where it is located. There are 17 different types of pollutant sensors and 8 different sensors for meteorological stations. Between the available pollutant sensors, we highlight: Particles Matter smaller than 10 or 2.5 microns (PM2.5 and PM10), Sulfur Dioxide (SO2), Nitrogen Dioxide (NO2), Carbon Monoxide (CO) and Ozone (O3). On the other hand, for meteorological monitoring we have available sensors for wind speed and direction, humidity, temperature and solar radiation, among others.

In our study, we are only going to use the previously mentioned most relevant air pollution sensors: PM2.5, PM10, CO, O3, NO2 and SO2; those are the values mainly taken into account by worldwide air quality indexes and, therefore, they are relevant for our study purpose.

For this reason, although the whole network is composed of 91 stations with measurements in near real-time (one measurement per sensor every 10 minutes), we only have made use of 61 stations, all of them for air quality monitoring.

The sensors located in these stations send the information collected for each pollutant to a main server by General Packet Radio Service (GPRS) or Internet every 10 minutes as CSV files. The data are processed, stored in a database and published in the Andalusian government web site. Then, as offered by the Andalusian regional government, we can download these data in PDF or HTML using a web form. From our perspective, these data are not presented in a friendly way, so a user with no knowledge can barely understand this information. Also, processing such information in near real-time would not be feasible because of the data structure and format.

3.2 Air Quality Levels

In order to measure the air quality based on the information provided by the air quality monitoring stations, we have to use a standard to be able to process and detect air quality alerts that affects the population.

Currently, there is a lack of an internationally recognized standard for measuring air quality levels. For this reason, many public and private institutions have developed their own air quality standards in order to be able to report air quality levels in their countries. Some of the most used are the Air Quality Index (AQI) [18], IND-AQI [19], CAQI [20], AQHI [21], and DAQI [22].

Our system is able to work with any of these standards, although we have decided to use the AQI provided by the US Environmental Protection Agency (EPA) [18] as our air quality level classification because it is based on pollutants that are very common on the air quality monitoring network that we use as data source.

According to EPA, each pollutant has several ranges of values, which determinates its quality level. Additionally, the EPA establishes that the highest (worst) level given by a single pollutant would be the air quality level in that area, e.g., if O3, CO, SO2, NO2 and PM2.5 show levels lower than 4, but PM10 level is 4; then it is considered that the air quality level in that area is 4 (Unhealthy).

Furthermore, the EPA provides a series of recommendations for the O3, CO, SO2, PM2.5 and PM10 pollutants in [18] that should be followed by citizens who are affected by these pollutants.

4. OUR AIR QUALITY PLATFORM

We propose a FIWARE-based platform capable of receiving information from different air stations and processing it, detecting dangerous air quality levels and promptly notifying them to interested parties, depending on their context.

4.1 Platform Requirements

An air quality platform should meet the following requirements to be technologically efficient: (1) to read air pollutant concentrations from diverse heterogeneous data sources, (2) to make users aware of the dangers and harms that poor air quality can cause to their health, (3) to process and analyze data in order to be aware of air quality levels in real time as well as to generate and notify the resulting alerts immediately, (4) to let users register to particular alerts to be notified in case of danger for their health, (5) to be aware of user context, based on his location, physical activity and personal characteristics, (6) to send personalized notifications based on user context, and (7) to let users check historical data about pollutant concentration levels for any required purpose: environmental issues, medical information, statistics, et cetera.

To comply with the established requirements, we designed and implemented the platform shown in Figure 1, which is organized in four logic layers: *Data Producers*, *Data Collector*, *Data Processing* and *Data Consumers*.

4.2 Data Producers Layer

This layer is composed of air sensor stations with the capacity of measuring air pollutants. In particular, 61 sensor stations located around the Andalusian territory are used, as described in Section 3.1.

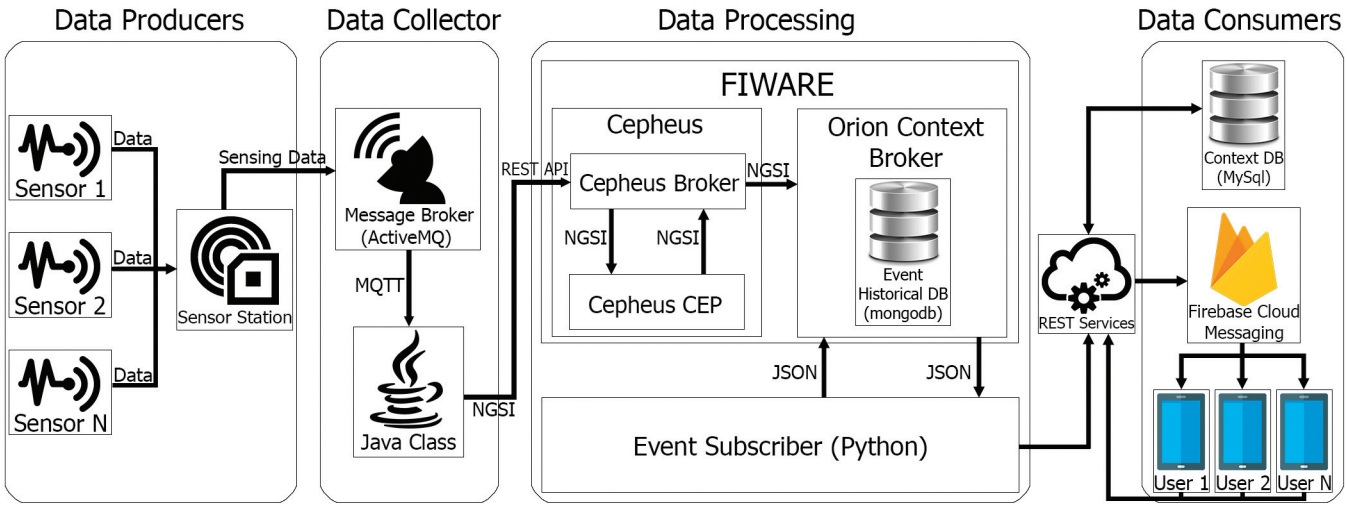

Figure 1 Platform for Remote Sensing and Smart Monitoring of Air Stations.

More specifically, this layer is in charge of making observations of the environment and sending them to the *Data Collector Layer* through ActiveMQ [23], one of the most popular and powerful open source message brokers, making use of MQTT and AMQP protocols. Note that air readings can have different formats depending on the air sensor manufacturers and features.

We would like to highlight that the Andalusian regional government shares their air sensing data with us, sending it to an own dedicated server every 10 minute by File Transfer Protocol (FTP). Then, we are able to process these CSV files and send them to our platform, which is responsible for real-time analyzing and correlating this information, detecting air quality levels unhealthy for citizens, as well as promptly notifying alerts to them through citizens' mobile phones, depending on their situational context.

4.3 Data Collector Layer

This layer works as a smart gateway between the *Data Producers Layer* and the *Data Processing Layer*. Therefore, the main functionality of the *Data Collector Layer*is to collect air observations taken around the territory to transform them into a common event format and to send the resulting events to the *Data Processing Layer*.

This smart gateway is composed of a Java class that (1) consumes air sensing data from the Message Broker using the MQTT protocol, (2) transforms the received messages containing the sensing data into the NGSI (Next Generation Service Interface) event format supported by FIWARE platform, and (3) sends these transformed events to FIWARE.

This layer facilitates the scalability and loose coupling of the *Data Producers* from the *Data Processing* since this is the unique layer of the whole platform that should be adapted to communicate other new heterogeneous air sensor stations with the *Data Processing Layer*. In that case, the implementation of a new data-event transformer would be required, without modifying any other parts in the platform.

4.4 Data Processing Layer

This layer has the capacity of receiving sensing data in form of simple events from the *Data Collector Layer*, and analyzing and correlating them to detect situations of interest in real time, and notifying such situations to the data consumers. This layer is composed of two FIWARE components: Cepheus and Orion Context Broker.

Cepheus is the component in charge of processing data to detect situations of interest in real time. It is composed of two modules: Cepheus Broker and Cepheus CEP engine.

Cepheus Broker is responsible for receiving the external sensing data in form of simple events and sending them both to be processed by the Cepheus CEP engine and to be registered in the Event Historical MongoDB database by the Orion Context Broker. In addition, it receives the complex events (situations of interest) created by the Cepheus CEP engine and submits them to the Orion Broker. Note that all communications between Cepheus and Orion are made by FIWARE NGSI API.

Cepheus CEP, which uses the open source and well-known Esper CEP engine, makes real-time detection of complex events possible when the conditions of one of the event patterns implemented and deployed in the engine are satisfied. A configuration file (*config.json*) defining the listening port, the incoming simple event schema, the generated complex event schema, and the implementation of patterns to be detected must be applied to the Cepheus CEP component.

In particular,the simple event schema defined is as follows: *AirMeasurement (stationTs long, stationId integer, pm2_5 double, pm10 double, o3 double, no2 double, so2 double, co double)*. This schema allows us to register the following information from every air sensor station: the timestamp in which the air measurements are taken, the station identification and location, as well as the reading values for every air pollutant.

The event patterns for detecting air quality levels for every location in real time (see Section 3.2) are implemented in EPL. These patterns have been deployed in the CEP engine through the REST API provided by Cepheus Broker.

In particular, seven patterns for each pollutant (one for the average and six for the different levels) together with one pattern for the air quality level are required:

Listing 1 EPL implementation to determinate the air quality level in a station.

```
@Name ('AirQualityLevel')
@PRIORITY (1)
INSERT INTO AirQualityLevel
SELECT a1.stationId.toString() || current_timestamp().toString() as id,
 current_timestamp() as timestamp,
 a1.stationId as stationId,
max(a1.levelNumber) as level
from pattern [every a1 = PollutantLevel].win:time_batch(600000
milliseconds)
GROUP BY a1.stationId
HAVING  max(a1.levelNumber) is not null
```

- Pollutant Average, to get the average value of the pollutant in the *X* hours sliding window, being *X* the number of hours specified by EPA for that pollutant.

- Pollutant Level, to determine the alert level of an air station for a particular pollutant. Having six levels, we will need a pattern for each level and for each pollutant.

- Air Quality Level, to get the highest level of alert detected in a station. This pattern will determinate the air quality level for that station.

Listing 3 illustrates *Air Quality Level* pattern; as it can be seen its implementation is quite similar to a standard SQL sentence. First, we set a name for the pattern using the *@NAME* keyword. Second, we set the priority of the pattern using the *@PRIORITY* keyword. Third, we define the complex event flow in which the pattern will store the information with the *INSERT INTO* keyword. Fourth, we select with the *SELECT* keyword the event attributes to be used to generate the complex event; new ones can be used. Fifth, by using the *FROM* keyword, we define the pattern conditions to be matched. Finally, we use other keywords, such as *GROUP BY* or *HAVING,*to perform certain operations (grouping by *id* and having an alert level not null, respectively) over the results already filtered. Note that all pattern implementations and the air quality levels detected for several months using such patterns are available at http://dx.doi.org/10.17632/84kn7s4c3n.1

In this Data Processing Layer, Orion Context Broker can be considered the backbone of our solution responsible for integrating the remaining system components. It allows us to gather, store and route data through the platform as well as to manage the whole context life cycle including updates, queries and logs by using a REST API. Note that Orion does not require a previous configuration as Cepheus does; upon executing, it is ready to receive, modify, query and delete information if its data schema and model are satisfied.

In particular, our platform is capable of managing two types of context in order to customize the alerts sent to users:

- Static: there is no need to continuous context monitoring. The user provides the particular location and physical activity as well as the personal characteristics (age, heart illnesses, etc.) to the system when registering in the app the first time he uses it, or occasionally updates it afterwards through the mobile app menu.

- Dynamic: the context requires continuous monitoring. The particular location and physical activity are obtained through the mobile GPS.

Further information about static and dynamic situational context can be found at [24].

4.5 Data Consumers Layer

This layer includes all the components necessary to manage the detected alerts and to interact with the citizens. In particular, it provides a REST service in charge of notifying customized alerts to users'Android mobile devices through Firebase Cloud Messaging [25]. These devices enable us to provide the platform with the static and dynamic contextual information described in Section 3.3, such as GPS user location, timetable user location and their personal data and preferences. This contextual information can be stored in the Context SQL database by using another REST service; such information is used to create the alerts customized according to the user context.

5. EXPERIMENTAL VALIDATION AND RESULTS

This section presents an experimental validation in order to demonstrate the effectiveness of our platform for remote sensing and smart monitoring of air stations.

Our platform is based on FIWARE. In particular, we have used a community account in Spanish FIWARE Lab cloud infrastructure enabling the development of a Virtual Machine hosting an instance of Orion Context Broker (version 1.6), Cepheus (version 0.1.8) and MongoDB database (version 3.0.12). The VM has the following characteristics: 4 VCPUs, 4 GB of RAM and 40 GB disk. The message broker is installed in an external machine to not consume resources from the processing one. The Java collector class is deployed in a machine with eight 2.6 GHz CPUs and 8 GB of RAM memory. All the communications are done by a broadband network. Finally, Samsung Galaxy A5 smart phones have been used as architecture clients.

We have analyzed and reported the obtained results during the air observations made around the Andalusian territory from December 1st, 2016 to April 30th, 2017. As explained in

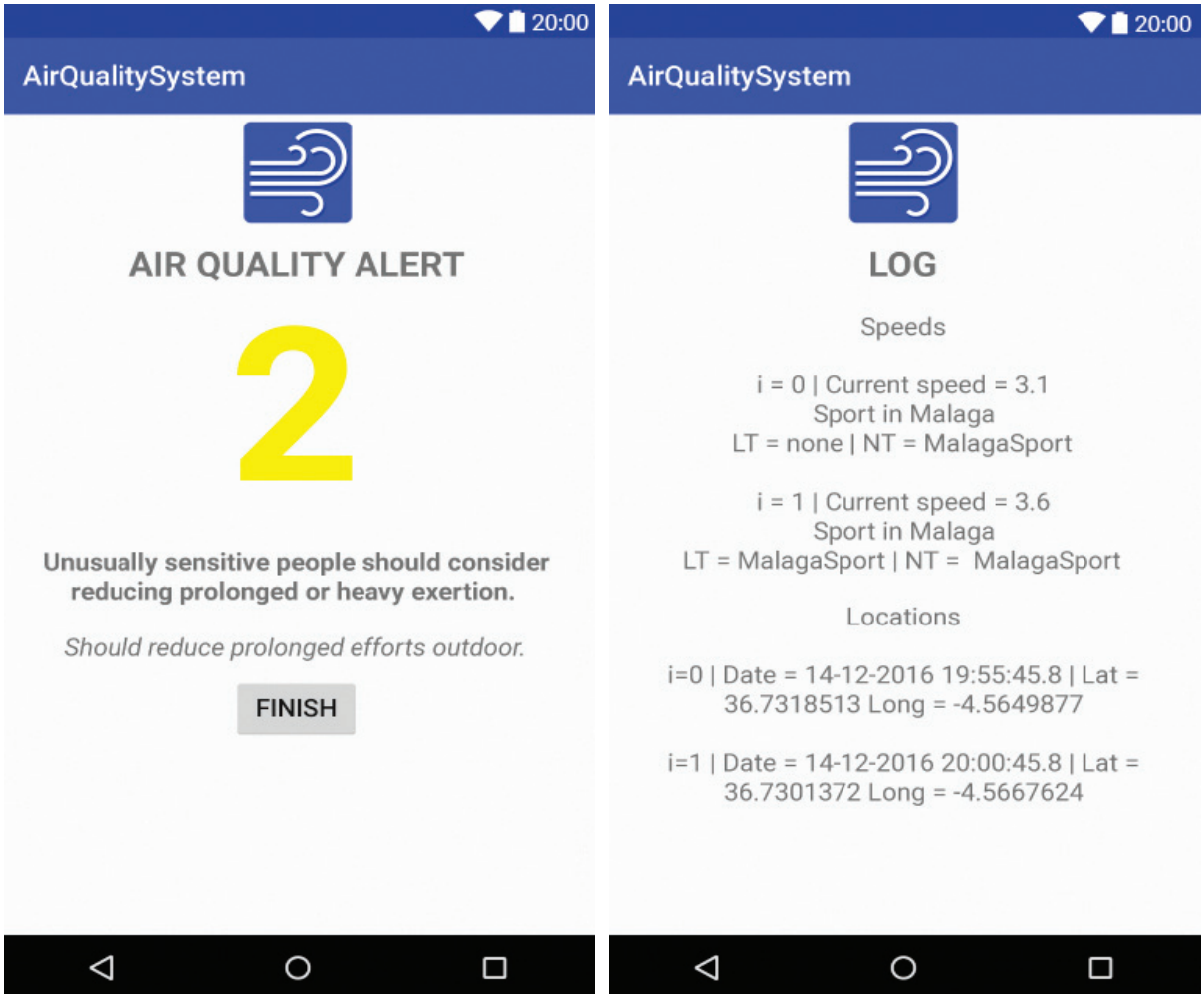

Figure 2 App screenshot of the 2-air quality alert received by a citizen. (b) App screenshot of the log activity where citizen's location and physical contexts are displayed.

Section 3.1, the air sensor network is composed of 61 air stations capable of measuring the key air pollutants: $PM_{2.5}$, PM_{10}, CO, O_3, NO_2 and SO_2.

Every time an air quality level changed in a location, our platform sent a customized alert to every citizen subscribed to such location alerts. For instance, Figure 2(a) illustrates a level-2 air quality alert received in the smart phone of a citizen that was close to the location where the alert was detected (the city of Málaga). Depending on the user context, the alert could have given him/her a different recommendation. As an example, the user, which was running in this location on December 14th, 2016 at 20:00 h. –this contextual information was obtained from the user device, see Figure 2(b)–, received a level-2-alert with the following advice: "Should reduce prolonged efforts outdoor" in the user's smart phone through the Firebase Cloud Messaging technology.

Moreover, we have evaluated the performance of the system using different rates of incoming events in order to determinate the maximum incoming events rate that our platform is able to process. To address it, we have used Apache JMeter [26] as the data producer for our platform since this open source Java application is designed for load testing and performance evaluation. More specifically, the data generated by JMeter were sent to the ActiveMQ broker previously defined. Then, these data were consumed and transformed into events by the Data Collector layer, and sent to the Data Processing layer.

As we can see in Table 1, we tested the system with different incoming events speeds and measured how many events were processed both by the Data Collector layer and by the Data Processing layer. In this test, JMeter is generating from 150 to 400 events per second (e/s) for 10 minutes, resulting in a maximum of 225 773 events produced. Note that the e/s rate is not always accomplish by JMeter due to the high processing load, that is why in Table 1 we see reflected, for example, 87 616 events in 10 minutes rather than 90 000 ones for an incoming rate of 150 events/s for 10 minutes). In particular, the system is able to process up to 300 incoming events per second; however, with 350 incoming events per second and up, the system suffers a short delay when processing the events in right now, requiring some more time to complete the processing. As we can see in FIWARE QA Activities [27], in the "Tested GEs and results" section, the performance in stress condition for the conducted tests is a hundred requests per second, while our architecture can manage three times that ratio without any problems.

Considering that air quality levels usually change with a frequency higher than 10 minutes and that most air sensor stations do not take the air measurements at exactly the same time, we can affirm that our proposed platform performance results are adequate for the air quality application domain.

Therefore, we can conclude that our platform satisfies all the requirements for effective remote air sensing and intelligent monitoring established in this paper:air quality information and

Incoming Event Rate (events/second)	JMeter (events produced in 10 minutes)	Data Collector (events processed in 10 minutes)	Data Processing (events processed in 10 minutes)
150	87 616	87 616	87 616
200	117 683	117 683	117 683
250	147 838	147 833	147 833
300	178 084	178 082	178 082
350	192 923	191 914	119 861
400	225 773	202 245	107 867

alerts are updated in real time, the information is actively provided to citizens in a clear and user-friendly way, the information provided to users is adapted to their specific characteristics, and the system takes into account the user developed activity and adapts notifications accordingly. Besides, historical air quality data are available through a REST API to any interested party.

6. RELATED WORK

Over the last years, several IoT applications have been proposed. In particular, these applications are available for different IoT cloud platforms such as the open-source platforms FIWARE, OpenMTC and SiteWhere, and by the proprietary one Amazon Web Services. A comparison of these platforms can be found in [28].

Concerning proposals based on FIWARE, the cloud platform used in our solution, Fazio et al. [12] make use of FIWARE to design a real e-health remote patient monitoring architecture, which allows caregivers to improve remote assistance to patients at home. Wolfert et al. [29] describe how FIWARE has been applied to the smart farming domain, giving support to tackle the data chain of big data applications: data capture, data storage, data transfer, data transformation, data analytics and data marketing. Martínez et al. [30] also propose a FIWARE-based IoT platform capable of managing very frequent exchanges of small amounts of data from low capacity sensors in the domain of precision agriculture. In addition, López-Riquelme et al. [31] implement a software architecture for precision agriculture, making use of specific FIWARE smart agricultural nodes as data producers. Fernández et al. [32] have developed SmartPort, a FIWARE-based platform for sensor data monitoring in a seaport located in Gran Canaria, Spain. On the other hand, Granell et al. [33] have investigated some technologies for environmental applications and have proposed the Environmental Observation Web, in which most data are considered observations that can be generated from hardware sensors, model simulations and human sensors. Moreover, two particular examples of environmental applications have been conducted in such work: biodiversity (ENVIROFI-BIO) and marine (MAST).

Comparing these proposals with our work, most of them do not benefit from using the CEP technology and Cepheus component to automatically detect meaningful patterns in real time; Fazio et al.'s one is an exception. In addition, none of them applies FIWARE to the particular air quality domain, as our platform does. The most relevance difference is that none of them deals and takes into account the user's particular situational context.

Besides, there are other works specifically developed for air quality monitoring such as SmartSantander [5], CITI-SENSE [6] and OpenSense II [7]. Nevertheless, they present some limitations compared to our work. As an example, CITI-SENSE lacks a REST API available for users, so data have to be downloaded in CSV or XLS formats. Moreover, none of them provides recommended air quality notifications based on citizens' personal, location as well as physical contexts.

Finally, although it is not a particular system for data processing, Sun et al. [34] propose an open IoT framework based on micro services that shares some of our platform's components: ActiveMQ for data communication and Esper for real-time event processing and pattern detection.

7. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel IoT platform based on FIWARE in order to prevent further health risks derived from unsuitable air quality, through the submission of personalized alerts based on real-time air quality levels and on location, personal and physical contexts of registered users. The system has been designed to be flexible and configurable, so that it can be integrated with third party air stations and sensors without the need for system structure modification. Besides, it can be particularized for other case studies and scenarios where IoT data had to be processed in real-time in order to detect situations of interest for the scope in question. Moreover, this system can be invoked from different clients thanks to the implementation based on recognized standards.

As future work, we expect to extend our solution with several features: extending the number of contexts taken into account by the system; including observations generated from air-quality models to predict pollution exposures; and integrating the system with medical services so that doctors, for instance, can be aware of the air quality which their patients are breathing.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Ministry of Science and Innovation and the European Union FEDER Funds under Projects TIN2015-65845-C3-3-R and RTI2018-093608-B-C33.

REFERENCES

1. WHO, "Review of evidence on health aspects of air pollution – REVIHAAP Project," 2013.

2. WHO, "Ambient (outdoor) air quality and health," *WHO*. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs313/en/. [Accessed: 11-Oct-2018].
3. "Air quality in Europe 2017," European Environment Agency (EEA), Luxembourg: Publications Office of the European Union, Publication 13/2017, 2017.
4. Silveira, C, Roebeling, P, Lopes, M, Ferreira, J, et*al.*, "Assessment of health benefits related to air quality improvement strategies in urban areas: An Impact Pathway Approach," *Journal of Environmental Management*, vol. 183, no. Part 3, pp. 694–702, Dec. 2016.
5. "SmartSantander," 2014. [Online]. Available: http://www.smartsantander.eu/. [Accessed: 11-Oct-2018].
6. "CITI-SENSE," 2016. [Online]. Available: http://www.citi-sense.eu/. [Accessed: 11-Oct-2018].
7. "OpenSense II," 2016. [Online]. Available: http://opensense.epfl.ch/wiki/index.php/OpenSense_2.html. [Accessed: 11-Oct-2018].
8. S. He, B. Cheng, H. Wang, Y. Huang, and J. Chen, "Proactive personalized services through fog-cloud computing in large-scale IoT-based healthcare application," *China Communications*, vol. 14, no. 11, pp. 1–16, Nov. 2017.
9. "FIWARE." [Online]. Available: http://www.fiware.org/. [Accessed: 11-Oct-2018].
10. Atzori, L, Iera, A and Morabito, G "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
11. Khan, M, Din, S, Jabbar, S, Gohar, M, *et al.,* "Context-aware low power intelligent SmartHome based on the Internet of things," *Computers & Electrical Engineering*, vol. 52, pp. 208–222, May 2016.
12. Fazio, M, Celesti, A, Márquez, FG, Glikson, A,*et al.*, "Exploiting the FIWARE cloud platform to develop a remote patient monitoring system," in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, pp. 264–270.
13. Douzis, K, Sotiriadis, S, Petrakis, EGM and Amza, C "Modular and generic IoT management on the cloud," *Future Generation Computer Systems*.
14. Boubeta-Puig, J, Ortiz, G and Medina-Bulo, I, "MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0," *Knowledge-Based Systems*, vol. 89, pp. 97–112, Nov. 2015.
15. EsperTech Inc., 2018. [Online]. Available: http://www.espertech.com. [Accessed: 11-Oct-2018].
16. Schmidt, A "Implicit human computer interaction through context," *Personal Technologies*, vol. 4, no. 2–3, pp. 191–199, Jun. 2000.
17. Furno, D, Vincenzo, L, Veniero, M, Anisetti, M,*et al.*, "Towards an agent-based architecture for managing uncertainty in situation awareness," in *2011 IEEE Symposium on Intelligent Agent (IA)*, 2011, pp. 1–6.
18. EPA, "Technical Assistance Document for the Reporting of Daily Air Quality - the Air Quality Index (AQI)," U.S. Environmental Protection Agency, North Carolina, US, EPA-454/B-16-002, May 2016.
19. CPCB - India, "National air quality index," Government of India, New Delhi, India, 82, Oct. 2014.
20. CITEAIR, "Air Quality Now in Europe," 2016. [Online]. Available: http://airqualitynow.eu/. [Accessed: 11-Oct-2018].
21. Government of Canada, "Air Quality Health Index," 2016. [Online]. Available: http://www.ec.gc.ca/cas-aqhi/default.asp?Lang=En&n=065BE995-1. [Accessed: 11-Oct-2018].
22. Department for Environment Food & Rural Affairs - UK, "Daily Air Quality Index," 2016. [Online]. Available: http://uk-air.defra.gov.uk/air-pollution/daqi?view=more-info&pollutant=pm10#pollutant. [Accessed: 11-Oct-2018].
23. Apache, "ActiveMQ," 2017. [Online]. Available: http://activemq.apache.org/. [Accessed: 11-Oct-2018].
24. Garcia-de-Prado, A, Ortiz, G, Boubeta-Puig, J and Corral-Plaza, D "Air4People: a Smart Air Quality Monitoring and Context-AwareNotification System." [Online]. Available: http://www.jucs.org/jucs_24_7/air4people_a_smart_air. [Accessed: 04-Oct-2018].
25. "Firebase Cloud Messaging | Send notifications across platforms for free," *Firebase*. [Online]. Available: https://firebase.google.com/products/cloud-messaging/. [Accessed: 11-Oct-2018].
26. Apache Software Foundation, "Apache JMeter," 21-Dec-2017. [Online]. Available: http://jmeter.apache.org/. [Accessed: 11-Oct-2018].
27. "FIWARE QA Activities - FIWARE Forge Wiki." [Online]. Available: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/ FIWARE_QA_Chapter#GEs_and_obtained_results_of_release_5.2_stress_testing. [Accessed: 11-Oct-2018].
28. Guth, J, Breitenbücher, U, Falkenthal, M, Leymann, F, *et al.*, "Comparison of IoT platform architectures: A field study based on a reference architecture," in *2016 Cloudification of the Internet of Things (CIoT)*, 2016, pp. 1–6.
29. Wolfert, S, Ge, L, Verdouw, C and Bogaardt, MJ "Big Data in Smart Farming – A review," *Agricultural Systems*, vol. 153, pp. 69–80, May 2017.
30. Martínez, R, Pastor, JA, Álvarez, B and Iborra, A, "A Testbed to Evaluate the FIWARE-Based IoT Platform in the Domain of Precision Agriculture," *Sensors*, vol. 16, no. 11, p. 1979, Nov. 2016.
31. López-Riquelme, JA, Pavón-Pulido, N, Navarro-Hellín, H, Soto-Valles, F, *et al.,* "A software architecture based on FIWARE cloud for Precision Agriculture," *Agricultural Water Management*, vol. 183, pp. 123–135, Mar. 2017.
32. Fernández, P, Santana, JM, Ortega, S, Trujillo, A, *et al.*, "SmartPort: A Platform for Sensor Data Monitoring in a Seaport Based on FIWARE," *Sensors*, vol. 16, no. 3, p. 417, Mar. 2016.
33. Granell C, Hvlik, D, Schade, S, Sabeur, Z, *et al.*, "Future Internet technologies for environmental applications," *Environmental Modelling & Software*, vol. 78, pp. 1–15, Apr. 2016.
34. Sun, L, Li, Y and Memon, RA "An open IoT framework based on microservices architecture," *China Communications*, vol. 14, no. 2, pp. 154–162, Feb. 2017.