An Online Platform For Real-Time Air Quality Monitoring

Jorge Silva[1], Pedro Lucas[1], Filipe Araújo[1], Catarina Silva[1,2], Paulo Gil[1], Alberto Cardoso[1], Joel Arrais[1], Bernardete Ribeiro[1], Daniel Coutinho[3], Pedro Salgueiro[3], Luís Rato[3], José Saias[3], Vítor Nogueira[3]
[1]CISUC - Center for Informatics and Systems, University of Coimbra, Portugal
[2] School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[3] LISP - Laboratory of Informatics, Systems and Parallelism, University of Évora
{jadsilva@student.dei.uc.pt, phlucas@student.dei.uc.pt, filipius@uc.pt, catarina@dei.uc.pt, pgil@dei.uc.pt, alberto@dei.uc.pt, jpa@dei.uc.pt, bribeiro@dei.uc.pt, m43354@alunos.uevora.pt, pds@uevora.pt, lmr@uevora.pt, jsaias@uevora.pt, vbn@uevora.pt}

***Abstract*—The interest in the quality of air is likely to increase, as the public concern for health and environmental issues is on the rise. So far, most data available comes from a small numbers of government-owned sensors, lacking a wide coverage of the entire reality. Improving the amount of data available thus involves reducing the cost of sensors and make their readings accessible to the public. The NanoSen-AQM project aims to do precisely that. Create and use vast numbers of low-cost nano-sensors, to make their data accessible for the public. To achieve such an ambitious goal, the project will use state-of-the-art techniques from Machine Learning and mobile and web development frameworks. As a result, the NanoSen-AQM platform should provide free access to the public and low-cost of entry for sensor owners willing to share their data.**

***Index Terms*-Air Quality Monitoring, Low-cost Nano-sensors, Cloud Platform, Mobile Application**

I. Introduction

Air pollution presents challenges both for the environmental and for social scenarios, since it often raises adverse effects on human health, ecosystems and the climate [1]. In fact, people are becoming aware that air pollution is one of the foremost causes of environmental and social health risks [2].

Hence, several approaches are being carried out to monitor air quality as a preventive approach to allow for citizens and municipal and country authorities to take timely appropriate decisions regarding population safety. A recent survey on air pollution monitoring systems [2] can help shedding light on the current state of the art.

However, air quality monitoring is expensive [3], which makes it difficult to obtain massive, reliable, real-time data. A popular approach is to use mobile personal devices to gather information on air quality, forming a crowd, as in [4]. An alternative approach is the development and use of low-cost gas nanosensors to measure air-quality and provide real-time data, as the NanoSen-AQM project, providing massive, distributed and ubiquitous air quality monitoring.

The rest of the paper is organized as follows. Next section will briefly present some basic concepts, and the following section will introduce related work. Section IV will present the NanoSen-AQM project and the following sections will introduce the development carried out and results achieved. The paper ends with conclusions and future work.

II. Basic Concepts

When we talk of air quality monitoring, a number of concepts emerge, like atmospheric pollution, air quality monitoring and Air Quality Index [5]. **Atmospheric pollution** is any form of matter or energy with intensity, concentration, time or characteristics that may render the air inappropriate, harmful or offensive to health. As a consequence, **air quality monitoring** is carried out to determine the concentration of pollutants in the atmosphere.

Air Quality Index is a standardized index that represents the degree of pollution in the air. It is the product of the interaction of a complex set of factors, including the magnitude of the emissions, the topography and the meteorological conditions of the region, which may be favorable or not to the dispersion of pollutants. The air quality of an area or region is determined by analyzing atmospheric pollutants, which are compared to the pollutant concentration standards set out in environmental legislation (e.g the european CAQI [6]). Air quality standards refer to concentrations of air pollutants which, when exceeded, may affect the health and well-being of the population, as well as damage flora and fauna, materials and the environment in general. These standards are based on scientific studies of the effects produced by specific pollutants and are set at levels that can provide an adequate margin of safety.

The Air Quality Index is usually an integer number representing an hazard level, often translated to color codes. For example, the Environmental Protection Agency in the United States uses the following scale [7]: Green/Good,

This work has received funding from the Program Interreg-Sudoe of the European Union under grant agreement SOE2/P1/E0569 (NanoSen-AQM) and was partially financed by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2019.

978-1-7281-3636-3/19/$31.00

Yellow/Moderate, Orange/Unhealthy for Sensitive Groups, Red/Unhealthy, Purple/Very Unhealthy, Brown/Hazardous; whereas other countries or regions, like Europe, use their own scales, e.g., with only five levels [8]: Light Green/Very Good, Dark Green/Good, Yellow/Average, Orange/Poor, Red/Bad.

III. Related Work

A. Currently Available Applications

To do a comprehensive evaluation of the market, we searched for applications/platforms that work on the analysis and monitoring of air quality in the various mobile and web platforms. This research was based on filtering the most relevant results regarding air quality concepts, while searching in the mobile app stores (AppStore and PlayStore) and the Google search engine. Based on this effort, we gathered the following list of applications:

- **Android platform:** Airveda[9], AirVisual [10] and Plume[11];
- **iOS platform:** Air Matters[12] and AirVisual[13];
- **Web platform:** Real-Time World Air Quality Index[14] and AirVisual[15];

From this list, we decided to evaluate a smaller subset of applications, based on the number of their downloads (more than 10,000) and on their evaluation (more than 4.2 stars). In the case of websites, we only considered those that covered more than one country and appeared first in the surveys. These reasons led us to select the AirVisual, AirVeda and Plume applications for a deeper evaluation, as they were the best classified on the previous parameters.

*1) **AirVisual**:* The AirVisual application provides real-time Air Quality Index, where it is used by government agencies to communicate air pollution to the public in the most diverse locations in the world. In addition to this information, it still includes the values of $PM_{2.5}$ and PM_{10}, allows the verification of historical data on air quality in any location, comparison of data with other cities and visualization of forecasts of air quality in real time. This application still allows the user to view the data registered by each sensor (Air Quality Index, inhalable particles — $PM_{2.5}$ and PM_{10} —, humidity, wind and a forecast of air quality in the coming days), have a list of favorite sensors, to monitor closely without having to go to a general listing, have a history of records and locate the sensors geographically. In addition to all these features related to the theme, it also has user profile management features and application settings.

*2) **AirVeda**:* The AirVeda application provides real-time Air Quality Index from a variety of locations around the world. Like the AirVisual application, this application presents the same data on air quality and data comparison, having, as a differentiating factor, the functionality of connecting AirVeda personal monitors to the platform. The main pages of this application present the sensors stored as favorites in the main listing, as well as the most important data about them in their section (Air Quality Index, $PM_{2.5}$ and PM_{10}). In addition to the monitoring of air quality, the application also advises activities that can be done by the user depending on the air quality at the moment, such as, playing sports if the air quality is good. The data visualization is aided by filters, so that the user can visualize the desired data in the best possible way. It is also possible to personalize the application settings according to the desired preferences. In addition, it also includes a Frequently Asked Questions section with the answers to the most common user questions.

*3) **Plume**:* The Plume application provides all the air quality indexes of the previous applications, but shows them in a more simplistic, less intuitive way. When the application starts, the user is prompted to define a sensor as the primary sensor, by searching for the name of his/her location. Once the sensor is set, the user is presented with the main page with the information about the air quality registered by a certain sensor. It allows the user to check the air quality forecasts during a certain time interval, as well as the record history of the air. Through a sidebar menu, it is possible to check the air quality worldwide, by means of a map filled, selecting the option "World Air Map". In the same menu, the user can also access options such as settings, Frequently Asked Questions, application information, application blogs and other extra features that are related theme (e.g., buy portable sensors, etc). In this application it is also possible to register in the system, so that the user can access his information stored in any device. This allows the user to not always have to walk with his device to access the information of the sensors that are stored in his profile.

*4) **Comparison**:* After analyzing the functionalities of the applications mentioned above, it is possible to present a comparison of the common and non-common functionalities of the applications. For this comparison experiments were made on the applications, not only to understand the functionalities of each one, but also to understand the points that might be missing.

TABLE I
Comparison of applications functionalities.

Applications	AirVisual	AirVeda	Plume
Android	X	X	X
iOS	X		
Web-site	X		
Geo-localized data	X		X
Favorites	X	X	X
Alerts	X	X	
Sensor management		X	
Data extraction			
Data Insertion			
Manage Calibrations			
Forecast	X	X	X

As shown in table I, the previously analyzed applications do not present some necessary functionalities, like data extraction/insertion and calibration management. These functionalities are proposed since they are the ones that cover more needs, since it not only focuses on the analysis and comparison of data, but also allows the extraction of data for external analysis. In addition it also allows the existence of several

users, such as the owners of sensors (user that can register its own sensors in the system, so that they are later integrated into it), which will have all the management functionalities to their sensors, as is the case of data insertion/extraction and management of calibrations. Another aspect missing by some applications is the scope of the iOS and web platform. Even so, with these differentiating points, all the mentioned platforms are considered successful, as they achieve their goal, which is the presentation of air quality data, and many users use them.

B. Similar Air Quality Platforms

Some air quality platforms were also analyzed, where the focus was the internal mechanisms, such as, the architecture, the data sources and the technologies used by each one. In addition to describing these systems, a comparison was also performed, evidencing their divergence from the NanoSen-AQM platform.

1) OpenAQ: OpenAQ is an open-source platform that aggregates real-time and historical air quality data, originated from worldwide sources [16]. Access to stored data is completely free, allowing it to be used by fields, such as, education, research, other open-source platforms, etc. The purpose of this platform is to fight air inequality, by collecting air quality data from governmental sources and publishing them in a universal format.

It is completely deployed in the (Amazon Web Services) AWS cloud infrastructure and also uses some of its services (e.g data older than 3 months is stored in a Amazon S3 bucket). Air quality data is fetched from external governmental sources, in a process similar to a web crawler.

One of the aspects that distinguishes the OpenAQ platform from the NanoSen-AQM project is the processing of air quality data. OpenAQ simply fetches data from governmental sources, which exclude air pollution measurements collected from low-cost sensors, the primary data source of the NanoSen-AQM platform. As such, the latter has the additional responsibility of calibrating the air quality data, in order to correct problems related with low-cost sensors (e.g low sensitivity, specificity and stability).

2) Breezometer: Breezometer is a platform that provides real-time, dynamic, accurate and location-based air quality data [17]. The service is directed at the enterprise sector, in particular, companies that have interest in air purification, smart homes, medical field, smart cities and fitness devices. Real-time air quality data is obtained by using a proprietary dispersion algorithm. It takes into account not only raw data collected from monitoring stations, but also local traffic data, satellite weather data, wind patterns and more. Air quality data is then calculated and displayed on a two-dimensional grid, allowing for accurate data at the street level (500 meters) [18, 19].

The data sources, like with the OpenAQ platform, are governmental monitoring stations, that employ reliable, but expensive sensors. The NanoSen-AQM platform will try to reach a similar coverage as Breezometer by using low-cost sensors, making it a more financially viable solution. However, the project has also to take into account the lack of accuracy and reliability of measurements collected by low-cost sensors, requiring a calibration procedure in order to correct this problems.

IV. The NanoSen-AQM Project

A. Overview

The NanoSen-AQM Project is in the scope of the Interreg Sudoe Program that supports regional development in south-western Europe. Thus, it promotes transnational cooperation to address issues common to the regions of that territory, such as low investment in research and development, low competitiveness of small and medium enterprises and exposure to climate change and environmental risks. The NanoSen-AQM Project, as illustrated in Fig. 1, is in line with these main objectives. It aims at delivering three main products, namely:

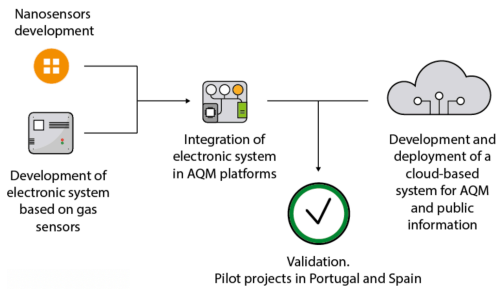

Fig. 1. Overall scheme of the NanoSen-AQM

1) an Advanced nanostructured gas sensors (NO_2, O_3 and CO), 2) an Electronic system based on nanosensors for real-time detection of ambient air pollutants and 3) a Distributed cloud system as a basis to offer Software as a Service (SaS) for Air Quality Monitoring (AQM). To reach these goals the project is organized into six main task groups that cooperate in the transnational ground field for fulfill the above goals. The task groups cover the following parts of the overall system: (i) specification of the scenarios and field campaigns for air quality monitoring, (ii) development of low cost sensors, (iii) design of the electronic system, (iv) integration of the nanosensors in the electronic system, (v) cloud system online platform and (vi) overall validation in the field terrain. The project has 15 main beneficiaries among academic institutions, research institutes, and companies spread over Spain, France and Portugal.

B. Low-Cost Sensors

In this project, we mostly focus on a small subset of pollutants, which we can arguably consider as the most important ones, due to their impact on human health. In particular, we will develop and deploy sensors for the following list of pollutants:

- Sulfur Dioxide (SO_2): a chemical resulting from the combustion of sulfur-containing fuels, such as diesel oil, fuel oil and gasoline.
- Nitrogen Dioxide (NO_2): a chemical formed during the combustion processes. In large cities, vehicles are generally responsible for the emission of this chemical.
- Carbon monoxide (CO): colorless, odorless gas resulting from the incomplete burning of organic fuels (fossils, biomass, etc.). In general it is found in areas of significant circulation of vehicles.
- Ozone (O_3): pollutant formed by the reaction between oxides of nitrogen and volatile compounds in the solar presence.
- Inhalable particles (PM_{10}/$PM_{2.5}$): airborne particles whose aerodynamic diameter is less than, resp., 10 or 2.5 μm, which may be aspirated by the mouth or nose during the breathe process.

C. The Online Platform

The online platform that will support the NanoSen-AQM air quality monitoring system is comprised of two main subsystems, the back-end and the front-end. The back-end will encompass the entire system from the sensor endpoints to a central database that will store all collected data. It will also include an application programming interface (API), to let the front-end and third-party applications to access the data. In simple terms, the back-end moves data from the sensors in the field to the central repository.

The front-end subsystem provides the visible face of the project and lets end users interact with data, mostly for consuming. The front-end should provide user interfaces for the vast majority of mobile devices, and for desktop and laptop users. The front-end system should support different sorts of users, including administrators and owners of sensors throughout the world.

We will develop the platform with a number of goals in mind: the source code should be open, to let third parties build their own platforms and extend the solution; the solution should be scalable, to support a large number of sensors and users; the platform should be able to run on the cloud, but, at the same time, we should avoid lock-in and we should be able to run it on premises, under the control of the project stakeholders.

V. The Back-End System

The monitoring system must take into account aspects regarding data collection, sensor management, and access to stored data on air quality. The following subsections describe how the back-end of the platform will be developed.

A. Architectural Drivers

The system should collect air pollution data from the air quality sensors and store it for later access. As mentioned previously, this data will need to be processed, using a calibration function, in order to increase its accuracy. Furthermore, the system will also fetch data from external air quality platforms, such as OpenAQ or governmental air quality agencies, thus by increasing the coverage of the NanoSen-AQM platform.

Regarding user interactions, the system should allow them to access stored air quality information and related metadata through client applications, described in more detail in Section VI. If requested, this data should be able to be downloaded as a csv file. Other user operations range from setting up alerts (e.g when air pollution goes above a defined threshold) to account management (e.g registration, authentication, profile edition, etc). Finally, privileged users (the sensor owners) should be able to manage their clusters (groups of sensors), their sensors and associated calibration functions.

In addition to performing the above functionalities, the system should be fault-tolerant and be able to scale with high workloads. As it will be deployed in the cloud and on-premises, the system should also be portable.

B. Technologies

In order to comply with requirements mentioned in the previous section, the following technologies were chosen for the system.

Data produced by the sensors will be uploaded using Message Queuing Telemetry Transport (MQTT), a publish-subscribe messaging protocol. This protocol was chosen due to its simplicity, which makes it perfect to be implemented in low computing power systems, such as the ones used in the sensors [20]. To use it, there is the need to have a MQTT broker, responsible for subscribing and publishing all messages. We have chosen the VerneMQ [21] MQTT broker, a scalable, widely used and open-source MQTT broker, which provides fault-tolerance mechanisms.

Data ingestion is expected to be high, as each sensor can produce multiple data points per second. This requires a data streaming platform to handle data transportation and storage while in-transit. The chosen technology was Apache Kafka [22], as it is scalable, contains fault-tolerance mechanisms and has already proven to be able to handle high workloads [23].

Regarding the database, a time-series database was preferred, because of the same reasons specified above. TimescaleDB [24], an extension to PostgreSQL, was chosen, as it adds time-series capabilities to an already reliable database. This simplifies the system setup, facilitates integration with other technologies and guarantees stability to the data storage.

Client applications communicate with back-end system through a RESTful API, developed in Django [25], a Python framework for creating web applications. Besides from being a mature framework, it can automatically generate an administration platform and allows APIs to be browsable.

Since the system should operate both in the cloud and on-premises, it must work in different operating environments. This can lead to situations such as some modules crash or won't start because some dependency is not available in an specific operative environment. In order to solve this problem, the NanoSen-AQM will use containers [26], as it packages the code and required dependencies into a dingle deployable unit.

Finally, regarding data calibrations, the system will use a serverless function platform, OpenWhisk [27]. This approach was chosen since it provides a way to easily manage and run custom functions.

C. Architecture

From the previously mentioned architectural drivers, the architecture of the system was designed. It is illustrated by Fig. 2, a container diagram that follows the level 2 of the C4 model [28]. By observing the diagram, some key aspects can be distinguished:

- Air pollution data is collected from the sensors by a "Data Aggregator", using the MQTT protocol. It is then sent to the central server by a Kafka topic, that is consumed by the "Data Loader", which stores the data in the database. Another data source is the external air quality platform, from which the NanoSen-AQM system fetches data, storing it in the database;
- Users interact with the NanoSen-AQM platform by using client applications. These, in turn, communicate with the central server by an "API Application", a RESTful API. This module also interacts with an external email service, using it to send emails (e.g registration, password reset, alerts, etc);
- For the calibration procedure, the "Data Calibration Loader" gathers raw data from the database and publishes it to a Kafka topic, which is then consumed by an OpenWhisk function, that calibrates the air quality data and stores it in the database. In addition, the calibrated data is published to another Kafka topic, which is consumed the "Alert Generator", that generates alerts.

VI. The Front-End Application

Considering the goals of the project, the need to run the application on multiple platforms (web and mobile) and the main approaches that target multiple platforms (Progressive Web Apps and Hybrid Applications [29]), we picked the Ionic technology to develop a hybrid application. In this way, with a single branch of development, we can have a web site and an application that can be available in mobile application stores.

Due to the complexity of the system, and according to existing roles, we decided to divide the users that interact with it, into three categories: **User**, with all the functionalities of data visualization of the system relative to the air quality; **Owner of sensors**, with all the features of common user with the privilege to add and manage his own sensors — this profile will only be allowed after registration and correct authentication in the system; **Administrator** with all privileges on the system, only available on the NanoSen-AQM system back-end.

This separation was carried out in order to protect the system and to distinguish what each user can do. So one of the platforms will contain data visualization and sensor monitoring features, only to sensors entered by its users, in this case the sensors owners. These latest features will only be available to users who authenticate with their "Sensor Owner" profile. Finally the other platform will only be for the context of management and control of the data by users registered in the system like Administrators.

VII. System Requirements

We can now identify the functional requirements very clearly:

1) **User**: this actor will have all the functionalities related to the visualization and monitoring of air quality. This will include the system's clusters/sensors listing functionality, current air quality data visualization, air quality statistical data visualization, addition/removal of clusters to a personal favorites list, extracting data through a comma-separated values (CSV) file, listing and editing alerts on air quality, and all the management of the personal account (eg. registration/authentication/editing of actor profile data). This role will only be fully available when the actor authenticates properly in the system.
2) **Sensors owner**: in addition to the capabilities of a simple User role, this actor can edit/create/remove data related to sensors and clusters, possibly using a CSV file. This role requires authentication with an account associated to sensor ownership.
3) **Administrator**: this actor can create/edit/delete the user records, the sensor logs and the records of clusters in the system. In addition, the actor will also have system authentication and password recovery capabilities.

With these requirements, the platform can meet the needs of the most wanted types of users, from those who only want to consult the air quality data, to those who want to insert their own air quality analysis sensors and make their own management.

VIII. Conclusion

In this paper, we presented the NanoSen-AQM project and went through the major technical challenges to create a platform capable of generalizing access to air quality data. We covered the back-end technologies from the sensors to the database and the front-end application to be built with Ionic.

References

[1] Marin B. Marinov, Ivan Topalov, Elitsa Gieva, and Georgi Nikolov. Air quality monitoring in urban environments. In *39th International Spring Seminar on Electronics Technology (ISSE)*, pages 443–448. IEEE, 2016.

[2] Raghavendra Khot and Vidya Chitre. Survey on air pollution monitoring systems. In *2017 conference on innovations in information, embedded and communication systems*. IEEE, 2017.

[3] Diane Mooney. A guide for local authorities purchasing air quality monitoring equipment. Technical report, Department for Environment Food & Rural Affairs, 2006. UK.

[4] Iasmina Gruicin and Mircea Popa. Airify – a mobile solution for monitoring air quality in urban areas. In *25th Telecommunications forum TELFOR 2017*. IEEE, 2017.

[5] José F. G. Mendes Lígia T. Silva1. Determinação do Índice de qualidade do ar numa cidade de média dimenção. https://repositorium.sdum.uminho.pt/bitstream/1822/6333/3/n_27_pag_63-74.pdf, 2006. Accessed 6 January 2019.

[6] Hermann Heich Karine Léger. 2012. Sef van den Elshout, Hans Bartelds. Caqi air quality index. comparing urban air quality across borders. https://www.airqualitynow.eu/download/CITEAIR-Comparing_Urban_Air_Quality_across_Borders.pdf. Accessed 4 April 2019.

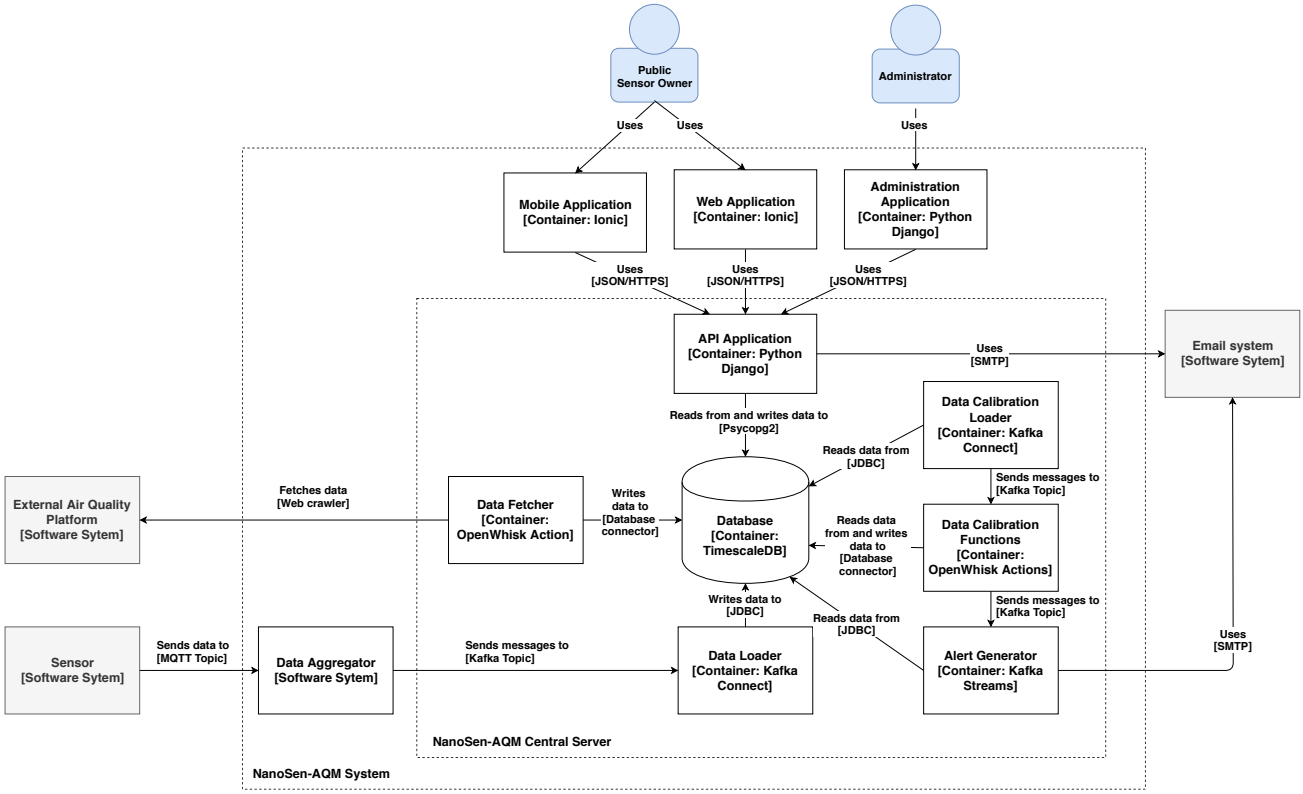

Fig. 2. Container diagram of the NanoSen-AQM platform

[7] the free encyclopedia Wikipedia. Air quality index - wikipedia. https://en.wikipedia.org/wiki/Air_quality_index#Europe, 2019. Accessed 14 January 2019.

[8] Agência Portuguesa do Ambiente. Agência portuguesa do ambiente - qualidade do ar. https://qualar.apambiente.pt/qualar/index.php?page=1&subpage=7. Accessed 14 January 2019.

[9] Airveda Technologies. Airveda - monitor air quality(2016). downloaded from: https://play.google.com/store/apps/details?id=in.airveda&hl=pt. Version 4.9.16 for platform.

[10] AirVisual. Air quality — airvisual(2015). downloaded from: https://play.google.com/store/apps/details?id=com.airvisual. Version 4.3.1-20 for Android platform.

[11] Plume Labs. Plume air report - live and forecast smog reports(2015). downloaded from: https://play.google.com/store/apps/details?id=com.plumelabs.air. Version 2.4.7 for Android platform.

[12] Air Matters. Air matters(2013). downloaded from: https://itunes.apple.com/us/app/air-matters/id477700080#?platform=iphone. Version 6.7.5 for iOS platform.

[13] AirVisual. Airvisual air quality forecast. downloaded from: https://itunes.apple.com/us/app/airvisual-air-quality-forecast/id1048912974?mt=8, 2015. Version 4.3.1 for iOS platform.

[14] World Air Quality Index team. World's air pollution: Real-time air quality index. https://waqi.info, 2007. Accessed 11 December 2018.

[15] IQAir. Airvisual — air quality information you can trust. https://www.airvisual.com, 2015. Accessed 11 December 2018.

[16] OpenAQ. 2018. Openaq about. https://openaq.org/#/about. Accessed 10 November 2018.

[17] BreezoMeter. 2019. Breezometer — worldwide aqi data api. https://breezometer.com. Accessed 3 January 2019.

[18] Idit Lowenstein. 2019. The 3 steps to the world most accurate air quality

[19] Scientia. 2019. Breezometer technology explained. https://info.breezometer.com/air-quality-technology-paper. Accessed 3 January data. https://blog.breezometer.com/accurate-air-quality-data. Accessed 3 January 2019.
2019.

[20] Andrew Banks and Rahul Gupta. Mqtt version 3.1. 1. *OASIS standard*, 29, 2014.

[21] VerneMQ. 2019. Vernemq - a mqtt broker that is scalable, enterprise ready, and open source. https://vernemq.com. Accessed 5 February 2019.

[22] Apache Kafka. 2019. Apache kafka - a distributed streaming platform. https://kafka.apache.org/. Accessed 18 January 2019.

[23] LinkedIn. 2019. Benchmarking apache kafka: 2 million writes per second (on three cheap machines) — linkedin engineering. https://engineering.linkedin.com/kafka/benchmarking-apache-kafka-2-million-writes-second-three-cheap-machines. Accessed 18 January 2019.

[24] Timescale. 2019. Timescale — an open-source time-series sql database optimized for fast ingest, complex queries and scale. https://www.timescale.com. Accessed 6 January 2019.

[25] Django. 2019. The web framework for perfectionists with deadlines — django. https://www.djangoproject.com. Accessed 6 January 2019.

[26] Docker. 2019. What is a container? — docker. https://www.docker.com. Accessed 14 January 2019.

[27] Apache OpenWhisk. 2019. Apache openwhisk is a serverless, open source cloud platform. https://openwhisk.apache.org. Accessed 7 January 2019.

[28] Simon Brown. 2019. The c4 model for software architecture. https://c4model.com/. Accessed 18 January 2019.

[29] Tripti Rai. Native vs hybrid vs pwas: What really matters to your business. https://android.jlelse.eu/native-vs-hybrid-vs-pwas-what-really-matters-to-your-business-c0058ef78ac1, 2018. Accessed 6 Decenber 2018.