INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(3):370-381, June, 2015.

Applications of Ubiquitous Sensor Network: Micro-Scale Air Quality Monitoring

S. Lee, S. Lim

Sangdong Lee
Korea Institute of Science Technology Information (KISTI)
245 Daehangno, Yuseong, Daejeon 305-806, Korea
sdlee@kisti.re.kr

Sang Boem Lim
Department of Advanced Technology Fusion and Division of Interdisciplinary Studies
Konkuk University, Seoul, Korea
1 Hwayang-dong, Kwangjin-gu
Seoul 143-701, Korea
*Corresponding author: sblim@konkuk.ac.kr

Abstract: The rise of densely populated societies in metropolitan areas has led to several forms of pollution, thereby leading to increasing health concerns. Among the various types of pollution, air pollution has become a crucial factor in the quality of life. Currently, the meteorological administration in Korea provides a regional scale of air quality index. However, this broad-based information is not relevant to local urban areas. We believe that air quality information about local environments such as homes, workplaces, and parks would be more relevant to people. We have developed a micro-scale air quality monitoring system called AirScope that covers local environments using a ubiquitous air quality sensor network. In this study, we focus on the applications of AirScope. We have developed AirScope as a general-purpose desktop PC application as well as a mobile application that can increase user mobility and provide real-time air quality information about the local user environment.
Keywords: Air Pollution Monitoring, USN, AirScope, Mobile Application

1 Introduction

The rise of densely populated societies in metropolitan areas has led to several forms of pollution, thereby leading to increasing health concerns. Among the various types of pollution, air pollution has become a crucial factor in the quality of life. However, currently, there is no public access to information about air quality. Efforts that provide air quality information and hence air pollution warnings are restricted to very few areas. A majority of such information is provided as an air quality index for entire cities. Such an index cannot provide information that adequately represents the air quality in local urban areas in a city.

Currently, the meteorological administration in Korea provides a regional scale of air quality index. However, this broad-based information is not relevant to local urban areas. We believe that air quality information about local environments such as home, workplaces, and parks would be more relevant to people. To this end, we have developed a micro-scale air quality monitoring system called AirScope that covers local environments using a ubiquitous air quality sensor network. Our previous studies have already provided a general overview of the AirScope system [1] [2]. A primary feature of AirScope is that it can be run as a smartphone application. Smartphones provide the advantages of mobility and access to information from anywhere at any time. Further, applications for smartphones directly target the end-user, and thus, the smartphone is particularly suited for providing air quality information.

In this study, we focus on the applications of AirScope. We have developed AirScope as a general-purpose desktop PC application as well as a mobile application that can increase user mobility and provide real-time air quality information about the local user environment. These applications are examples of AirScope usage which can be adopted as a service of company that is interested in real-time air quality service. In order to provide our services to the other research group or companies, we offer open API of AirScope. This paper is mainly demonstration of usage of AirScope using its Open API.

The desktop and smartphone user applications are discussed in section 2, while section 3 provides a brief introduction to AirScope. Two of the most important sub-systems of AirScope Local AirScope and Global AirScope are explained in sections 4 and 5.

2 User Applications

Our desktop application is designed to provide a wide variety of air quality information services. On the other hand, our mobile application is designed to increase user mobility and provide real-time location-based air quality information. Common services for both the desktop and mobile applications include maps, charts, and 3D services. The desktop application provides a rich set of graphics, useful environment information, and high 3D performance, while the strength of the smartphone application is its location-based service. In the study, we have selected iPhone, which provides GPS services, as our choice of smartphone. The application uses GPS information to provide location-based AirScope services. This means that when a user launches the AirScope application using his/her smartphone, the map service automatically detects his/her current location and displays it on the map.

2.1 Desktop Application

The desktop application provides real-time data services using maps, one-day data services using charts, data search and bookmark services, and local environmental information. Fig. 1 shows the AirScope desktop portal interface.

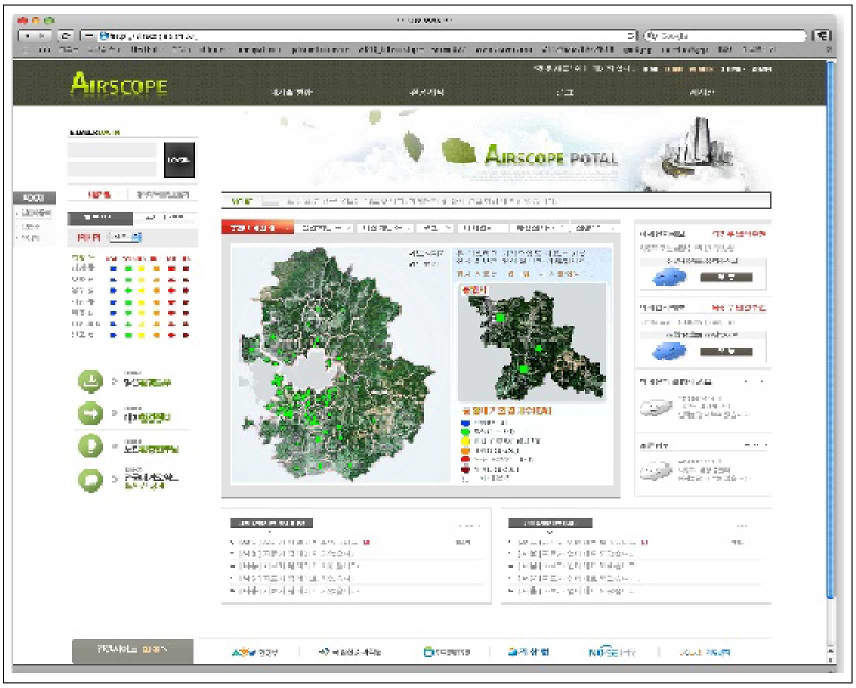

Figure 1: Desktop Portal Interface of AirScope

In the study, we have used Google Maps [3] for real-time data services. A marker that is positioned at the top of the map indicates the location of the air quality sensors. The portal requests data regarding the sensor location from Global AirScope when the real-time data service is launched. Global AirScope replies to the portal with the name of the location, longitude and latitude of sensor, and current sensor data. Upon receiving this information, the AirScope portal displays a marker at the top of the map. When the user clicks on this marker, the current air quality information is displayed by SmartWindow (see Fig. 2).

SmartWindow shows an image of the building in which the sensor is located and the current value of temperature, humidity, CO and CO2 concentrations, and PM-10 pollution levels. Further, SmartWindow provides the local environment data for the previous 24 hours when the user clicks on the average button. Fig. 3 shows the temperature data for the past 24 hours.

Figure 2: Display of real-time sensor data in SmartWindow

The periodic air quality data service (Fig. 4) provides air quality information over certain time periods. The goal of this service is to provide a general idea of the amount of pollutants for specific areas; thus, the user can compare the amount of pollutants between different areas. In order to use this service, the user begins by choosing the relevant start and end data. The user can select one or more sensor locations and the pollutant whose concentration he/she wants to check. The selection provides the user with a bar graph of the concentrations of the selected pollutant for comparison. Further, the application provides a line graph of the pollutant for online access. This graph allows the user to study changes in the amount of pollutants. The iPhone application also provides the same information.

The difficulty in identifying a single location on a map is addressed by the AirScope systems bookmarker service; the bookmarker service allows faster search and easier map navigation. This service works similar to the bookmark feature in any Internet browser. A user can bookmark his/her choices of locations in the AirScope portal by create a location folder. Subsequently, he/she can click on a location in the map and enter the location name. The user can revisit a location by simply clicking on the saved bookmark.

Figure 3: One-hour average data service for local sensor environment

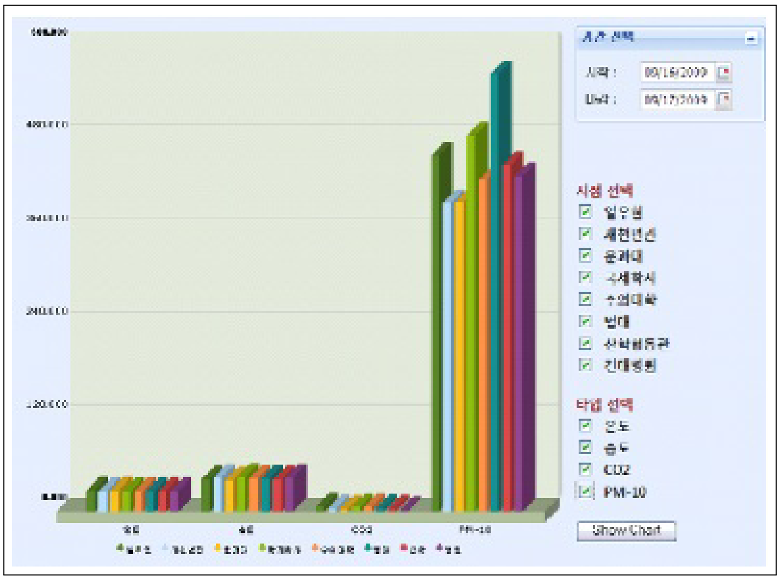

Figure 4: . Real-time monitoring service

2.2 Mobile AirScope

Until 2009, iPhone SDK 2.0 [4] did not support map API development. At that time, we developed the AirScope map service in the server by using Google Maps. In addition, we developed the web-based application with the iPhone browser. In 2010, iPhone SDK 3.0 [5] included map development API libraries called the MapKit framework. Our test results with SDK 3.0 showed a significant increase in process speeds and reliability of our application. Therefore, we redesigned and rebuilt the mobile AirScope application for iPhone SDK 3.0. This has resulted in maximized use of SDK 3.0s enhanced touch features and cursor movement speed when compared with our previous version. The loading speed of text and pictures is also enhanced because the browser component does not require loading. The application interface has been retained with increased reliability of the mobile application.

The smartphone application essentially provides the same functionalities that are available with the desktop application. The application includes real-time data services, sensor data chart services, and computational fluid dynamic (CFD) modeling (Fig. 5). A characteristic mobile service is a location-based user service. The GPS data from the smartphone is used for the map and notification services. The map service automatically displays the users current position and notifies him when the air quality in the environment changes.

Figure 5: Services of mobile AirScope

2.3 AirScope Open API Applications

Recent emergence of Web 2.0 [5], most of services are provided by the form of Open API [6]. Currently, various projects are using ad providing open API to fulfill openness and scalability of system. Project areas are including but not limited to mobile network API [7], enterprise application integration [8], and even in area of biology research [9] and Geographic Information System (GIS) research [10]. Particularly micro-scale environment services are used in variety of applications. To support these services and an API, we are providing Open API of our AirScope system.

A company, called Green Ecos, developed a mobile application using our open API. In this application, they provide real-time sensor data, historical chart, air quality Augmented Reality (AR) and Virtual Reality (VR) application of current air quality, social network service connection, and tracking of expose to harmful meterals. Other than social network service connection, they are using AirScope open API to provide services. Once they get sensor data using open Api, they will process data for their services. Fig. fig:gre shows UI of Green Ecos application.

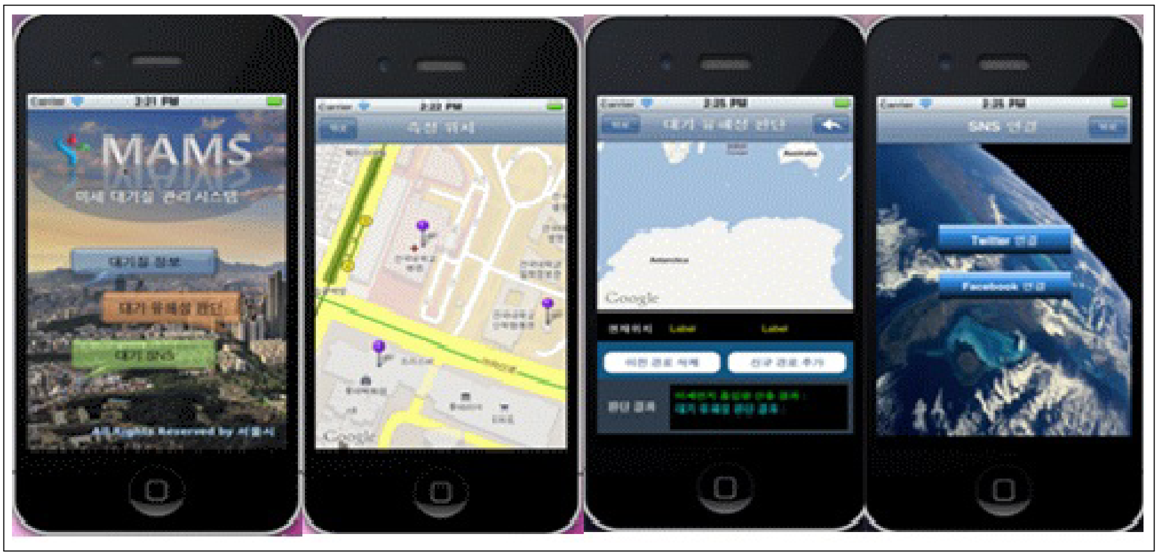

Figure 6: Green Ecos Application

3 AirScope System

In this section, we provide an overview of the AirScope architecture, including the specifications of sensors, communication between the systems involved, and the need and use of CFD modeling data. There are various ubiquitous applications using wireless sensor network researches [11] [12]. Each project has different goals. Former project targeted integrate development of an application between domain experts and network experts without specific knowledge of each others area. Later research uses bicycle to increase mobility. Our project targeted general audience to use our system. Local AirScope, Global AirScope, and their user applications are discussed in the following section.

3.1 General Architecture

Fig. 7 shows the general architecture of AirScope. Our system comprises three important components Local AirScope, Global AirScope, and their end-user applications. A portion of our study includes the installation of several sensors (approximately 1,000 to 10,000 sensors) in one district in Seoul (which comprises 25 districts). The impossibility of handling all the data by one server led to the plan of having one or more servers in a district (based on its size and population). We call this server as Local AirScope. It is expected to handle one area comprising the installed sensors and collect and process the sensor data in its operational area, store the data in a local database (DB), and manage the sensors. Local AirScope can solve the problem of server overloading. However, we require a mechanism to manage and gather information from Local AirScope to provide end-user services. Global AirScope is used to address this issue. This server manages Local AirScopes, distributes queries to obtain air quality information, processes the received data, and sends information to the user application. In this study, we have created the desktop portal and smartphone applications; however, there is no limit on the number and types of user applications. Our architecture has been designed to accept any user applications regardless of the platforms and programming languages used.

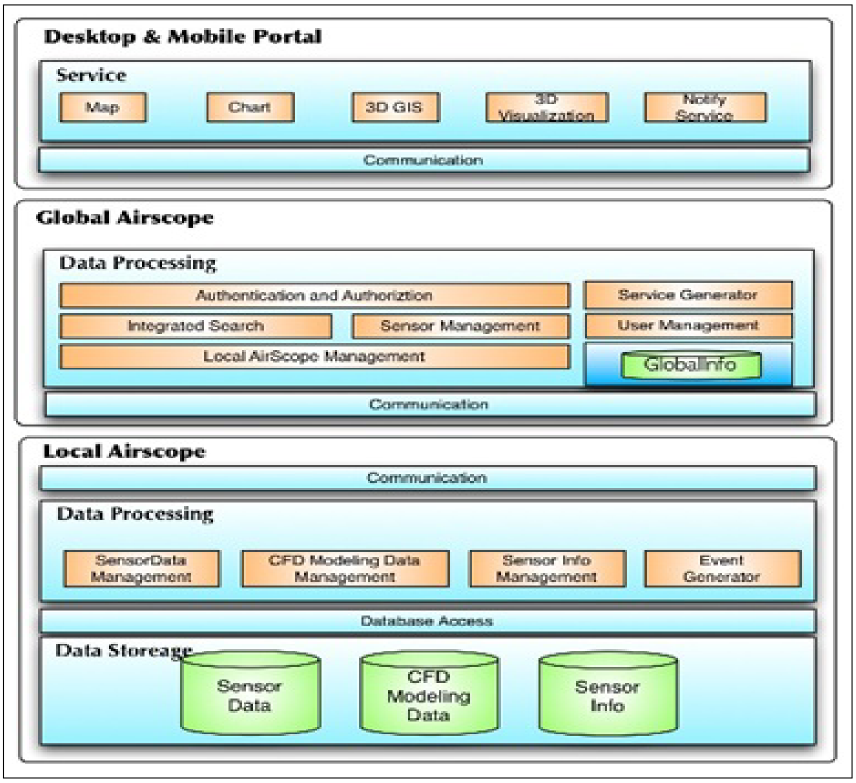

Figure 7: Architecture of AirScope system

4 Local AirScope

Local AirScope is a system that manages the collected data from the sensors and CFD model and provides requested data to the user to provide micro-scale air quality monitoring services. In our study, the purpose of this system was to parse and store data from both the sensors and the CFD model; further, it was required to manage sensor information such as name, type, and location. We assumed the presence of more than one Local AirScope system in the study because the actual implementation of the system will involve the installation of several thousand sensors in one of the districts in Seoul. The large quantities of data generated will require many servers. For the purposes of the study, we assume one Local AirScope system per district but the actual coverage of this system is expected to depend on the number of sensors in the area.

In the following section, we describe the architecture, data packet protocol, data processing, event generation, and DB schema of Local AirScope in detail.

4.1 Architecture of Local AirScope

Local AirScope system (Fig. 8) comprises three layers communication layer, data processing layer, and data storage layer. The communication layer manages communication between Local AirScope and Global AirScope and between Local AirScope and the sensor gateway. By means of this layer, one can send data requests and responses and obtain data from the gateway. The data processing layer is responsible for managing sensor data, CFD data, and sensor information.

All the actual sensor data, CFD modeling data, and sensor information are stored in the data storage layer.

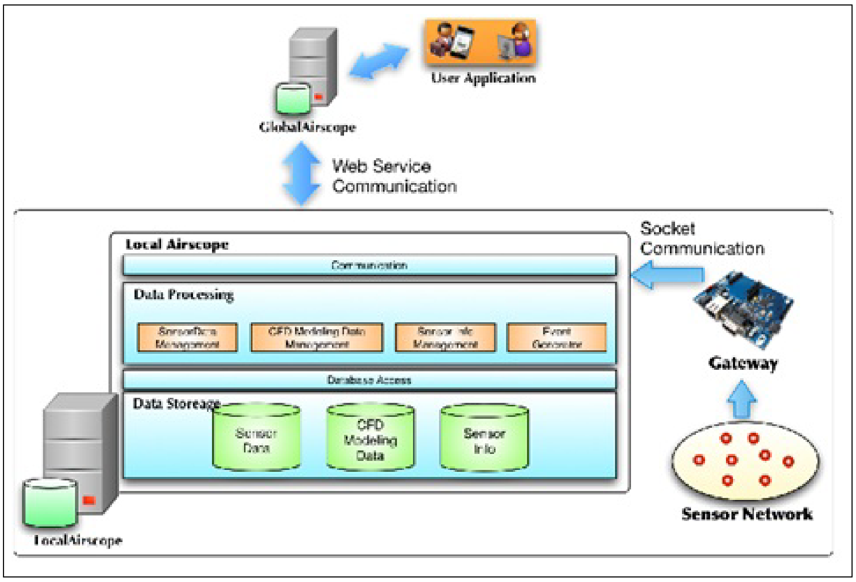

Figure 8: Local AirScope architecture

The communication layer uses Web Service and Socket as components of the communication module. We have explained the Web Service communication module in a previous section. The Web Service (RESTful) is used for communication with Global AirScope. Among the current Web Service technologies, we used a light weight communication package with a light weight data format called REST and JSON, respectively. The Socket communication module is used between Local AirScope and the sensor gateway to collect sensor data. All the sensors send data to the sensor gateway which in turn sends data to Local AirScope. The reason for the use of socket communication is due to the sensor gateway that uses embedded Linux as its operating system; embedded Linux does not provide any communication method other than socket communication.

The data processing layer contains management modules for sensor data, CFD modeling data, sensor information, and event processing. The main task of the sensor data management module is to obtain sensor data from the communication module, store data in the server DB, and respond to data requests from Global AirScope. The sensor data from the communication module is in the form of 39-byte hexadecimal raw data. This data cannot be understood by humans unless it is processed according to the sensor manufacturers protocol. The sensor data management module processes the incoming raw sensor data (Fig. 9) that contains temperature, humidity, CO, CO2, and PM-10 information into meaningful data and stores this processed information in the sensor DB. This raw data also contains a sensor ID to pinpoint the data source. This ID acts as a prime key to the DB.

The CFD modeling data management module manages data from the CFD modeling that can be considered as data from virtual sensors in order to overcome the shortage of sensors in our test-bed. This allows us to provide micro-scale air quality information without the installation of many actual sensors. The virtual sensor data is obtained in double number formats (e.g., 0.495508E-10, 0647687E-11) that need to be processed in order to be understood. The CFD modeling data management module processes this data and stores it in the CFD modeling DB. In addition, this module responds to data requests from Global AirScope for locations that do not have actual sensors.

When a user registers with our system through the Internet, his/her user id, name, age,

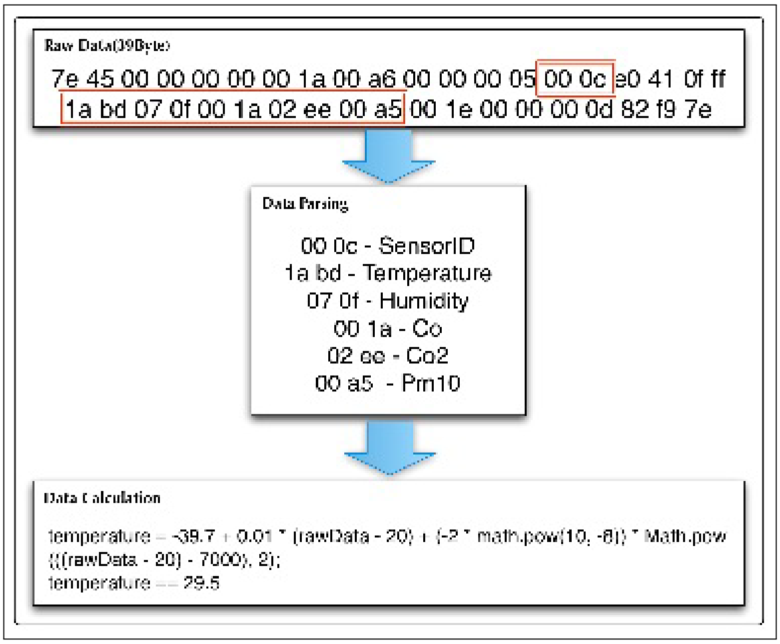

Figure 9: Data parsing

favorite location information, and his/her related medical history are collected. This information is managed by the sensor information management module. The favorite location is used to display a map of the environment when a user logs into the system. Consequently, users do not have to recall their list of most frequented locations or their locations of interest every time. The users age, air-quality related ailments, and his/her favorite location information are used to alert the user about air quality via the smartphone application. Our definition of an event is the occurrence of any unexpected circumstances such as those outside the upper and lower bounds of sensor data or the complete lack of any data from the sensor(s). Such events require to be reported to the system administrator in order to correct the problem. When such an event occurs, the event generator module generates a proper message that contains the event information and sends it to the system administrator. Further, this event generator can be activated when crucial events like fire occur.

The data storage layer comprises three DBs that hold the actual data. Each DB is described in detail in the following section.

5 Global AirScope

The main task of Global AirScope is to provide integrated query and management of the distributed Local AirScope servers and manage sensor and user information. In the early stage of our project, we did not design or accommodate a Global AirScope because our target area (the test-bed) was limited to a small number of sensors (Fig. 10). Therefore, all the system functionalities have been integrated into one Sensor Data Management System. Once our idea is implemented in the Seoul metro area, we plan to divide our system functionalities into two systems Local AirScope to manage sensors and sensor data for small areas and Global AirScope to manage the various Local AirScopes and user information. In this section, we focus on the architecture and the various modules of Global AirScope.

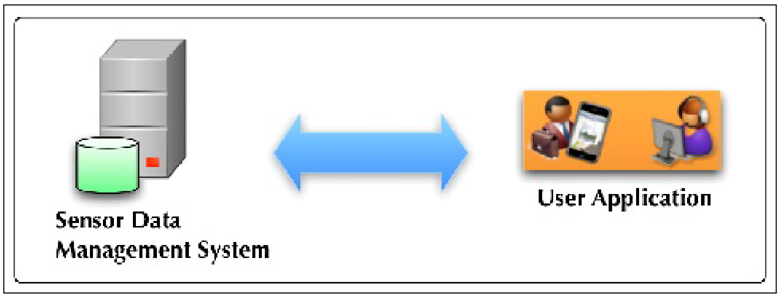

Figure 10: System architecture for existing sensor network

5.1 Architecture of Global AirScope

The main functionalities of Global AirScope include user authentication and authorization, integrated search capability, sensor management, and management of Local AirScopes (Fig. 11). Further, a service creator is added for user notification services. Global AirScope is queried for a service by users. In order to respond to the query, it collects data from Local AirScopes. Global AirScope integrates and processes the collected data and sends it back to the users device.

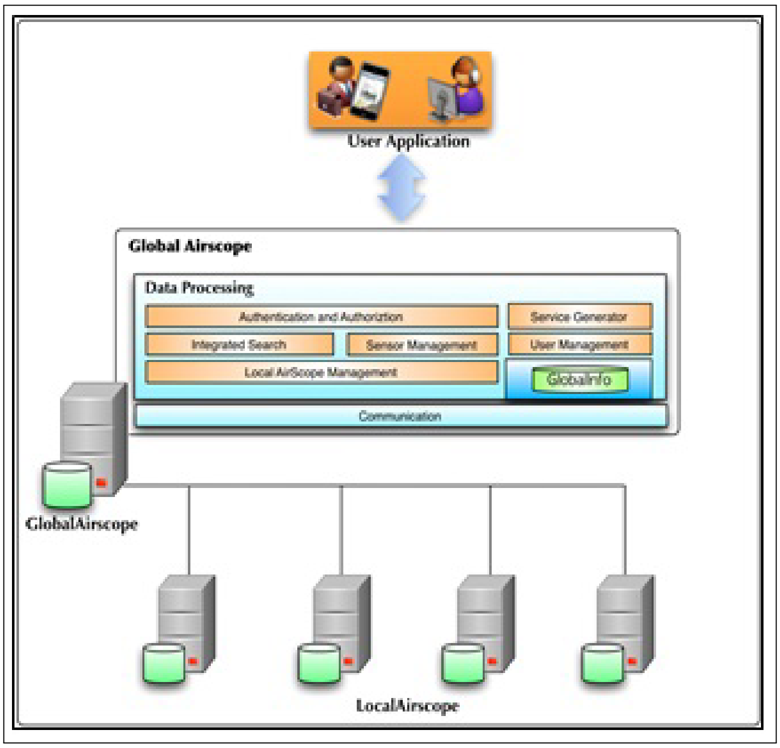

Figure 11: Global AirScope architecture

One of the advantages of the system is the fact that users can use it without registration. However, users are provided with richer personalized services such as user notification and bookmark services upon registration; the registration procedure is simple and easy to use. In addition to general registration, we are collecting any disease information to provide personalized health service. This information can provide health alerts based on the air pollutant rates in his/her current location. Further, users can bookmark their favorite or most frequented locations. When a user logs in to our system, the map automatically displays his/her residence and the surrounding area.

Because users may request for data from multiple Local AirScopes, we designed Global AirScope to perform and respond to distributed queries. This design provides better user accessibility. When data from multiple Local AirScopes are requested, Global AirScope queries the multiple Local AirScopes and obtains and integrates the relevant data. The integrated data is sent back to the user.

Global AirScope manages Local AirScopes because it receives all the connecting information between Local AirScopes. The main purpose of Local AirScope management service is sensor management [13] [11] (such as automatic sensor failure detection and notification to the system administrator). The manual handling of data from large numbers of sensors is almost impossible, and it may cause critical problems. Under abnormal circumstances such as sensor system failure or environmental disasters, a sensor generates erroneous data.

The event generator module of Local AirScope has been mentioned in a previous section of the paper. If data outside the upper and lower bounds of the sensor are detected, Local AirScope generates an event and sends to it to Global AirScope for user notification. Based on the event, Global AirScope generates a personalized message by utilizing the user information collected during registration; the age, sex, and the relevant medical history of the user are utilized to generate a personal message. This service is designed to send the message to the web portal on a desktop PC and to mobile devices including smartphones, feature phones, and iPads. The user can also receive the message via a windows widget and pop-up windows on the desktop PC without logging into the portal. The message format is generated to suit the capabilities of the users mobile device. For example, we can send a text message to a feature phone, while a combination of text and pictures can be sent to a smartphone.

6 Conclusions

We believe that air quality information about local environments such as homes, workplaces, and parks would be more relevant to people. To this end, we have developed a micro-scale air quality monitoring system called AirScope that covers local environments using a ubiquitous air quality sensor network. A primary feature of AirScope is that it can be run as a smartphone application. Smartphones provide the advantages of mobility and access to information from anywhere at any time. Further, applications for smartphones directly target the end-user, and thus, the smartphone is particularly suited for providing air quality information.

Our desktop application is designed to provide a wide variety of air quality information services. On the other hand, our mobile application is designed to increase user mobility and provide real-time location-based air quality information. Common services for both the desktop and mobile applications include maps, charts, and 3D services. The desktop application provides a rich set of graphics, useful environment information, and high 3D performance, while the strength of the smartphone application is its location-based service.

We provide an overview of the AirScope architecture, including the specifications of sensors, communication between the systems involved, and the need and use of CFD modeling data. Local AirScope is a system that manages the collected data from the sensors and CFD model and provides requested data to the user to provide micro-scale air quality monitoring services. In our study, the purpose of this system was to parse and store data from both the sensors and the CFD mode. The main task of Global AirScope is to provide integrated query and management of the distributed Local AirScope servers and manage sensor and user information.

Bibliography

[1] Woo, J. et al (2011); Constructing u-City of Seoul by future foresight analysis, *Concurrency and Computation: Practice and Experience*, Wiley, ISSN 1532-0626, 23(10):1114-1126.

[2] Woo, J. et al (2011); AirScope: A Micro-scale Urban Air Quality Management System, *Lecture Notes in Computer Science*, Springer, ISBN 978-3-642-24649-4, 520-527.

[3] http://code.google.com.

[4] http://developer.apple.com.

[5] Oreilly, T. (2007); What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, *Communications & Strategies*, Digiworld, ISSN 1157-8637, 1:17- 37.

[6] Choi, M. et al(2012); Open API Design for Research Information System, *Lecture Notes in Electrical Engineering*, Springer, ISSN: 1876-1100, 181:187-195.

[7] Liu, K.; Xu, K. (2012); Open Service-Aware Mobile Network API for 3rd Party Control of Network QoS, *Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 172-175.

[8] Lu, W.; Liu, Y. (2011); Research of Event-Driven Enterprise Application Integration Base on Service Oriented Architecture, *Applied Informatics and Communication Communications in Computer and Information Science*, Springer, ISSN: 1865-0929, 224:33-37

[9] O'Donovan, D.J.et al (2012); A grid-enabled web service for low-resolution crystal structure refinement, *Acta Crystallographica Section D, Biological Crystallography*, ISSN 0907-4449, 68(3):261-267.

[10] Botts, M. et al (2008); OGC Sensor Web Enablement: Overview and High Level Architecture, *GeoSensor Networks*, Springer, ISBN 978-3-540-79996-2, 4540:175-190.

[11] Rodrigues, T. et al (2012); A Model-Based Approach for Building Ubiquitous Applications Based on Wireless Sensor Network, *Mobile and Ubiquitous Systems*, Springer, ISBN 978-3-642-29154-8, 73:350-352

[12] Nakamura, T. st al (2012); Proposal of Web Framework for Ubiquitous Sensor Network and Its Trial Application Using NO2 Sensor Mounted on Bicycle, *Proceedings of the 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT)*, IEEE, ISBN: 978-1-4673-2001-6, 83-90.

[13] Suganthi, N; Vembu, S.; Energy Efficient Key Management Scheme for Wireless Sensor Networks, *International Journal of Computers Communications & Control (IJCCC)*, ISSN 1841-9844, 9(1):71-78.

[14] Jiang, N. et al (2014); Distributed Compressed Sensing Algorithm for Cluster Architectures of WSNs, *International Journal of Computers Communications & Control (IJCCC)*, ISSN 1841-9844, 9(4):430-438.