IoT-Based 3D Visualisation Platform for an Efficient Management of the Smart City Ecosystem

George Suciu, Lucian Necula, Teodora Ușurelu
Research & Development
Beia Consult International
Bucharest, Romania
george@beia.ro, lucian.necula@beia.ro, teodora.usurelu@beia.ro

Ioana Rogojanu, Maria-Cristina Dițu
Applied Electronics
University Politehnica of Bucharest
Bucharest, Romania
ioana.rogojanu@beia.ro, maria.ditu@beia.ro

***Abstract*— Nowadays, most of the urban developments are based on ICT (Information and Communication Technology) solutions which can increase the socio-economic development the quality of life. Smart Cities have been pointed out as the most promising applications of Internet of Things (IoT). Despite all standardization processes, building a Smart City IoT platform remains a challenge. The continuous changes in IoT environment require the system to adopt new technologies dynamically. To address this challenge, our proposal is the deployment of the CitiSim platform which can simulate 3D models for events processing and supports different Operating Systems (OS) enabling new business models. The main advantage of the CitiSim platform is that it enables Big Data analytics tools to monitor data coming from heterogeneous sources. The primary objective is the identification of the environmental, social, and economic indicators used in order to build smart urban models that contribute to the process of decision making for improving land use, transport, energy, and user needs, classified and evaluated with respect to the criteria of a specific district.**

Keywords—IoT, Smart City, 3D visualization, CitiSim

I. Introduction

In the past decade, the introduction of ICT technologies in our daily activities has opened up new pathways of cooperation and organization. The society is evolving towards a new digital era where citizens will become more and more connected to the environment. Cities are the core of the society and they will need to embrace and support the evolution [1]. As a consequence of this process, a significant number of laboratory studies and research projects have been developed to study how to deal with problems such as pollution, traffic, energy consumption, mobility etc. [2]. A digital ecosystem is growing within the Smart City concept which has an estimated market worth of $1201.69 billion by 2022 [3]. However, along with the new protocols, technologies and services utilized in smart cities, the service deployment and the service development, in the context of Smart City monitoring and control, are likely to be problematic for citizens, companies and city councils [4]. The absence of a platform with advanced tools and services for the Smart City is hindering the practical development of the associated ecosystem. This type of ICT platform, also named as e-infrastructure, will be an important regulator for the future development of cities as energy infrastructure and transport infrastructure have been in past decades. Under this context, CitiSim's project [5] main objective is devoted to the implementation of a new generation platform for the ecosystems in smart cities. This platform will provide control infrastructure and a powerful monitoring in order to enable planners to make critical management decisions on strategic and tactical levels based on the knowledge provided by the specific platform developed. For a natural interaction and better understanding of the events that happened in the city, 3D visualization techniques as augmented virtuality and augmented reality will be explored. CitiSim [6] will provide a set of services, standards and tools for the applications' development of the Smart City. In the context of the CitiSim project, the paper provides an overview of an innovative platform for efficient monitoring and management of the Smart City environment.

The rest of the paper is organized as follows: Section II provides an overview of Smart City management solutions available on the market, Section III describes the conceptual architecture of the CitiSim platform, Section IV presents the CitiSim common framework, while Section V concludes the paper and foresees future work.

II. Related Work

Smart Cities are defined by the European Commission [7] as cities that integrate technologies for energy management and transport optimization, and other IT&C technologies for enhancing the quality of citizens' lives and protecting the environment. Thus, Smart Cities address social, environmental and economic goals.

The concept of a Smart City is probably one of the most significant challenges for the IoT applications as in the recent years this concept was involved in the expansions of several platforms. The main reason is given by the increasing number of population and the need to integrate technologies that improve the quality of life [8].

VITAL [8] is a Cloud of Things (CoT) Smart City platform that aims to fulfil most of the needs for a Smart City platform. The purpose of this platform is to integrate ICOs (Inter-Connected Objects) into IoT environments. This project considers aspects such as the cost of deployment. Moreover, VITAL brings into attention the concept of virtualization of interfaces that enables the management of heterogeneous objects.

This platform allows the implementation of Virtualized Universal Access Interfaces (VUAIs) which makes VITAL part of CoT central concept. Interconnection of different IoT platform is built using meta-architecture and migration layer, making the VITAL platform go beyond the typical definition of IoT concept as it includes the possibility of linking geographical data [9].

The idea of a Smart City can be found in VITAL concerning energy, transport, ways of spending free-time or the management of resources. This concept aims to report problems such as accidents or crimes, bins that need to be emptied, pollution or traffic congestion. VITAL is also used for smart street management as it interconnects data coming from Clouds notifying for example where are trash containers that need to be deployed [8].

Over time, the climate has undergone changes that have led not only to an increasing annual average temperature [10] but also to the depletion of fossil fuels [11]. Thus, economic pressure has grown, municipalities are being forced to adapt to new conditions and to intelligent management measures at the city level. The main aim was to create a platform for increasing the potential for innovation and for optimization of services in emergency cases in order to ensure the safe operation of urban services [11].

Electricity is in the center of most infrastructures, interruptions or failures in the energy system with economic and social consequences. An intelligent control system for monitoring and controlling the energy system at the urban level, especially in the event of interruptions or failures has a big role in improving network performance in city emergencies and safety.

A Smart City must have the following benefits: drinking water and secure electric power, efficient public transport and management system, non-stop access to public data and open local economy.

Over the years, the term intelligent building has had different connotations. If in the 1980s a smart building was considered to be an energy efficient building, in the 1990s a smart building was a building whose infrastructure system was controlled by a computer. Nowadays, smart buildings have sensors, actuators networks, controllers, Internet connection, human-machine interfaces, which ensure an efficient monitoring of energy consumption and beyond [11]. In the field of power systems, a testbed for Supervisory control and data acquisition (SCADA) infrastructure will be able to simulate network properties, communications infrastructure and models of human’s behavior within the system. The proposed platform is intended to monitor emergencies for future energy systems. Phenomena like terrorism or natural disasters have a strong impact on the price of electricity and cannot be analyzed in real life, due to the fact that a real test would involve interrupting the system.

Questions arising from testbed development are related to monitoring and controlling the power system, how a human operator will respond to natural hazards, power outages, cyber-attacks, distributed resources and intelligent networks for improving security in a city. Within the proposed architecture, simulations can be developed with real-life data. The testbed should be composed of physical grid, communication infrastructure, information flow.

Typically, a smart city solution is integrated through the input/output interface. At the conceptual level, the system is integrated so that the data processing results are routed through the communications infrastructure, and then the calculations are carried out. Usually, the constraints are transmitted to the central operator. Thus, the components of the network can react while eliminating imbalances. The process is iterative and involves resuming the stages until the balance is reached at the final state. Entries are controlled by operators which have access to general results that can be filtered for a more in-depth review. At the same time, they control all parameters in the system. In this regard, an operator can see if a communications line has been compromised. The purpose of the testbed is to identify the hazards that may occur in the electrical network's stability and to predict the occurrence of a defect in the system. The testbed monitors the system’ status which is passed afterwards to the database to recognize their background. Therefore, the platform verifies the received data, and if the system is in a critical state, then it is acting accordingly, with the implementation of loss recovery or runtime algorithms in cases of extreme urgency.

III. CitiSim Proposed System Architecture

The architecture proposal for CitiSim should be flexible enough to support different operating systems (Windows [12], Linux [13], Android [14], etc.) and different programming environments (Java [15], C++ [16], etc.). REpresentational State Transfer (REST) Application Programming Interfaces (APIs) should do the access to those services of CitiSim from third parties through Internet according to the current State of the Art. The internal middleware used in core service implementation should be efficient enough to support scalability, security issues, flexibility, etc.

The CitiSim architecture is presented in Fig. 1.

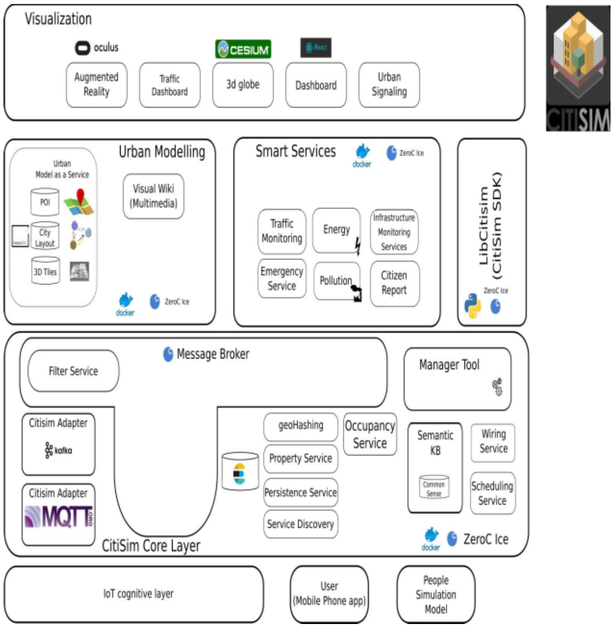

Fig. 1. CitiSim conceptual architecture

IoT Layer: This component is a layer where all the information is collected from sensors/actuators. The sensors will register in the message broker through an interface as publisher and each sensor will call message broker to interface periodically with its value and some metadata about reading (e.g. Expiring time, Timestamp, quality of the reading, etc.). The actuators will implement a simple interface for the change of the environment. In the simulation plane, the simulation algorithms will use the same method. From a practical point of view, a service in the upper layer does not distinguish among real data and simulated data.

However, time critical aspects should be taken into consideration [17].

User mobile phone app: This application enables the user to access the information provided by the CitiSim platform. For example, this component can be used to show information about traffic.

Core Layer: The core layer architecture represents the CitiSim [5] [6] platform itself. It is used if a city runs a single instance of the core layer in order to support the rest of the services. The key components of the core layer are:

- Message Broker: The message broker is the main component of the CitiSim [5] [6] system architecture which is used for information (raw data, events, sensor information, etc.) distribution among core services and with the smart services layer. In the message broker, each service has to subscribe to specific topics where information is published and, of course, a service also has to publish the information generated. A topic is a logical bus where information is published. An interface defines the types of the messages. For example, from the IoT layer, several sensors sending information related to true/false information will use the following interface: void notify (Boolean value, string source, Metadata). All the services interested in this type of information will implement the notify method so the message broker will call all subscribed services by calling each notify method of each service;

- Filter Service: In order to scale properly, a filter service is defined in order to subscribe specific topic with some filter options. In this way, only information that pass all the filters defined by each service will be communicated. The filters will be set up on subscription phase;

- Property Service: The property service is a service devoted to store static/semi-static properties of devices/services in an instance of CitiSim platform [5] [6]. For example, the position of a smoke sensor, the last revision of an extinctor, the manufacturer of specific actuators are examples of information stored/accessed through the property service. The property service will store all this information in a data store;

- Persistent service: This service is subscribed to all topics in the message broker and store, in the data store. This persistent service will store and compact the information about the city;

- Semantic KB: this knowledge database will store three types of information: (1) the vocabulary and relations of concepts in a current city. This semantic information is common to any developed city. The services will use this vocabulary in its metadata, interfaces, etc., (2) the rules about how a city works regarding traffic and pedestrian and (3) a service description of the instances running in this instance of CitiSim;

- Scheduling service: This experimental service will orchestrate complex behaviors according to the services deployed and the new desires expressed by a user or service. For example, the authorization (literally open the door) to go into a building can be done by several methods (facial recognition, radio-frequency identification (RFID) tag, personal identification number (PIN) code, etc.) according to the deployed ICT infrastructure. This scheduling service could link the access methods with the Authorization/Access service dynamically under an access request done by a user;

- Semantic Service: This service will manage information at the semantic level and it will integrate other domains with CitiSim domain;

- Manager Tool: A tool for monitoring the CitiSim platform (state of the services/devices);

- Adapters: These modules will interconnect CitiSim (together with the semantic service) with other domains (e.g. MQTT devices, Sofia2 services, etc.).

Urban modelling layer: This layer will store structural information about the city. The key components of this layer are:

- Visual Wiki: This is information (including multimedia) about the city in different formats;

- Urban model as a service: This component is a repository with different models with information about urban furniture, street layout, 3D building models, supplies model (energy grid layout, water grid layout, etc.). The three initial repositories will be a Point of Interest repository with specific information about monuments, cultural buildings, etc. devoted to tourist applications. A street layout with a graph with the streets and pedestrian paths and a 3D tiles repository for 3D virtual world construction. The idea is to offer this information through an API to build visualizers accessing remotely to the information and also to provide information updates by a push model (e.g. street cut by some unusual event, water leak and affected area, etc.).

Smart service layer: The smart service layer takes the information collected in the core layer, and by using the urban model layer provides a service related to stakeholders of a Smart City.

The services planned in CitiSim smart layer are extracted from the use case defined in the consortium. Third party services can be modelled and implemented attending to different use cases. From the currently defined use cases, the following smart services will be performed:

- Pollution, Energy and Infrastructure monitoring service: A general monitoring service provides the information related to a specific domain and enables the control of devices in this field. In the case of energy, the monitoring service will enable/operate/control specific devices to turn on/off sections of the energy infrastructure;

- Cultural agenda: This service will enable to introduce cultural event information about the city from a central point. This central point will act as the primary source of information for visualizers, apps, etc;

- People monitoring: The key idea of this service is to estimate, from different sources of information (e.g. video analysis, sensors, access services, etc.), the occupancy of different areas at specific times. This service will store, with different accuracy, the number of people in particular areas;
- Traffic monitoring: In a similar way to "People monitoring", this service will collect information about the traffic conditions in the street layout;
- Emergency service: This smart service will follow a set of steps when a specific emergency is detected. In the use case of Spanish consortium, this emergency service will provide with evacuation paths according to the type of emergency and the status of the infrastructure;
- People Simulation model: In the field of simulation models, this service will provide information regarding simulations of movements of virtual people. For example, this service can be used to estimate the time of people to follow an evacuation path.

The set of services of the smart service layer is not closed, according to the evolution of the project, it could be possible to split services in more functional one or implement new services if needed.

Finally, the visualization layer provides to final users (mayor, citizens, companies, etc.) with information regarding different aspects of the data managed in a CitiSim instance.

IV. CitiSim Common Framework

A. Smart services to be implemented

The main objective is to provide a development environment for event processing. This is a method of tracking and analysing streams of information about things that happen (events), and deriving a conclusion from them.

Complex event processing (CEP) is event processing that combines data from multiple sources to infer events or patterns which suggest more complicated situations. The goal is to identify relevant events and respond to them in an efficient way.

These events may be happening across the layers of CitiSim (sensor layer, preprocessed data, etc.). An event can also be defined as a "change of state". CEP will provide to CitiSim a new way to analyze patterns in real-time and will help the business side communication with IT and service layers. A top layer offers REST services to other IoT clients or the analysis layer.

Fig. 2 presents how the information flows from the bottom layer where the sensors generate events to the top layer where this information is consumed.

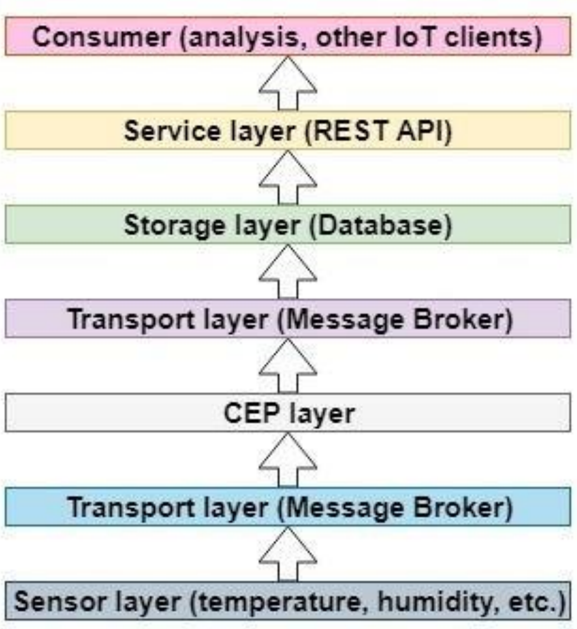
Consumer (analysis, other IoT clients)
Service layer (REST API)
Storage layer (Database)
Transport layer (Message Broker)
CEP layer
Transport layer (Message Broker)
Sensor layer (temperature, humidity, etc.)

Fig. 2. The flow of information from the sensor layer to the consumer layer

The data flow starts from the sensors and it's received by the complex event processor, which throws alerts and are received by the server. This server stores the meaningful data and send the alert to the client via Web Sockets. Moreover, the REST API will have data available. The data flow between the components for the CitiSim platform is presented in Fig. 3.

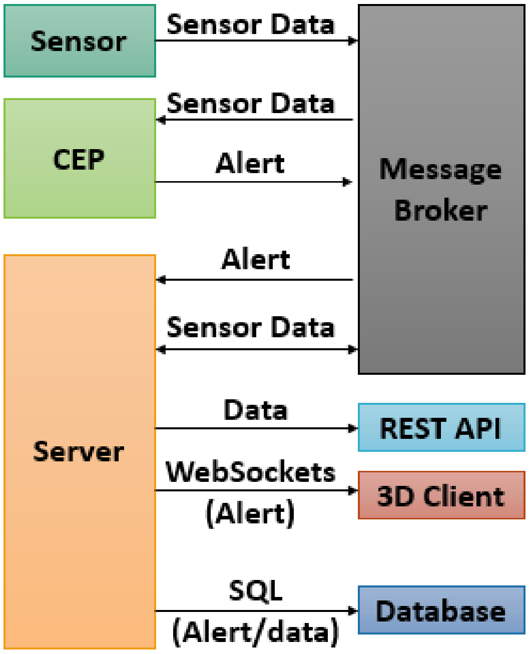

Fig. 3. Data flow between components

B. Identifying of Development languages and API's

Each module of the CitiSim platform is developed with a different technology appropriate for it. Fig. 4 depicts the modules of the CitiSim solution.

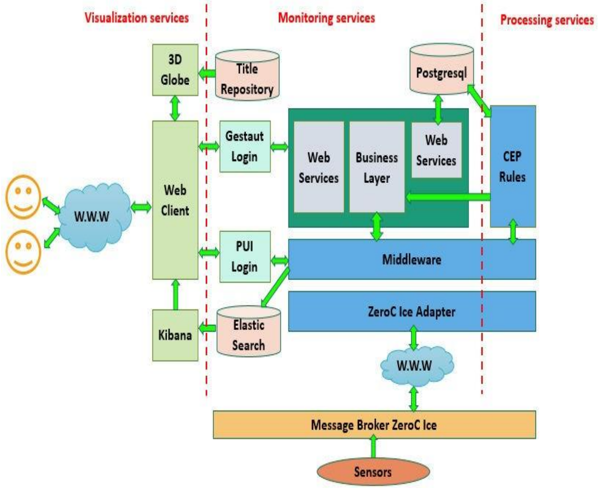

Fig. 4. Modules of the CitiSim platform

Following the figure presented below, modules are grouped by technology:

- Applications server (PUI Server) and Middleware: Mainly developed in Java. Supports interchange formats like JSON, CSV or XML and use technologies as JavaEE, Spring and PUI;
- Complex Event Processing: Developed with WSO2 framework and Siddhi language;
- General Database: PostgreSQL database and SQL as a consulting language;
- Web clients: Developed with PUI client technology in JavaScript. Several frameworks or libraries as requirejs, jquery, knockout are used;
- Message Broker: ZeroC Ice using RPC. It supports several languages as C++, C#, Java, JavaScript or Python.

This middleware provides a bridge to the application server or other clients. This bridge is an API that simplifies the way to access data and how it is synchronized with the message broker. Thus, it provides a solution that improves the Quality of Service (QoS), as well as the security, the message sending, the updating of the service directory and so on.

This middleware can work with different message brokers and for each one of them, it needs a specific driver. In CitiSim, it is useful to allow the clients to be independent of the message broker technology. Thus, if we change the message broker in order to adapt the project results to another platform, it won't be necessary to modify clients' implementation.

The IoT interfaces and the information model will be common for the smart service layer. There are six basic interfaces devoted to basic sensors/actuators:

- The first interface called Observable will enable CitiSim to link services in an automatic way configuring a set of services interested in the information generated by a specific service. Together with semantic information, the service scheduler will use this interface to configure an under-demand chain of services to provide new advanced services not previously specified;
- The DigitalSink interface will be called by any sensor with a digital value (e.g. on/off). In the same way, an actuator with two possible states (e.g. on/off, true/false) will implement this interface;
- The DataSink is devoted to sensors/actuators with a complex data structure (e.g. a video camera) where such data structure is injected in the byte sequence data;
- The AnalogSink is the equivalent to the DigitalSink but with a float value;
- The EventSink is a generic interface for event generation, the details about the event will be injected in the meta parameter (e.g. fire alarm). As we said before, the full set of type events will be published in function of alarms identified in the implementation phase;
- The AccelerometerSink is a special type of interface for accelerometers.

That information will be used to the 3D model to visualize the elements of emergencies geolocated in a map, i.e. sensors, emergency exits, rooms, corridors, etc. In addition, the readings of the sensors placed in the building by the smart glasses could be viewed in real time during the simulation phase.

Finally, the simulation will be able to show the user's instructions on first aid so that the user who is using an Augmented Reality device can attend to the injured through the information that is shown through the device.

V. Conclusions

The CitiSim competitive advantage comes from the combination of a Smart City platform that supports the development and the simulation of smart services in conjunction with an elastic monitoring infrastructure that is based on a potentially huge number of heterogeneous things deployed over a city. CitiSim project will implement several use cases to demonstrate the outcomes of the project related to, but not limited to, the domains of smart tourism, smart telemetry, smart emergencies, and smart transport. The Smart City market is in continuous growth and has a clear impact on the economy and business. The benefits of the exploitation of CitiSim project are destined both to society (citizens) and to economy (markets value, employment, spin- offs). CitiSim aims to provide means for users (e.g. citizens, tourists, and marketers) to build their own services, evaluate them in a real, changing environment, observe their progress, and interact with other actors of the simulation (things). Thus, the major innovation of CitiSim project is to provide the first Smart City-specific platform to monitor in real time and in 2D/3D a large infrastructure of the city, and that enables to interact with users by developing/testing added-value and customized services in an agile and simple way.

ACKNOWLEDGMENT

This paper was partially supported by UEFISCDI Romania and MCI through projects CitiSim, Power2SME 3DSafeGuard, and funded in part by European Union's Horizon 2020 research and innovation program under grant agreements No. 777996 (SealedGRID project) and No. 787002 (SAFECARE project).

REFERENCES

[1] G. Viale Pereira, G. Eibl, and P. Parycek, “The Role of Digital Technologies in Promoting Smart City Governance,” in Companion of the The Web Conference 2018 on The Web Conference 2018, International World Wide Web Conferences Steering Committee, pp. 911-914, April 2018.

[2] A. Caragliu, and C.F. Del Bo, “The impact of Smart City policies on urban innovation. Technological Forecasting and Social Change,” in Technological Forecasting and Social Change, July 2018.

[3] MarketsandMarkets. (2017). Smart Cities Market by Focus Areas, Transportation (Types, Solutions, Services), Utilities (Types, Solutions, Services), Buildings (Types, Solutions, Services), Citizen Services (Types), and Region - Global Forecast to 2022. Retrieved on August 2018 from: https://www.marketsandmarkets.com/Market-Reports/smart-cities-market-542.

[4] K. Borsekova, and P. Nijkamp, “Smart cities: A challenge to research and policy analysis,” in Cities - The International Journal of Urban Policy and Planning, vol. 78, pp. 1-3, August 2018.

[5] CitiSim. (2018). Project description. Retrieved on July 2018 from http://citisim.org/project-description/.

[6] CitiSim. (2017). Project description. Retrieved on July 2018 from http://www.beiaro.eu/citisim/.

[7] European Commission, Directorate-General for Energy, “Analysing the potential for wide scale roll out of integrated Smart Cities and Communities solutions,” June 2016.

[8] R. Petrolo, V. Loscri, N. Mitton, “Towards a Smart City based on cloud of things, a survey on the Smart City vision and paradigms”, Transactions on Emerging Telecommunications Technologies, pp 1-11, 2015.

[9] VITAL. (2018). VITAL R&D programme description, Retrieved on July 2018 from http://vital-iot.eu.

[10] C. Burnete, C. Tomescu, A. Popescu, I. Samoila, M. Dobrin, V. Rusu, G. Popescu, O. Falup, I. Bunda, C. Balaceanu, M. Balanescu, E. Dascalu, Romania’s 7th National Communication under the United Nations Framework Convention on climate change, UNFCCC, Bonn, Germany, 2017.

[11] L. Lugaric, S. Krajcar, Z. Simic, “Smart City - Platform for Emergent Phenomena Power System Testbed Simulator” in IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), pp 1-7, November 2010.

[12] Microsoft. (2018). Microsoft Windows description. Retrieved on June 2018 from https://www.microsoft.com/en-us/windows/features.

[13] Linux. (2018). Linux description. Retrieved on July 2018 from https://www.linux.org.

[14] Android. (2018). Android description. Retrieved on August 2018 from https://www.android.com.

[15] Java. (2018). Java description. Retrieved on August 2018 from https://www.java.com.

[16] C++. (2018). C++ language description. Retrieved on August 2018 from http://www.cplusplus.com.

[17] H. Zhou, A. Taal, S. Koulouzis, J. Wang, Y. Hu, G. Suciu, V. Poenaru, C. Laat, and Z. Zhao. "Dynamic Real-Time Infrastructure Planning and Deployment for Disaster Early Warning Systems." In International Conference on Computational Science, pp. 644-654. 2018.