# A Component-based Modeling of Edge Systems Computing

O.N. Sehout, M. Ghiat
University of Constantine2-Abdelhamid Mehri
Faculty of NTIC
{nassim.sehout, mouad.ghiat}@univ-constantine2.dz

Z. Benzadri, F. BELALA
LIRE Laboratory
University of Constantine2-Abdelhamid Mehri
{zakaria.benzadri, faiza.belala}@univ-constantine2.dz

## Abstract

Internet of Things (IoT) aims to bring every object, such as smart cameras, wearable devices, etc., online, hence generating massive amounts of data that can submerge storage systems. The Cloud computing paradigm is unable to meet these requirements, particularly in some critical applications (as health monitoring, emergency response, etc.) that require low latency, and delay caused by transferring data to the cloud and then back to the application can seriously impact their performances. To overcome this limitation, Edge paradigms, as Fog computing, Edge computing and Mobile Cloud computing, have been proposed, where cloud services are extended to the edge of the network to decrease the latency and network congestion. This paper proposes a software architecture supporting the modelling of Edge computing dedicated systems. The current situation of emerging Edge paradigms is analyzed in detail. An approach based on software component technology is proposed and applied using Fractal components model, which can specify semantics of every-changing dynamic and distributed integration of software components. Furthermore, a case study of the patient monitoring system (PMS-EDGE) along with its application model and network topology is expounded.

**Keywords** - Distributed Computing, Cloud computing, Edge computing, Component Model, Fractal, FractalAD, Patient Monitoring System.

## 1. Introduction

Recently, Cloud computing, has added a new dimension to the traditional means of computation, data storage, and service provisioning [Rajk17]. Indeed, the rapid increase in the number of ubiquitous mobile and sensing devices which are connected to the Internet, challenges the traditional network architecture of the cloud computing framework. IoT envisions a new world of connected devices and humans in which quality of life is enhanced, because management of city and its infrastructure is less cumbersome, health services are conveniently accessible, and disaster recovery is more efficient. A large number of devices are located at the edge of the network and require support for mobility, low latency, real-time, and location-aware services [Rajk17]. New computational paradigms, termed as Fog computing, edge computing and Mobile Cloud computing, that subdue the shortcomings of cloud computing by transferring some of the core functions of cloud towards the network edge, have emerged. Hence, the cloud computing paradigm is unable to meet certain requirements such as: the low latency and jitter, the context awareness, the mobility support, etc. However, these requirements are crucial for several current applications (e.g. vehicular networks, augmented reality). To fulfill these requirements, for instance, Edge computing, a distributed computing paradigm has emerged in recent years. It empowers the network devices at different hierarchical levels with various degrees of computational and storage capability, suitable for the services and applications of IoT [Rod18].

Few research attempts are devoted to the modelling and development of Edge systems dealing with their main characteristics, as the bandwidth/response time trade-off in traditional cloud computing systems with heterogeneous types of computation tasks and wearable mobile devices.

The main goal of this study is to holistically analyse the Edge computing architecture, a non-trivial extension of cloud computing [Sark16], which serves as a platform that bridges numerous sensing devices, situated at the network edge, to the core computing structure of the cloud. Then, we propose a generic software architecture supporting the most features of these systems. Our approach is driven by the Component-Oriented Software Development (COSD) [Hazl11] that has been recognized as a viable way of building software systems. Although there are many different views on what a component is and what its features are, common consensus regards a component as a black-box entity with well-defined interfaces and behavior, and emphasizes, as one of the key features, its reusability in different contexts.



In: Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering (ICAASE18), Constantine, Algeria, 1,2-December-2018, published at http://ceur-ws.org

Our contribution in this paper is based primarily on these strengths of COSD while considering software Edge systems as a composition of some hierarchical components, viewed as gray-box/glass-box entities with the internal structure visible as a set of communicating subcomponents. We present a realization of the proposed model for Edge computing systems by exploiting the component model Fractal [Hazl11]. Thus, a component Fractal is used as a software Edge element, conforming to its component model and being independently deployed and composed without modification according to a composition standard [Coup06]. Besides, Fractal components are reflective, in the sense that their execution and their internal structure can be made explicit and controlled through well-defined interfaces. These reflective capabilities, however, are not fixed in the model but can be extended and adapted to fit the programmer's constraints and objectives.

The remaining of the paper is organized as follows. Section 2 provides a comprehensive outline of the Edge computing architecture and analyses the performance of this paradigm in contrast with that of the traditional cloud computing framework. In Section 3, the architectural modelling of the Edge computing dedicated system is presented. We design the different software components using FRACTAL model. The applicability and execution of our proposed model is demonstrated in Section 4 through a realistic case study, the Patient Monitoring System "PMS-EDGE" before finally concluding the work in Section 5.

## 2. EDGE COMPUTING: Principles and Definition

Cloud Computing has emerged as a new technology and an infrastructure in which storage and computing power are managed by remote servers to which users connect via a secure internet connection.
Nowadays, it becomes one of the most famous word that it is used by virtually all modern businesses. This trend is also used to market many types of software and network services facilitating further our daily life. Cloud computing, in general, has added a new dimension to the computation technology, including the data storage and the service provisioning.
However, the proliferation of connected objects (IoT) takes a way of a fairly large growth over time and questions the traditional network architecture of the cloud computing. Indeed, the Cisco Internet Group Business Solutions predicts that by 2020 there will be nearly 50 billion objects connected, some applications of IoT must have a very short response time, others may contain private data, and others may generate a very large mass of data, something that will produce a big load on the network.
Thus, the large distance between the user and the cloud as well as the massive amount of data received by the user from IoT objects causes an increase in latency and bandwidth, which will have a negative influence on the response time of some crucial applications, as for instance those related to human health. Therefore, the need for new Edge computational paradigms to support the requirements of these latency-sensitive IoT applications is in the offing. Fog computing, Edge computing and Mobile Cloud computing, have been proposed to subdue the shortcomings of cloud computing by transferring some of the core functions of cloud towards the network edge. We notice that all these technologies may complement the Cloud ones by serving the requirements of the real-time, low-latency IoT applications running at the network edge [Sark16], and also supports complex analysis and long-term storage of data at the core of the network. Each of these technologies has functions that characterize it and offer a typical solution to the above problems.
The common denominator in these edge paradigms is the deployment of cloud computing-like capabilities at the edge of the network. In this work context, we are interested in the Edge computing paradigm; we study through this section its main functions while explaining its platform.

### 2.1 Edge Computing Platform

Edge Computing is an open distributed computing architecture; it's a convenient optimization method used in the cloud, pushing intelligence towards the periphery of the network allowing treatments to be close to the latter [Weis16].
This practice permits to express a decentralized data processing power, these are processed by the device itself without having to transmit them to a data center (cloud) which makes it possible to optimize the network infrastructure, reduce operating expenses and ensure a high level of flexibility at the level of IT services. Figure 1 illustrates this principle and shows the three main layers of an Edge computing architecture.

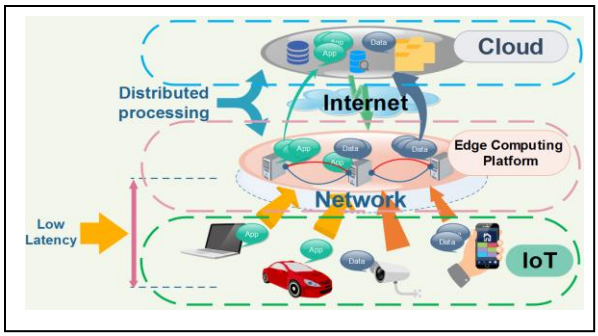


Figure 1: Edge Computing principle

The Edge Computing Platform Layer mainly provides mini data centers at the edge of the network, where the devices of the IoT are connected. This will make it possible faster processing (low latency), as shown in Figure 1, due to the short distance between the Edge Computing layer and the IoT devices (unlike the cloud, where data centers can be very far from the devices). We clarify some key words used by this paradigm and whose definition will serve later to draw our architecture.

- **Edge:** means the edge that differs from one use case to another, in an automotive scenario, the edge could be a car, in the making, it could be a machine in a workshop, and in the computer, the edge could be a laptop, etc.;
- **Edge Gateway:** represents the buffer between edge computing and the wider fog network. It is at the latter level that the altering and processing of the hundreds of thousands of data generated by the sensors is performed to increase efficiency by decreasing bandwidth and latency;
- **Edge Servers:** a set of computers placed in warehouses, distribution centers or factories that run the applications located near the network, they manage the incoming and outgoing internet mail flow;
- **Edge devices:** represent any device, or machine that produces, collects or processes the data. This can encircle the sensors, controllers, smart devices, such as smartphones, tablets, vehicles, computers as well as routers and servers;
- **Edge Application Services:** they reduce volumes of data to be transmitted, the resulting traffic and the distance to Browse. This reduces the transmission costs, the latency of shrink and improve the quality of service;
- **Edge Computing equipment:** represent a wide range of existing and new equipment used in Edge Computing. Many devices and peripherals edge are provided by CISCO and other hardware vendors;
- **Mobile Edge Computing:** offers IT service environment at the edge of the mobile network. This is characterized by ultra-low latency and high bandwidth as well as 5G scenarios.

### 2.2 Motivation

The Edge Computing is implemented to improve the performance of the Cloud Computing by bringing several interests which can be summed up in the following points [Hped17]:

- It allows to perform analyses, calculations on collected data from different IoT devices closer to the source of their generation (on the edge). This will lead us to efficient and latency less data processing;
- It ensures data security by moving security elements closest to the original source of the attack;
- It allows applications and smart devices to be used effectively in separate locations, and respond to data almost instantly;
- It also provides new capabilities in IoT applications. Namely, object detection, facial recognition, obstacle avoidance. These benefits reduce response time, energy consumption, etc. [Saty17].

## 3. FRACTAL4EDGE: A COMPONENT BASED MODEL FOR EDGE SYSTEMS

In the scope of our modelling approach, we opted for the use of component-based models to define the essential elements of the Edge Computing systems. These abstract models have a simple hierarchical nature which allows the mastery of their complexity.

### 3.1 Our Approach Principle

First, we give a generic layered architecture for Edge computing systems allowing a separation of concerns mastering thus their complexity. Then, this architecture constitutes an intermediate model for the formal one. This latter is based on Fractal components model, allowing specifying either structure or behaviour of these systems. Indeed, the separation of components of the Edge computing systems by adopting this architecture, called LAyered Edge architecture, ensures a simple and understandable crossing to the formal model that serves to analysis and execution purposes.

Traditional non-formal design methods generally fail to meet all the requirements of the Edge paradigm. The main problem is that they usually result in incoherent perspectives and ambiguous representation. The method we propose aims to give a unified way to describe systems so as to result in a deterministic, understandable model. In addition, it should be possible to 1/ (Specialization): divide into several sub-groups the development team of Edge computing system. Each one focuses on the development of a precise component; 2/ (Update easiness) update a component without requiring a complete recompilation of the hole model; 3/ (Choice of development languages) develop related software components with different programming languages; 4/ (Reusability) use existing software components to create new software; 5/ (Extensibility) extend a given Edge system by adding new components.

### 3.2 A Layered Architecture for Edge Computing Systems

In this section, we give the definition of the basic elements of our proposed LAyered Edge Computing architecture (see Figure 2). Obviously, this architecture is divided in three main layers according to the Edge computing principle (see Figure 1).

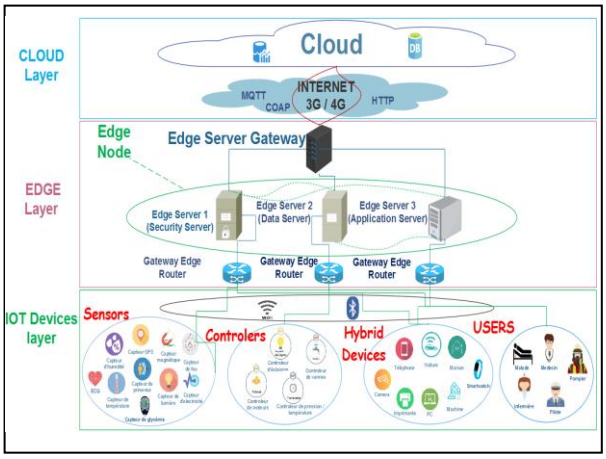


Figure 2: LAEdge Architecture

- **The Iot Devices Layer:** It brings together four layers (sensors, controllers, hybrid devices and users). The sensors are supposed to retrieve data, and the controllers are supposed to process and manipulate these data. As for hybrid devices, they are seen as consumers and data producers at the same time. However, users represent a source / interceptor of information. The connection between these sub layers and the upper layer (the Edge layer) is made through an under layer (layer connectors). The latter brings together a set of connection means (Wi-Fi, Bluetooth, etc.), which enables the retrieved data to be transferred to the Edge layers where the appropriate treatments are made.

- **The Edge Layer:** It represents the platform of the Edge Computing; it is the main layer of our architecture. It encompasses a set of peripherals, including edge servers which handle data manipulation and management such as security, analysis, processing of data, etc. in the closest way possible to IoT devices. This layer also encapsulates edge getaways such as: routers, switches, servers, etc., which in turn represent a connection point between the edge nodes and the set of IoT devices. This makes it possible to manage the network, in addition to validating, storing, processing and filtering hundreds of thousands of data on time before sending them to a data centre (cloud). When the appropriate treatments are completed, the resulting data are communicated to the upper layer (cloud layer) through an edge

gateway server and by means of a sub-layer "the communication protocol-layer", (http for example).

- **The Cloud Layer:** It represents a space of storage, analysis and data processing, coming from the lower layer (the Edge layer), by means of communication protocols. Hence, these data can be accessible at any time and from anywhere.

### 3.3 Software Elements of FRACTAL4EDGE Model

We propose a comprehensive and a generic formal model for Edge Computing systems, reducing their complexity and design, using Fractal components model. Our Fractal4edge model gives a precise semantics to architectural elements of these systems and possible interactions that govern their complex behavior. Especially, all the advantages of Fractal components model are inherited by applying our process of developing such systems. Thus, we have defined some matching rules that allow associating a clear semantics to the various Edge Computing concepts identified in the LAEdge architecture. The following table summarizes a set of these rules.

Table 1: Fractal for LAedge

| **Architectural element in LAEdge** | **Fractal Component** | **Fractal interfaces** | **Fractal controller** |
|---|---|---|---|
| Iot Layer | Composite component “C-Iot” | Required interfaces “edge” and “cloud” Provided interface “iot” | Link controller “BC” and “CC” |
| Edge Layer | Composite component “C-Edge” | Required interface “iot” Provided interface “edge” | Link controller “BC” and “CC” and “LC” |
| Cloud Layer | Primitive component “C-Cloud” | Required interface “edge” Provided interface “cloud” | Link controller “BC” and “CC” and “LC” |
| Sensors | Primitive component “C-CPT” | Provided interface “iot” | |
| Controllers | Primitive component “C-CTRL” | Required interfaces “edge” or “cloud” Provided interface “iot” | Link controller “BC” |
| Users | Primitive component “C-User” | Required interfaces “edge” or “cloud” Provided interface “iot” | Link controller “BC” |
| Edge Gateway | Primitive component “C-Edge-GT” | Required interface “iot” Provided interfaces “gt1”, “gt2” “gt3”, “gt4” | Link controller “BC” |
| Edge Node | Composite component “C-Edge_Node” | Required interfaces “gt1”, “gt2” “gt3”, “gt4” Provided interfaces “ne1” or “ne2” | Link controller “BC” and “CC” |
| Server Edge Gateway | Primitive component “C-SV-Edge-GT” | Required interfaces “ne1” or “ne2” Provided interface “edge” | Link controller “BC” |

The above mapping table is important because aims to represent the proposed architecture model based on the three FRACTAL main concepts: ‘’Fractal Component’’, ‘’Fractal interfaces’’ and ‘’Fractal controller’’. At this level of abstraction, the semantics of ‘’IoT Layer’’ ‘’Edge Layer’’ and ‘’Edge Node’’, is realized by composite components, while ‘’Cloud Layer’’, ‘’Sensors’’, ‘’Controllers’’, ‘’Users’’, ‘’Edge Gateway’’ and ‘’Server Edge Gateway’’, are just primitive components. We have opted for the FRACTAL component model, it is an extensible and reflexive model which has a simple hierarchical nature, and which is independent from any programming language. Additionally, FRACTAL offers a clear

separation between the functional and non-functional needs of a given application, with a great modularity and possibility of extended extensions, as well as a persistence management of data through its API, which perfectly suits our case study.

## 4. CASE STUDY: PMS-EDGE

In the last few years, healthcare applications based on Internet of Things (IoT) have been receiving increasing attention because they provide a seamless platform to millions of people to facilitate ubiquitous health monitoring of patients [Jong18]. These applications are challenging the existing healthcare systems (PMS) and make them increasingly difficult to manage. The PMS (Patient Monitoring System) is a system for monitoring the health of patients. It combines hardware and software devices to protect and monitor patient health in real time using a set of sensors, controllers and medical intervention devices. These allow the collection of primary vital signs which are standard in most medical settings [Shen18]: Body temperature, Heart rate or Pulse, Respiratory rate and Blood pressure.

The medical and healthcare industry in the context of functions, operations, and applications can be divided into three areas [Fara18]: (i) large healthcare organizations such as hospitals, (ii) small clinics and dispensaries, and (iii) non-clinical environments such patients' homes. The following case study "PMS-EDGE" provides an effective solution on how data can be collected through sensors and provided in terms of the patient's vital signs, in small clinics. We will take the potentials areas where IoT and Edge computing could play an instrumental role, and we will discuss applicability of our proposal "FRACTAL4EDGE" by considering this case study. The aim of the "PMS-EDGE" case study is to improve response time by delegating the data-processing authority to a local edge node.

### 4.1 System architecture

The proposed "PMS-EDGE" consists of a smart clinic solution that is set up with medical sensors, a patient subject with mobile devices such as smartphone. The sensor data (patient's vital signs) are transmitted using a smart gateway to an Edge node (located near the network) where pre-processing algorithms are implemented. The processed data with contextual information is finally transmitted to medical intervention devices (controllers) or outright to doctors and nurses for performing necessary treatments. The floor plan and overview of the "PMS-EDGE" solution is shown in Figure 3.

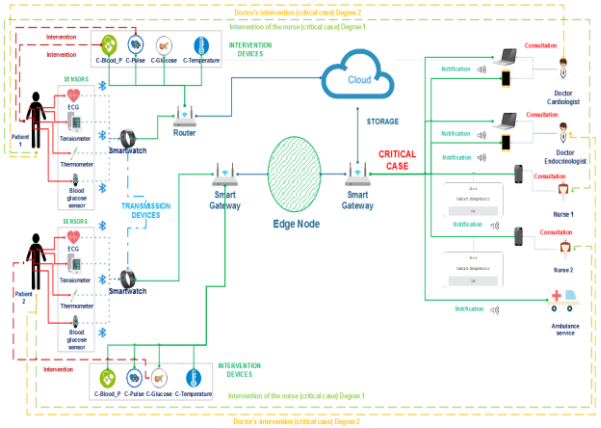


Figure 3: PMS-EDGE solution

### 4.2 PMS-EDGE Fractal model

The proposed methodology is described in Figure 4. It shows the general structure of our case study and illustrates the different components (fractal) and their connection through communications interfaces (provided in red, and required in green). It also shows the set of controllers that allow the management and execution of components, and ensure their assembly.

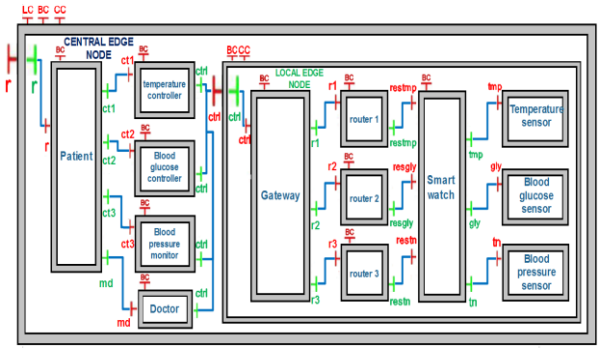


Figure 4: PMS-EDGE architecture

The architecture model "FRACTAL4EDGE" for our case study "PMS-EDGE" consists of fifteen components, thirteen of them are primitive while only two are composite. Table 2 summarizes the main

elements of the “PMS-EDGE” and their corresponding ones in the proposed methodology “FRACTAL4EDGE”.
Thanks to the model proposed for edge computing systems 'Fractal4Edge', we could instantiate it and apply it to represent the model of our case study (PMS-EDGE) in the form of fractal components. This solution has made us easier modelling task of our PMS system through a set of fractal components drawn from the Fractal4Edge model and which results in the Table 2.

Table 2: FRACTAL4EDGE PMS

| **Components** | **Fractal Interfaces** | **Fractal Controllers** |
|---|---|---|
| Composite component “central edge node” | provided interface “r” implementing java.lang.Runnable | -link controller “BC” -content controller "CC" -life cycle controller "LC " |
| Primitives components “primary vital signs controllers” | required interface” ctrl” provided interface “ctx” | link controller "BC" |
| Primitive component “doctor” | required interfaces “ctrl” and “md” | link controller "BC" |
| Primitive component “Patient” | provided interfaces “ctx” and “md” required interface” r” | link controller "BC" |
| Composite component “local edge node” | provided interfaces “ctrl” required interface” r” | -link controller "BC" -content controller "CC" |
| Primitives components “primary vital signs sensors” | required interface” ctrl” provided interface " tmp, gly and tn " | / |
| Primitive component “smartwatch” | required interface ” tmp, gly and t ” provided interface " restmp, resgly and restn " | / |
| Primitives components “Routers” | provided interface” rx ” required interface " restmp, resgly and restn " | -link controller "BC" |
| Primitives components “Smart-gateway” | provided interface” ctrl” required interface "rx" | -link controller "BC" |

### 4.3 Execution and Evaluation

In order to carry out an evaluation of the proposed methodology, an implementation consists of a judicious coupling between FRACTAL API as a high-level component-based language, and FRACTAL ADL for parsing XML-based specifications of the proposed PMS-EDGE solution. Figure 5 gives a broad outline of the defined components and their dependencies according to the proposed “LAEdge” architecture in section 3.1.

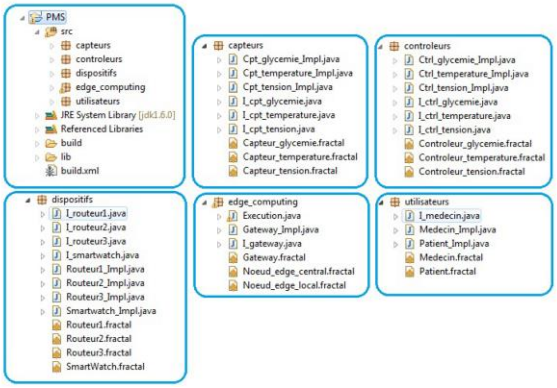


Figure 5: PMS-EDGE Implementation

To illustrate the execution of our case study “PMS-EDGE”, we suggest two scenarios of collecting data through sensors and providing it in terms of the patient’s vital signs; the first is a cloud computing-based solution, and the second consists of our proposed solution. For this purpose, we evaluate the patient's

condition through three vital signs: Body Temperature (BT), Blood Glucose (BG) and Blood Pressure (BP). For a given value of BT= 40; BG=1.7 and BP= 19.90, the simulation results using the two scenarios (cloud-based solution and our proposed solution) are shown in the figure 6 and figure 7 respectively. The Figures (6 and 7) help to explain the simulation result by displaying the response time for medical relevant treatments.

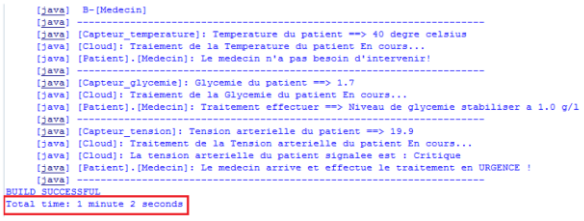


Figure 6: Simulation result for Cloud-based solution

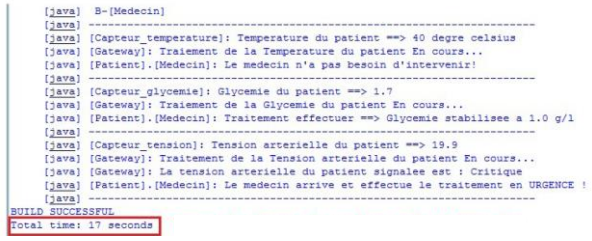


Figure 7: Simulation result for PMS-EDGE solution

Lastly, again according to the simulation result of figures 6 and 7, we conclude that the data is processed on a cloud-based solution whose response time is more than “1 minute”, while on the proposed solution (PMS-EDGE), the response time is improved; less than “18 seconds”. This early response time will save countless lives.

## 5. CONCLUSION

Component-Oriented Software Development (COSD) have been adopted as a methodology foundation in the Edge computing paradigm for their main characteristics of modularity and cohesivity. In that way, an approach based on software component technology “FRACTAL4EDGE” is proposed and associated to the proposed software architecture “LAEdge”; supporting the modelling of Edge computing dedicated systems. This allows the designer to separately specify the semantics of each layer using a FRACTAL component model, which can specify every-changing dynamic and distributed integration of software architecture elements. Furthermore, a case study of the patient monitoring system “PMS-EDGE” along with its application model and Edge-based network topology is expounded. It aims is to improve response time and gives the opportunity to save lives.

As future work, we plan to exploit meta-modelling and model transformation tools to conceive a graphical interface and a “FRACTAL4EDGE” specification generator in order to facilitate the exploitation of our proposed approach, thereby permitting graphical editing and automatic generation of the corresponding specifications.

## References

[Sark16] Subhadeep Sarkar, Sudip Misra, Theoretical modelling of fog computing: a green computing paradigm to support IoT applications, IET Netw, 2016, Vol. 5, Iss. 2, pp. 23–29 & The Institution of Engineering and Technology 2016, ISSN 2047-4954.

[Rod18] Rodrigo Roman, Javier Lopez, Masahiro Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges", Future Generation Computer Systems, Volume 78, Part 2, January 2018, Pages 680-698, Elsevier.

[Rajk17] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghoshy, and Rajkumar Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in internet of Things, Edge and Fog Computing Environments",
Journal of Software: Practice and Experience, Volume47, Issue 9, Special Issue: Cloud and Fog Computing, September 2017, Pages 1275-1296.

[Hazl11] Hazleen Aris and Siti Salim, "State of Component Models Usage: Justifying the Need for a Component Model Selection Framework", The International Arab Journal of Information Technology, Vol. 8, No. 3, July 2011.

[Coup06] Eric Bruneton, Thierry Coupaye, Matthieu Leclercq, Vivien Quéma R. Nicole, Jean Bernard Stefani, "The FRACTAL component model and its support in Java", First published: 17 August 2006, https://doi.org/10.1002/spe.767.

[Weis16] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing : Vision and challenges. IEEE Internet of Things Journal, 2016.

[Hped17] Hewlett Packard Entreprise Development LP. What is edge computing. Technical report, HPED, 2017.

[Saty17] Mahadev Satyanarayanan, The emergence of edge computing. Journal of Computer, 2017.

[Shen18] Minh Pham, Yehenew Mengistu, Ha Do, Weihua Sheng. Delivering home healthcare through a Cloud-based Smart Home Environment (CoSHE). Future Generation Computer Systems, 2018. Elsevier.

[Jong18] Min Woo Woo, JongWhi Lee, KeeHyun Park. A reliable IoT system for Personal Healthcare Devices, Future Generation Computer Systems, 2018. Elsevier.

[Fara18] Bahar Farahani, Farshad Firouzi, Victor Chang, Mustafa Badaroglu, Nicholas Constant, Kunal Mankodiya, Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare, Future Generation Computer Systems, 2018. Elsevier.