

# An Edge-based Architecture to Support Efficient Applications for Healthcare Industry 4.0

Pasquale Pace, *Member, IEEE*, Gianluca Aloï, *Member, IEEE*, Raffaele Gravina, *Member, IEEE*, Giuseppe Caliciuri, Giancarlo Fortino, *Senior Member, IEEE*, and Antonio Liotta, *Senior Member, IEEE*

**Abstract**—Edge computing paradigm has attracted many interests in the last few years as a valid alternative to the standard Cloud-based approaches to reduce the interaction timing and the huge amount of data coming from IoT devices toward the Internet. In the next future, Edge-based approaches will be essential to support time-dependent applications in the Industry 4.0 context; thus, the paper proposes *BodyEdge*, a novel architecture well suited for human-centric applications, in the context of the emerging healthcare industry. It consists of a tiny mobile client module and a performing Edge gateway supporting multi-radio and multi-technology communication to collect and locally process data coming from different scenarios; moreover, it also exploits the facilities made available from both private and public Cloud platforms to guarantee a high flexibility, robustness and adaptive service level. The advantages of the designed software platform have been evaluated in terms of reduced transmitted data and processing time through a real implementation on different hardware platforms. The conducted study also highlighted the network conditions (data load and processing delay) in which *BodyEdge* is a valid and inexpensive solution for healthcare application scenarios.

**Index Terms**—Body Sensor Networks, Edge Computing, Cloud Computing, Internet of Things, Heart Rate Variability.

## I. INTRODUCTION

Recent Research and Markets report predicted that the IoT healthcare market is expected to grow from USD 41.22 Billion in 2017 to USD 158.07 Billion by 2022, at a Compound Annual Growth Rate (CAGR) of 30.8% [1] demonstrating how medical care and healthcare services represent one of the most attractive fields for the development of IoT. However, although novel healthcare services are expected to reduce costs and increase the quality of life of users, the more data organizations collect, from both fixed and mobile medical monitoring devices, the more difficult is the effective use of a Cloud-based centralized data processing and repository.

In fact, in the case of almost real-time applications the massive use of Cloud-based infrastructure could affect the real-time constraints and burden the network infrastructure from the local to the Cloud. Moreover, to provide healthcare services to

a large scale of patients, simplistic approaches, in which the infrastructure between sensing devices and the Cloud is used only as a common communication infrastructure, are not often feasible due to the presence of other healthcare challenges. For example, in some cases, to ensure patient's privacy, data cannot be stored in the public Cloud or, in other cases, for patient safety, data must be immediately available and any delay or failure introduced by the Cloud cannot be tolerated.

Speed of data and analysis is essential in many industrial IoT applications but is also a key element of industrial healthcare and all the other areas where we move towards autonomous and semi-autonomous decisions made by systems, actuators and various controls. In particular, the degree of autonomy is one of the more important and desired goals of the next Industry 4.0 development stage. Thus, for a mix of reasons (*bandwidth, costs, speed, predictive analytic, remoteness, maintenance*) a faster, cheaper and smarter approach than the traditional Cloud-based one, needs to be implemented [2].

Very recently, smartphones are playing an increasingly important role in modern computing architecture thanks to their mass distribution and the increasing computing resources. They are often used as mobile IoT gateways [3] or to support heterogeneous wireless sensor networks integration [4], but in our opinion, each personal mobile device (*i.e.* smartphone, smartwatch) can play a central role also in Edge computing architecture. The inclusion of personal mobile devices in modern Edge computing IoT systems can speed up the development of people-centric mobile solution, not only in the healthcare area, with an evident fallout on the improvement of the quality of life. On this direction, the use of personal mobile devices is the quickest and easiest way to support the integration of Body Sensor Networks (BSN) and to provide large scale of m-Health services. According to this modern vision, healthcare Edge computing is emerging as a way for entities to embrace near real-time results by processing data as close as possible to the source, definitely stating that Cloud computing is not an efficient way to process data when data are produced at the edge of the network [5].

Within this challenging context we propose *BodyEdge*, a novel Edge-based architecture well designed to support emerging applications for healthcare industry. The main contribution of this work is the design of a new communication and computing architecture for mobile healthcare monitoring based on the Edge paradigm [5] rather than on the classical Cloud paradigm to achieve:

- Reduced communication delay.
- Wide support to scalability and responsiveness.

Manuscript received January, 31, 2018; revised April, 13, 2018; accepted May, 27, 2018. This work is supported by the INTER-IoT, Horizon 2020 European Project, Grant Agreement #687283, financed by the European Union. (*Corresponding author: P. Pace*)

P. Pace, G. Aloï, R. Gravina, G. Caliciuri and G. Fortino are with the Department of Informatics, Modeling, Electronics and Systems, University of Calabria, Rende 87036, Italy (e-mail: ppace@dimes.unical.it; aloï@dimes.unical.it; rgravina@dimes.unical.it; gcaliciuri@dimes.unical.it; g.fortino@unical.it).

Antonio Liotta is with the Department of Electronics, Computing and Mathematics, University of Derby, UK (e-mail: a.liotta@derby.ac.uk).

- BSN and mobility support.
- Limited cost in terms of bandwidth for data transmission (*i.e.* it is not needed to send all data to the Cloud but only statistics).
- Enhanced privacy level (*i.e.* the Edge network can be considered as a private Cloud).

The designed and implemented Edge-based architecture mainly consists of two complementary components: *i)* a tiny mobile client module to be installed on mobile devices (*i.e.* smartphones, smartwatch) to integrate BSN and *ii)* a performing gateway, placed at the edge of the network, deployable on different resource constrained or resource-rich hardware platforms supporting multi-radio and multi-technology communication to collect and locally process data coming from several application scenarios. Two different cardiac monitoring case studies have been implemented to detect high-stress conditions of automotive factory workers and athletes connected to both BodyEdge and a reference Cloud platform in order to measure the overall system performance in terms of processing time, delay and scalability.

The obtained results highlighted the system conditions and the application scenarios in which the proposed Edge-based approach is very useful and convenient while standard Cloud support should still be adopted for long-term storage and statistical analysis.

The remainder of this paper is organized as follows. Section II discusses the recent related works on IoT, BSNs and Edge computing for healthcare systems and applications, also pointing out the advantages of using Edge devices respect to the classical Cloud platforms. Section III describes the proposed BodyEdge system architecture, including specific features and application domains, also presenting the general system architecture to support healthcare composed by both body client and body edge sides. Section IV describes the case studies designed to test the new system architecture while the performance evaluation on a real testbed are presented in section V. Finally, Section VII concludes the paper pointing out the main future research directions and open issues.

## II. RELATED WORKS

The IoT paradigm is a viable solution to provide healthcare services to a large scale of patients; however, simplistic approaches in which the infrastructure between sensing devices and the Cloud is used only as a common communication infrastructure, are not often feasible due to the presence of other healthcare challenges. For example, in some cases, to respect the patient's privacy, data cannot be stored in the public Cloud or, in other cases, for patient safety, data must be immediately available and any delay or failure introduced by the Cloud cannot be tolerated.

In the last few years, several solutions have been proposed to support healthcare services using standard Cloud-based solutions. Hassanaliieragh *et al.* [6] proposed a mobile Android application for electrocardiogram (ECG) monitoring. This solution does not support Edge processing so health professional can take a decision only after Cloud processing and not at the time the health device collects the signal. In the same

direction, the authors of [7] proposed an application based on the ECG as a service that allows collecting, processing, storing and analyzing ECG data streams generated by sensors worn by several individuals. Also, this solution needs Cloud processing and miss reactivity. More and more other solutions exist but all highlight the needs to overcome specific healthcare challenges.

With the advent of Edge Computing, the healthcare industry has evolved considerably due to its ability to store, process and analyze data closer to patients, hospitals, and clinics. In fact, the Edge Computing is permeating the industry so powerfully that doctors and physicians are starting to increasingly rely on it to support patient's treatment. In this context, many new research works have been conducted to demonstrate that standard Cloud-based services still face various issues related to unpredictable delays, high bandwidth requirements and security/safety concerns. These issues are very critical to healthcare and Active and Assisted Living (AAL) scenarios where a correct and timely reaction can also result in saving a life or drastically reducing a disability (*e.g.*, after a stroke).

In [8], authors define the Edge Computing as the enabling technologies allowing computation to be performed at the edge of the network. Edge computing is often exchangeable with Fog computing [9], but Edge computing focus more closely to the Things side, while Fog computing comes closer to the Internet infrastructure side. Under the Edge computing paradigm, some application services are handled at the edge and some others are handled by a remote server in the Cloud. This type of distributed analytics Edge intelligence offers great potential to overcome healthcare challenges, improving the effectiveness and the efficiency of pervasive health monitoring.

According to this modern vision, a flexible multi-level architecture based on a computing paradigm in which heterogeneous devices at the edge of the network collect data, compute a task with minimal latency, and produce physical actions meaningful for the user, has been proposed and tested in [10]. In the same direction, the authors of [11] proposed another IoT enabled healthcare system architecture which benefits from the concept of fog computing providing advanced techniques and services such as embedded data mining, distributed storage, and notification service at the edge of a network. Authors of [12] employed pervasive fall detection as a specific case study since fall is a major source of injury and mortality among stroke patients. They proposed a distributed fall detection system, named U-Fall, utilizing both Edge devices (*e.g.*, smartphones) and data center services (*e.g.*, server in the Cloud) to achieve low miss rate and low false positive rate when compared to non-Edge state-of-the-art systems.

Many existing contributions use a gateway as an Edge intermediary between IoT devices and the Cloud. However, such solutions fail to satisfy some critical challenges of the healthcare area, particularly for nomadic health monitoring, availability, security, and privacy. The aim of our BodyEdge solution is to overcome these domain-specific requirements by providing a solution that moves the Edge closer to end users.

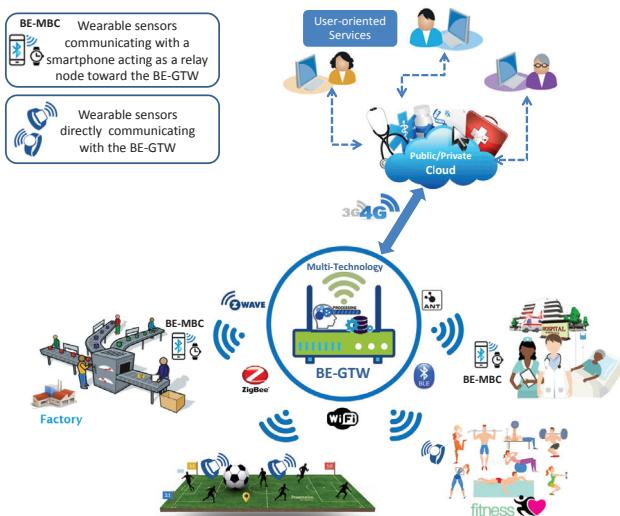


Fig. 1. High-level architecture of BodyEdge.

### III. SYSTEM ARCHITECTURE TO SUPPORT MOBILE HEALTHCARE

In this section, we describe BodyEdge, a general IoT system architecture well designed to support specific applications for emerging healthcare industry. The need to develop such novel architecture comes from the accurate analysis of the nowadays healthcare contexts in which application requirements related to communication delay, scalability, responsiveness, transmission capacity and data privacy are becoming more and more important; thus the wise integration and use of an Edge communication device can play a key role to face few limitations of the public/private Cloud platforms as pointed out in the previous section.

BodyEdge presents key features that make it much more than a simple improvement with respect to a reference architecture [13], [14]; in particular, it consists of two main components: a tiny BodyEdge Mobile BodyClient (BE-MBC) software module and a performing BodyEdge gateway (BE-GTW).

Figure 1 shows the use of the proposed BodyEdge architecture in different healthcare scenarios in which the users can be either workers in a factory, people playing sports or patients in a hospital. The multi-radio and multi-technology Edge gateway can collect and locally process data coming from different scenarios or it can exploit the facilities made available from both private and public Cloud platforms according to the specific requirements of each scenario in order to guarantee a high flexibility, robustness and adaptability level of service.

In particular, as shown in figure 2, the proposed framework is organized in a three-tier (*i.e.* cloud/edge/IoT devices) architecture in which the Edge Layer represents the connecting layer between the far cloud and the physical IoT devices whose data can be directly collected from the BE-GTW or through the BE-MBC in specific application contexts.

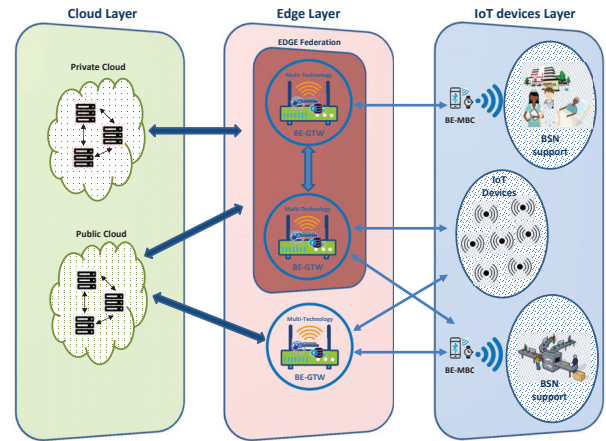


Fig. 2. A three-tier (cloud/edge/IoT devices) architecture.

#### A. BE-MBC architecture

The BE-MBC component, developed as a software application, can be installed on a smartphone and it communicates with the body sensors worn by the people using multi-radio interfaces as described in [3], [4]. It basically acts as a multi-radio communication relay node in order to reach the Edge gateway when the body sensors at the Edge gateway are too far (due to the use of a small-range communication technology) or they cannot be directly connected to it (due to a not easy peering process to be physically executed by the user on the Edge gateway). In these cases, such tiny mobile client is a mandatory component of the proposed communication architecture since it also acts as a simplified edge gateway with reduced capabilities toward the more powerful BE-GTW. However, the communication with the BE-GTW can also take place without any mobile client component when IoT devices can directly send their data to the BE-GTW as shown in figure 2.

Figure 3 shows the functional scheme of the BE-MBC component constituted by the following modules:

- *Manager* - it starts all software blocks by handling the logic functionalities of the application at a high level.
- *Communication Engine* - it contains all the possible bound services (*i.e.*, ZigBee, ZWAVE, Wi-Fi, Bluetooth etc) by handling the logical functionality of access to data and measures. The bound services are based on a standard client-server communication model by allowing other components and applications to be connected, to send requests and receive replies.
- *IoT Device Handler* - it is in charge of data interpretation exchange, control commands execution, and dynamic adapters loading of discovered devices. It also handles the communication between remote interface management and GUI by favouring the message passing from the IoT devices to the graphical interface.
- *IoT Device Controller* - it is in charge of the dynamic device adapter loading by setting the specific data structure for each new device, adding it to the protocol device module.
- *BE-MBC database* - it is used to store the acquired measurements on a local database on the smartphone in



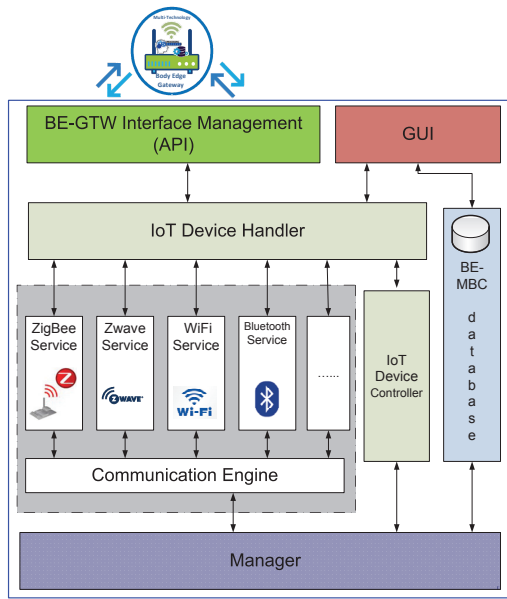


Fig. 3. BE-MBC software architecture.

order to visualize and manage the acquired data.

- *BE-GTW Interface Management* - it allows the communication with the BE-GTW for sending data and receiving control commands for the connected IoT devices.
- *Graphical User Interface* - it handles an easy interaction with users. Through these modules, it is possible to show all measurements and access all settings of the software modules.

### B. BE-GTW architecture

In this subsection, we describe the general architecture of the BE-GTW to support applications for healthcare industry that is mainly composed by i) a *Remote Interface Management (RIM)* through which the BE-GTW can directly communicate with external end users or Public/Private Clouds by using specific APIs; ii) a *BodyEdge Federator (BEF)* through which the BE-GTW can communicate with other Edge gateways placed in different locations with the aim of implementing a gateways federation for distributed services; a iii) *BodyEdge Middleware (BEMW)* to enable the various components of the BE-GTW architecture to manage data also guaranteeing interoperability; iv) a *COmmuNication, COordination, PRocessing and management (CONCePT)* module able to acquire data from different communication interfaces by using different communication protocols also guaranteeing a specific quality of service (QoS) level, to process data locally or to send acquired data to the Cloud for a remote processing and, to control several devices. In particular, the CONCePT module consists of the following main macro blocks that can interact with each other as shown in Figure 4:

- *Access Network Controller (ANC)*: it handles the reception and transmission of messages over the air, and manages the radio duty cycling. It is formed by a series of decoders for incoming packets and a series of encoders for outgoing packets. Each message received or sent is initially handled by the radio controller that provides a

common interface on a specific radio adapter to support several communication technologies such as Bluetooth Low Energy (BLE), ANT+, Z-wave, ZigBee, Wi-Fi, LTE/5G etc.

- *The QoS Manager*: it provides simple strategies to support the quality of service by prioritizing data flows depending on their relevance and timeliness. At the access layer, data traffic categories are differentiated according to their requirements. Data streams are then categorized into two main classes: delay tolerant and throughput elastic traffic (i.e. temperature sensing IoT devices) and delay sensitive and bandwidth *inelastic* (real-time) traffic (i.e. ECG sensing IoT devices). Inelastic traffic is further discriminated in two subclasses: a high-rate inelastic traffic (e.g., real-time video traffic) and a low-rate inelastic traffic (e.g., real-time ECG monitoring). The QoS module assigns a different priority to traffic streams coming from IoT devices according to their traffic classes and supports the selection of the best available link (e.g., Wi-Fi, LTE/5G, FTTx) towards the public/private Clouds or other federated BodyEdge gateways. The best link is selected by monitoring connections performances in terms of an available level of bandwidth, delay, and packet loss.
- *IoT Device Management (IoTDM)*: it is accessible from every other component that needs information about any sensor/actuator/IoT device. It stores in memory (and persisted in a local database) a map containing an entry for each sensor/actuator/IoT device. This entry is identified by a unique ID automatically generated and contains information about the specific IoT device, the protocol used for communication and the physical address of the device. It includes the following internal blocks:
  - *Discovery*: it is in charge of periodically executing a discovery procedure to find new IoT devices asking for a connection to the BE-GTW.
  - *Registry*: it is responsible for registering all the IoT devices with their multiple sensors and actuators in the BE-GTW. It will add an entry in the Data Manager database with the information about each sensor and actuator.
- *Application Protocol Controller (APC)*: it contains all the communication protocols supported by the BE-GTW also implementing the common interfaces between those protocols and the other components such as the BE-GTW Configurator, the BEMW and the BEM. This component provides support to the messages exchanged during the registration phase and the triggering action of each IoT device. It also plays a central role during the communication between the remote public/private Cloud platforms and the devices (through the BEMW) when they act as actuators and during the internal communications with its specific protocol modules (CoAP, MQTT, LwM2M, ...) [16].
- *BodyEdge Manager (BEM)*: It is used by the BE-GTW when a local data processing can be performed without using external Cloud resources or facilities. In these

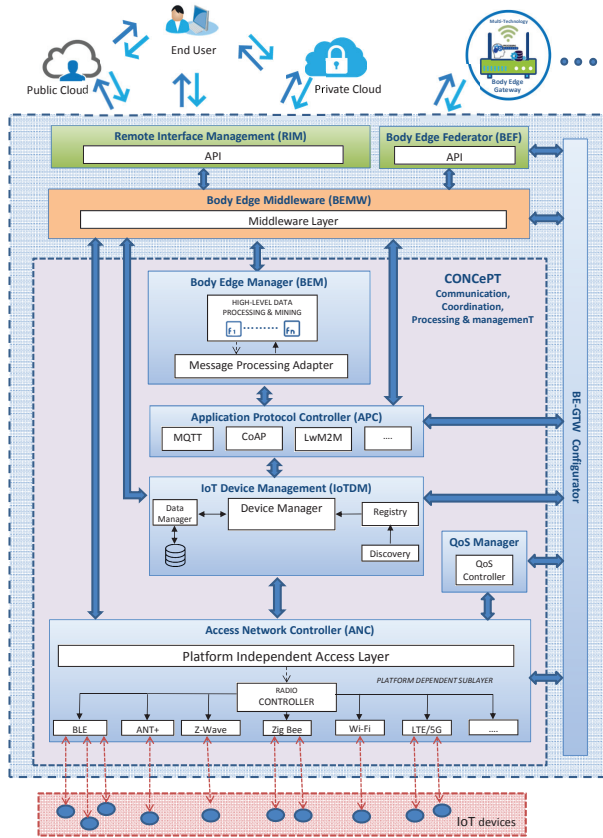


Fig. 4. Architecture of BE-GTW.

cases, a set of high-level data processing and mining functionalities such as those related to data fusion and computer vision, are made available for real-time local processing and computation.

All the described modules are finally connected to the *BE-GTW Configurator* that contains the general gateway configuration parameters uploaded through a text file to be stored in a local memory. Every component can access the BE-GTW configurator component to get the right configuration values.

#### IV. CASE STUDIES DESCRIPTION

In this section, we describe the Cardiac Monitoring case study to test the proposed system architecture to detect high-stress conditions for users in two different scenarios:

- Workers in a factory;
- Athletes training in a fitness center.

In these scenarios, we have to remark that the participants involved in each specific case study need to be equipped with a smart-watch in order to receive data from the chest band on the Bluetooth interface and to forward data toward the Edge gateway on the Wi-Fi interface. In this way, we can correctly support a proper radio coverage needed to implement the described case studies. In these contexts, the BE-GTW architecture will be used to monitor the mental stress by acquiring and analysing Heart Rate Variability (HRV) signals [17],[18], as discussed in detail in the following section. HRV analysis is based on the study of the beat-to-beat variations in the heart rate. Doctors and psychologists agree on the

importance of HRV for mental stress recognition, among other mental conditions such as anxiety, phobias and post-traumatic stress disorders.

##### A. HRV features for stress detection

The input signal for any type of HRV analysis is derived from the inter-beat, or *RRi* time series which are obtained as the time difference between successive R-wave occurrences. That is, the  $i^{th}$  RR interval is obtained as the difference between the heart beat at time  $i$  and  $i - 1$ :

$$RRi = beat(t_i) - beat(t_{i-1}) \quad (1)$$

Several features measure the variability within the *RR* series; they are mainly divided in time-domain and frequency-domain features. Fundamental time-domain HRV features for stress detection [19] include the standard deviation of *RRi* (SDNN), the root mean square of successive differences (RMSSD), the NN50 (*i.e.* the number of successive intervals differing by more than 50 ms), and pNN50 (the proportion derived by dividing NN50 by the total number of *RRi* in the observation period).

Frequency-domain features for stress detection [20] involve the spectral analysis of HRV, and specifically the power spectral density (PSD) of the *RR* series by considering the high frequency (from 0.18 to 0.4 Hz) component, the low frequency component, the normalized Power spectra in Low (from 0.04 to 0.15 Hz) (LF) and High (from 0.15 to 0.4 Hz) (HF) frequency ranges and the ratio of the low-to-high frequency spectra.

##### B. Testbed implementation

The testbed we implemented is specifically designed to be applied to all the considered communication scenarios to validate and measure the overall performance of the proposed system architecture.

In particular, the BE-MBC module has been installed on a Huawei watch 2 supporting the Android Wear 2.0 operating system and tested with a Polar H10 chest band to acquire the cardiac signal in real time; then the signal is sent to the BE-GTW installed on two different hardware platforms: *i)* Raspberry Pi3 single board and *ii)* Zotac CI540 NANO Pc.

In addition, we also integrated the HRVFrame [15] within the BEM module of the BE-GTW in order to support processing and analysis of the received heart signals for stress detection. In particular, this software module consists of an extensive Java-based framework containing many features covered in the HRV analysis literature and, among the supported features, we considered those one described in Section IV-A. Moreover, the stress level of the subjects is measured on a time window of ten minutes representing the minimum observation period to obtain significant results [21].

Finally, we also used a simple wireless access point to route data coming from the users to a standard Cloud platform such as Microsoft Azure on which we installed a Virtual Machine for remote processing. This further implementation is aimed to validate the proposed BE-GTW architecture with respect to the standard Cloud based approach. In this way, we can make a

TABLE I  
EDGE AND CLOUD COMPUTING PLATFORM SPECIFICATIONS

	Raspberry Pi3	Zotac NANO PC	Azure Cloud
CPU	Quad Core 1.2GHz Broadcom BCM2837	dual core 1.5GHz Intel i5-4210Y	dual core 2.4GHz Intel(R) E5-2673 v3
Memory	1GB RAM	4GB RAM	8GB RAM
Price (\$)	40	500	95 per month

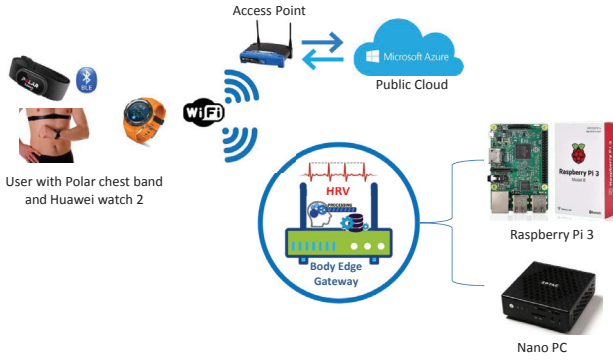


Fig. 5. Testbed scenario.

fair comparison in terms of processing time, propagation delay and used bandwidth to outline the best choice with respect to the application scenarios.

Table I resumes all the hardware characteristics of both Edge and Cloud platforms used during the testbed while the specific communication scenarios are shown in Figure 5. It is worth noting that also the prices shown in Table I are important parameters to guide the choice of the most suitable solution; in fact, since the raspberry Pi 3 platform is very cheap, it could be a valid choice in small environments with a small number of users. The performance analysis described in the next section will try to put evidence on these particular and meaningful aspects.

## V. PERFORMANCE EVALUATION

In this section, we measure the performance of our proposal in terms of processing time and needed bandwidth for data transmission by increasing the number of connected users. The aim of this analysis is mainly devoted to highlight the system conditions in which the role of BE-GTW can be played by a constrained hardware platform instead of using more powerful and costly Cloud computing resources that in some other conditions cannot be used because they do not meet the real time constraints imposed by specific applications.

Since the presented results aimed at evaluating the computational load of the presented BodyEdge architecture rather than the accuracy of the stress detection algorithm, we used real heart rate traces coming from previous studies on workers in automotive manufacturing [22] and athletes [23], [24] to progressively increase the computing load for the BE-GTW installed on three different platforms (*i.e.* Raspberry Pi3, Nano PC and Azure Cloud). This choice is also motivated by the lack of a significant number of subjects available for such field trials. The two reference scenarios are mainly motivated by the will to test the system performance on different heart rate baselines. In fact, since the factory workers have on average

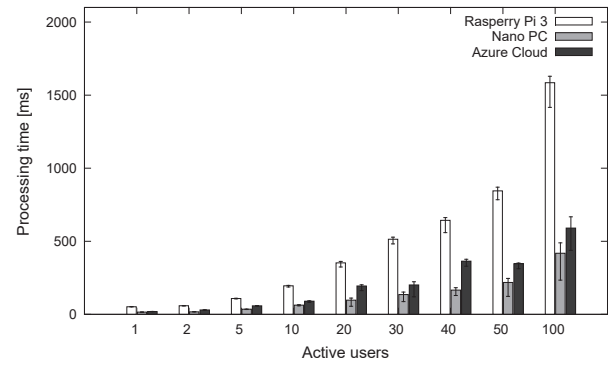


Fig. 6. Processing time increasing the number of workers.

75 beats per minute (bpm) whereas the athletes have 170 bpm, both the data traffic and processing load are different.

Following the presented approach, we have been able to measure the system performance without involving a big number of users directly. It is worth noting that the results, obtained by using already stored traces, are still meaningful from the computation load, bandwidth requirement and delay time perspectives.

Figure 6 shows the processing time by increasing the number of workers from 1 to 100 in order to study the system behaviour in different load conditions. In particular we monitored the heart activity of the users for 5 hours by computing the time and frequency domain features described in Section IV-A over 10 minutes windows. It is worth noting that many healthcare applications require processing consecutive data windows partially overlapped to improve recognition precision and to update the recognition output more frequently. Thus, a key parameter to be considered is the shift ahead which is often around 50% of the window length, but it could be much smaller [25].

The averaged obtained results show that the Nano PC and the azure Cloud virtual machine present very similar results in every load conditions while the constrained Raspberry Pi3 is always less performing but it can support up to 100 workers in less than 1600ms.

The same system parameter has been computed in the scenario involving the athletes doing trainer and the Figure 7 shows coherent results; in fact, the processing time experienced by all platforms in this scenario is greater than the previous one since the athletes generate more data to be processed due to the higher cardiac activity (*i.e.* 170 bpm on average). Anyway, the general trend of the obtained results is confirmed and we can argue that the constrained Raspberry Pi3 Edge platform is still less performing with respect to the others, however it can support up to 100 athletes in less than 3200ms. According to the presented results, it is possible to reduce the shift ahead of the data window up to 3200ms in the worst case represented by the constrained Raspberry Pi3 platform with 100 connected athletes.

Figure 8 shows the volume of transmitted data of RR signals from the users to the BE-GTW or to the Azure Cloud for further processing. In particular the presented data traffic are referred to a 10 minutes window; clearly those data are the same for the three tested platforms. The two monitored

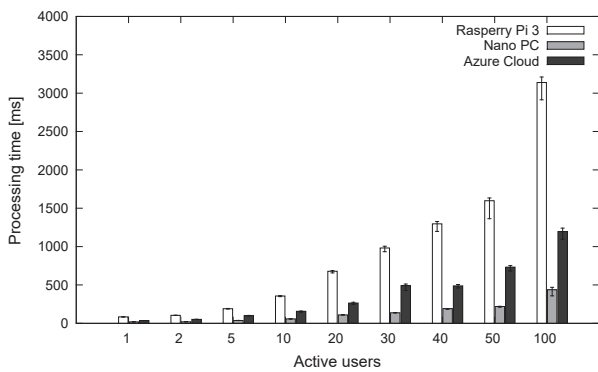


Fig. 7. Processing time increasing the number of athletes.

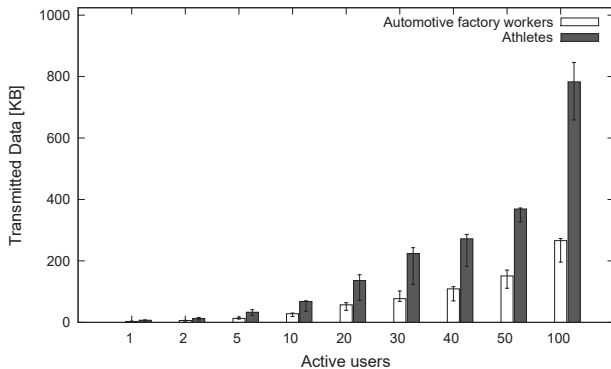


Fig. 8. Transmitted data increasing the number of users.

scenarios present the same trend but the main difference among them is due to the fact that the higher the average heart rate, the greater the amount of RR intervals to be transmitted. Of course, this analysis highlights one of the main advantage of the proposed BodyEdge architecture which consists of avoiding the data transmission on the Cloud platform.

Finally, we computed the round trip time (RTT) between the users and BE-GTW or the Azure Cloud platform to complete the overall system performance analysis. We would like to remark that the RTT delay is only due to the network propagation time for transmitted data to reach the specific processing platform and come back to the users with the processed results. Table II shows the averaged values from which it is clear that the Azure Cloud RTT delay is more than doubled respect to the Edge platforms in both tested scenarios.

## VI. OPEN ISSUES

As presented in this paper, there are clear advantages of introducing edge computing in the loop of healthcare system development; however, depending on application requirements, a number of research challenges have to be fully explored and faced yet.

TABLE II  
ROUND TRIP TIME DELAY

Platform	Workers	Athletes
Edge (Raspberry Pi3)	123ms	152ms
Edge (Nano PC)	120ms	148ms
Azure Cloud	244ms	338ms

Edge platforms for healthcare should effectively handle the gradual transition between different environments, *e.g.* when a patient leaves the highly instrumented infrastructure of a care center to enter the domestic context.

Device/patient mobility also introduces the need for seamless, dynamic execution of computational task; in fact, it is not enough to introduce computational resources between the devices and the cloud, but also to manage them according to changes in their availability.

Device integration and interoperability is another relevant challenge since new smart wearables continuously reach the market and should be added to the existing edge infrastructure which should serve as abstraction and compatibility layer.

Finally, an effective edge platform should support priority-based service administration since it should be possible to define and execute life critical services such as fall or heart failure detection with higher priority (in terms of bandwidth allocation and computation power) than *e.g.* fitness services.

## VII. CONCLUSION AND FUTURE WORKS

In this work we presented BodyEdge, a software framework to support healthcare applications in local environments with the main aim of reducing the data traffic toward Internet.

The conducted studies, through a real testbed implementation, validated the soundness of the proposed architecture, also providing a way to measure the system performance in different network load conditions and hardware platforms. Two different case studies have been defined to support the evaluation phase, respectively on workers operating in a factory and athletes training in a fitness center. The obtained results demonstrated a reasonable processing time (*i.e.* less than 3.5s) up to 100 users with the constrained Raspberry Pi3 Edge platform compared to a perceptible saving in terms of delay and data transmitted toward the Cloud.

Future works will be devoted to extend the validity and accuracy of the obtained results by conducting a test in which a significant number of people will be involved and monitored during their daily life styles. In addition, we plan to extend our framework to support dynamic adaptation according to user-defined, possibly mutable at runtime, security and privacy policies that might favor (and even force, in certain cases) the use of an Edge platform rather than a public Cloud.

## REFERENCES

- [1] marketsandmarkets.com, "IoT Healthcare Market by Component, Application, End User, and Region - Global Forecast to 2022", Technical Report, April 2017. [Online]. Available: <https://www.marketsandmarkets.com/Market-Reports/iot-healthcare-market-160082804.html>
- [2] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu "Edge Computing: Vision and Challenges", *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.
- [3] G. Aloï, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, C. Savaglio "A Mobile Multi-Technology Gateway to Enable IoT Interoperability", *IEEE International Conference on Internet-of-Things Design and Implementation (I4T)*, Berlin, April 4, 2016.
- [4] G. Aloï, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, C. Savaglio, "Enabling IoT interoperability through opportunistic smartphone-based mobile gateways", *Journal of Network and Computer Applications*, vol. 81, pp 74-84, 2017.



- [5] F. A. Kraemer, A. E. Braten, N. Tamkittikhun and D. Palma "Fog Computing in Healthcare - A Review and Discussion", *IEEE Access*, vol. 5, pp. 9206-9222, 2017.
- [6] M. Hassanalieregh *et al.* "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges", *IEEE International Conference on Services Computing*, New York, NY, pp. 285-292, 2015.
- [7] G. Fortino, D. Parisi, V. Pirrone, G. Di Fatta "BodyCloud: A SaaS approach for community Body Sensor Networks", *Future Generation Computer Systems*, vol. 35, pp. 62-79, 2014.
- [8] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu "Edge Computing: Vision and Challenges", *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.
- [9] F. Bonomi, R. Milito, J. Zhu, S. Addepalli "Fog computing and its role in the Internet of Things", in *Proc. 1st Ed. MCC Workshop Mobile Cloud Computing (MCC)*, NY, USA, pp. 13-16, 2012.
- [10] L. Cerina, S. Notargiacomo, M. G. Paccaniti, M. D. Santambrogio "A fog-computing architecture for preventive healthcare and assisted living in smart ambients", *IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, Modena, 2017, pp. 1-6.
- [11] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen "Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction", *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, 2015, pp. 356-363.
- [12] Yu Cao, Songqing Chen, Peng Hou, D. Brown "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation", *IEEE International Conference on Networking, Architecture and Storage (NAS)*, Boston, MA, pp. 2-11, 2015.
- [13] G. Fortino, R. Gravina, W. Li, C. Ma "Using Cloud-assisted Body Area Networks to Track People Physical Activity in Mobility", *International Conference on Body Area Networks (BodyNets 2015)*, pp. 85-91, 2015.
- [14] R. Gravina, C. Ma, P. Pace, G. Aloï, W. Russo, W. Li, G. Fortino "Cloud-based Activity-as-a-Service cyber-physical framework for human activity monitoring in mobility", *Future Generation Computer Systems*, vol. 75, pp. 158-171, 2017.
- [15] A. Jovic, N. Bogunovic, M. Cupic "Extension and detailed overview of the HRVFrame framework for heart rate variability analysis," *Eurocon 2013*, Zagreb, 2013, pp. 1757-1763.
- [16] M.H. Elgazzar "Perspectives on M2M protocols," *IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, pp. 501-505, 2015.
- [17] G. Sannino, G. De Pietro "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Future Generation Computer Systems*, Available on line April, 2018.
- [18] L. Bernardi, J. Wdowczyk-Szulc, C. Valenti, S. Castoldi, C. Passino, G. Spadacini, P. Sleight "Effects of controlled breathing, mental activity and mental stress with or without verbalization on heart rate variability," *J Am Coll Cardiol*, vol. 35, no. 6, pp.1462-1469, 2000.
- [19] A. Andreoli, R. Gravina, R. Giannantonio, P. Pierleoni, G. Fortino "SPINE-HRV: a BSN-based Toolkit for Heart Rate Variability Analysis in the Time-Domain," In *Wearable and Autonomous Biomedical Devices and Systems for Smart Environments: New issues and Characterization*. Chapter 19, Berlin: SpringerVerlag, Lecture Notes on Electrical Engineering, vol. 75, pp. 369-389, 2010.
- [20] P. Melillo, R. Castaldo, G. Sannino, A. Orrico, G. de Pietro and L. Pechia "Wearable technology and ECG processing for fall risk assessment, prevention and detection," *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, pp. 7740-7743, 2015.
- [21] H.K. Yang *et al.* "Application for the wearable heart activity monitoring system: Analysis of the autonomic function of HRV," *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vancouver, BC, pp. 1258-1261, 2008.
- [22] S. Lee, H. Kim, D. Kim, M. Yum, M. Son "Heart rate variability in male shift workers in automobile manufacturing factories in South Korea," *International Archives of Occupational and Environmental Health*, vol. 88, no. 7, pp. 895-902, October 2015.
- [23] K. Kostelis, J. Melnyk "Using Heart Rate Monitoring In Soccer," *ACSM American College of Sports Medicine*, January 2015. [Online]. Available: <http://certification.acsm.org/blog/2015/january/using-heart-rate-monitoring-in-soccer>
- [24] D.J. Ramos-Campo, J.A. Rubio-Arias, V. Ávila-Gandía, C. Marín-Pagán, A. Luque, P.E. Alcaraz "Heart rate variability to assess ventilatory thresholds in professional basketball players," *Journal of Sport and Health Science*, vol. 6, Issue 4, pp. 468-473, 2017.

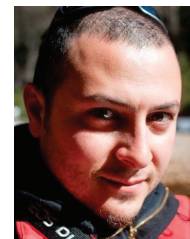
- [25] H. Zhang, M. Zhu, Y. Zheng, G. Li "Toward Capturing Momentary Changes of Heart Rate Variability by a Dynamic Analysis Method," *PLoS ONE*, vol. 10, no. 7, 2015. doi:10.1371/journal.pone.0133148.



**Pasquale Pace** (M'05) is an Assistant Professor in Telecommunications at the University of Calabria, Italy where he received his Ph.D. in information engineering in 2005. He was a visiting researcher at the CCSR, Surrey, UK and at the Georgia Institute of Technology. He has authored more than 80 papers in international journals, conferences and books. His research interests include cognitive and opportunistic networks, sensor and self-organized networks, interoperability of IoT platforms and devices.



**Gianluca Aloï** (M'02) received his PhD degree in Systems Engineering and Computer Science at DEIS Department of the University of Calabria in 2003. In 2004 he joined the University of Calabria where, currently, he is Assistant Professor of Telecommunications at the Department of Informatics, Modeling, Electronics and System Engineering (DIMES). His main research interests include spontaneous and reconfigurable wireless networks, cognitive and opportunistic networks, sensor and self-organizing wireless networks, IoT technologies.



**Raffaele Gravina** (M'16) is Assistant Professor of Computer Engineering at the Dept. of Informatics, Modeling, Electronics, and Systems of the University of Calabria (Unical), Italy. He received a Ph.D. in Computer and Systems Engineering from Unical, in 2012. His research interests include wireless body sensor networks, smart-Health, and IoT technology. He is author of over 60 papers in international journals, conferences and books. He is cofounder and CTO healthcare sector of SenSysCal S.r.l., a Unical spin-off focused on innovative IoT systems.



**Giuseppe Caliciuri** received the Master's degree in computer engineering from the University of Calabria, Italy and a certificate as industrial researcher oriented design and implementation of innovative software components for the telecommunications field, the Multimedia and the Grid. His research interests include location technique and position for smartphones, wireless sensor networks and gateway, embedded systems and software frameworks for cooperative smart objects in the Internet of Things (IoT) context.



**Giancarlo Fortino** (SM'12) is Professor of Computer Engineering at the Dept. of Informatics, Modeling, Electronics, and Systems of the University of Calabria, Italy. He received a PhD in Computer Engineering from Unical 2000. His research interests include agent-based computing, wireless sensor networks, IoT technology. He is author of over 300 papers in international journals, conferences and books. He is co-founder and CEO of SenSysCal S.r.l., a Unical spin-off focused on innovative IoT systems.



**Antonio Liotta** (SM'15) is Professor of Data Science and the founding director of the Data Science Research Centre, University of Derby, UK. He is the director of the Joint Intellisensing Lab (Europe, Asia and Australia); and a Guest Professor at Shanghai Ocean University, China and at Eindhoven University of Technology, NL. His team is at the forefront of influential research in data science and artificial intelligence, specifically in the context of Smart Cities, Internet of Things, and smart sensing. He is the Editor-in-Chief of the Springer Internet of Things book series; associate editor of the Journals JNSM, IJNM, JMM, and IF; and editorial board member of 6 more journals.