On the Deployment of Healthcare Applications over Fog Computing Infrastructure

Orestis Akrivopoulos[1], Ioannis Chatzigiannakis[3], Christos Tselios[2] and Athanasios Antoniou[1]

[1] SparkWorks ITC Ltd., United Kingdom
{akribopo, a.antoniou79}@sparkworks.net
[2] Univesity of Patras, Greece
tselios@ece.upatras.gr
[3] Sapienza University of Rome, Italy
ichatz@dis.uniroma1.it

***Abstract*—Fog computing is considered as the most promising enhancement of the traditional cloud computing paradigm in order to handle potential issues introduced by the emerging Interned of Things (IoT) framework at the network edge. The heterogeneous nature, the extensive distribution and the hefty number of deployed IoT nodes will disrupt existing functional models, creating confusion. However, IoT will facilitate the rise of new applications, with automated healthcare monitoring platforms being amongst them. This paper presents the pillars of design for such applications, along with the evaluation of a working prototype that collects ECG traces from a tailor-made device and utilizes the patient's smartphone as a Fog gateway for securely sharing them to other authorized entities. This prototype will allow patients to share information to their physicians, monitor their health status independently and notify the authorities rapidly in emergency situations. Historical data will also be available for further analysis, towards identifying patterns that may improve medical diagnoses in the foreseeable future.**

***Index Terms*—IoT, Fog Computing, Healthcare, Wearable Device, ECG, Prototype, Real-world Evaluation, 5G.**

I. Introduction

The advent of Internet of Things (IoT) and the anticipated exponential increase of interconnected devices, paves the way for the introduction of novel network architectures which aim to enhance the currently deployed cloud computing paradigm. In this new era, everyday objects, each with a unique identifier, will automatically connect to affiliated networking interfaces and will upload unprecedented amounts of diverse data. Being greatly distributed and often scarce, cloud infrastructure is incapable of handling the volume, the variety and the velocity of IoT data, especially taking into consideration the introduced latency when trying to transfer massive datasets to distant servers, or the bandwidth such transfers require. It is therefore necessary to establish a contemporary computing model, encompassing a specific breed of upgraded features deriving from the actual requirements IoT introduces to frameworks intending to capitalize on the new characteristics of the specific ecosystem.

As stated in [1] minimal possible latency, network bandwidth preservation, increased security and enhanced reliability are elements of paramount importance for any IoT-related application. Together with the necessity for data collection, storage and availability across large areas, the demand for uninterrupted services even with intermittent cloud connectivity and resource constrained devices [2], and last but not least the necessity of sometimes near-real-time data processing in an optimal manner, create an amalgam of challenges where only radical and holistic solutions apply.

Fog and edge computing in general is an emerging platform that provides computational, storage, and control resources in an intermediate layer between end-user devices and cloud computing datacenters. The physical proximity of fog infrastructure with the resource-bound last-mile sensors of any IoT-related application, allows limited latency, less bandwidth consumption, as well as elevated degrees of reliability and security. This approach extends the cloud computing paradigm by migrating data processing closer to production site, accelerates system responsiveness to events along with its overall awareness, by eliminating the data round-trip to the cloud. Offloading large datasets to the core network is no longer a necessity, consequently leading to improved safety and quality of experience (QoE) [3] . This solution confronts several of the intrinsic limitations of cloud and alleviates the deployment of services with low or even zero tolerance for error, such as industrial and healthcare applications. Due to the sensitive nature of the latter, there are additional issues that need to be considered for each use case, leading to certain limitations that prevent their rapid deployment. This paper aims to describe the most important ones and also to depict the means that fog computing provides towards addressing them properly.

The rest of the paper is organized as follows: Section II begins by presenting the dominant characteristics of fog architecture and the issues this new layer tackles towards deploying an end-to-end computing platform. Section III focuses on healthcare applications first by depicting our motivation and then by listing some of the dominant use cases that can be benefited by a potential integration of certain isolated sensors to a fog computing platform. Sections IV and V present a preliminary prototype of such an application along with the necessary evaluation method and functionality metrics. Finally, Section VI draws conclusions and summarizes the paper.

0730-3157/17 $31.00 © 2017 IEEE
DOI 10.1109/COMPSAC.2017.178

II. Dominant Fog Computing Characteristics and Solutions to IoT-related issues

Fog computing is a distributed computational paradigm that is strategically placed between IoT sensors/devices and the cloud datacenters. In particular, as shown in Figure 1, fog computing layer nodes have dedicated interfaces for communicating with the network core layer, the actual gateway of any cloud data center to the outside world. In modern networking deployments, the network core layer consists of software defined networking (SDN) nodes which facilitate extensive governance and precise supervision [4]. This approach, renders the architecture significantly more robust, due to the fact that packets originating from end devices are not directly accessing the entry point of the cloud per se, instead they undergo a second inspection process that discards all malicious, potentially harmful, or problematic content. End devices are rather easy to be compromised since they often remain unattended, giving physical access to attackers. In addition, their limited computational capacity and strict energy efficiency requirements prevent the deployment of sophisticated cyber-security or encryption mechanisms. Fog also resolves a series of IoT-related constraints as follows.

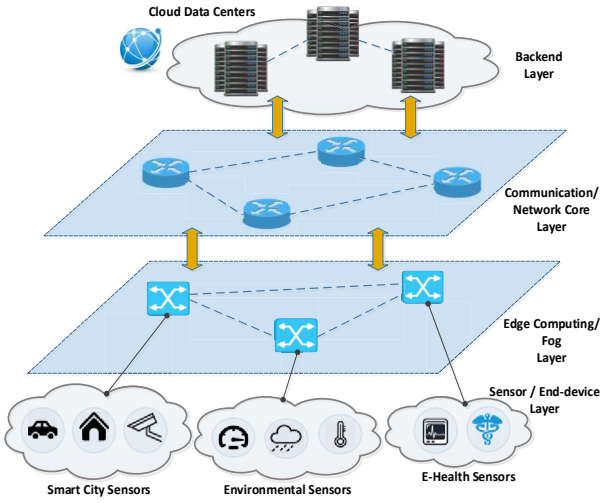

Fig. 1: Fog Computing Layer: An efficient intermediary between end-user equipment and cloud infrastructure

A. Extensive bandwidth requirements

The phenomenal growth of the IoT ecosystem towards supporting billions of devices, generates a data-oriented issue. Lots of barren datasets are collected to by the end nodes and are submitted to the cloud to be processed. Such an approach appears to be rather ineffective, since it consumes hefty amounts of bandwidth before possibly categorizing the processed data as null and meaningless. The collection rate of such datasets constantly rises, therefore a certain level of pre-processing at the edge of the network is rather compulsory. Data trimming on the edge will effectively reduce bandwidth requirements and consequently traffic costs and necessary cloud storage [5]. Dedicated fog computing nodes could alleviate computational resource provisioning in the cloud, where only valid data will be processed and categorized for a fraction of the networking expense.

B. Necessity for decreased latency and autonomous operation

As the total number of interconnected nodes increases, cloud services will encounter severe challenges towards providing uninterrupted services in cases of irregular connectivity. The advent of 5G [6] will most probably solve the majority of connectivity issues currently compromising service continuity in the cloud, but since redundancy and robustness are required in existing deployments, one could consider fog computing as an intermediate or supplementary method of addressing these issues. Many industrial or safety-critical systems such as patient monitoring platforms, automated production lines and traffic optimization applications, often require end-to-end latency of just milliseconds. This demand will be tackled by 5G, however current deployments are not yet capable to support it, rendering such systems bound to obsolete functional archetypes. Additionally, a certain level of autonomous operation is important for providing the aforementioned service continuity. Regardless of interruptions in connectivity, any safety critical system must remain operational and secure. Data accumulation should proceed and once connectivity is re-established, uploaded to the corresponding cloud repository. This is a perfect use case of how a dedicated fog node could help such situations. Obtained sensor data could be temporarily stored, potentially pre-processed in the intermediate layer, from which operators may get notifications regarding ill operation or imminent danger. Cloud connectivity is not a must and latency is decreased since only one logical hop is required for the system to provide a preliminary response.

C. Enhanced Reliability and Security prerequisites

As more data traverse through the network the possibility of errors also increases, since bit error rate, data transmission latency and packet droppings are proportional to the actual size of transmitted data. Such an increased error margin cannot be tolerated when emergency or safety critical applications rely their proper functionality on similar techniques. Uninterrupted service is of paramount importance for IoT applications, together with protecting resource constrained devices, update the security level of large distributed systems in a trustworthy manner and response to compromises without causing intolerable disruptions [2]. Fog enables service cohesion and stability by acting as complementary layer to the cloud and the necessary endpoints. Its nodes could possibly act as proxies for security updates delivery and management of sensors, perform additional security functions such as encryption or deep packet inspection and take advantage of local information and context to detect threads in near real-time. This embellished degree of functionality where resources and services of computation, communication or control, are now located closer to the users fortifies applications, boosts system awareness of end customer

needs and upgrades efficiency and performance to a whole new level.

Fog computing unveils a novel architectural concept that will most likely also enable fascinating business models for computing and networking. The major advantage of fog is no other than supporting networking in the edge, together with all the delay-critical services that can be deployed in this layer. Healthcare applications are definitely amongst the prime examples of solutions benefited by the efficient monitoring, secure and trustworthy data retrieval and seamless and continuous operation that fog computing introduces, as described in the following Section.

III. Deploying Healthcare Application Frameworks

The design of efficient health monitoring systems has been a topic of active research over the last few years, mostly reinvigorated by the numerous advances in communication protocols and access technologies. Most researchers tried to merge wireless sensor networks with smart gateways [7] [8], and also use smartphones as gateways for developing personal health monitoring systems [9]. More recently, specialized diagnosis techniques were utilized for specific types of health monitoring applications [10] or platforms using dedicated IoT communication protocols [11], however all aforementioned solutions somehow fail to harness the full extent of capabilities fog computing offers towards provide a holistic solution that will be applicable to a larger number of use cases, involving several actors.

Any advanced healthcare monitoring system must be used by both patients and doctors, offline or in real-time, and under all circumstances. Especially when doctors attempt to monitor more than one patients through such solutions, it is important to have reliable access to the provided data from each device and possibly the patients historical data that indicate the severity of the situation or similar patterns in the past. In general, a comprehensive fog-based healthcare monitoring system should support patient self-monitoring, physician off-line monitoring through accessing obtained datasets from various user devices, physician online monitoring and patient monitoring within healthcare infrastructures such as hospitals and day-care centers. This analysis leads to the accurate definition of the following four use cases:

(i) Patient self-monitoring: This use case refers to end-users that would like to perform self-monitoring of their medical conditions. These users might be either patients that have recuperated from an incident and need to monitor themselves periodically, or other individuals that need to check and be aware about their medical condition on a regular basis as part of a preventive health monitoring and early diagnosis of potentially alarming medical conditions.

(ii) Physician off-line monitoring: This scenario targets doctors that will utilize fog-enabled healthcare products to monitor their patients either at their office or at a patients home during a visit. The patients carry the wearable device at home and then, after a period of time, return the device to their physician to acquire the traces and conduct the diagnosis.

(iii) Physician on-line monitoring: The particular use case is an extension of the previous one, where the patients use the device over an extended period of time and the physician can remotely monitor the acquired vital traces via the available cloud services.

(iv) Patient monitoring within healthcare infrastructures: Being the most advanced scenario, this can take place inside an ambulance, hospital or adult day-care center whereby the wearable devices, operated by a personal health assistant or professional caregiver, instantly acquire the traces of the patient and enable real-time monitoring of the data which are stored to the cloud services.

IV. Prototype Deployment

In order to properly demonstrate the validity of the previous categorization and extract certain results on how such fog-enabled ecosystems can be deployed, the Spark IoT Platform, a prototype application which addresses all the aforementioned use cases through dedicated interfaces and subsystems was implemented. The overall architecture of this prototype is presented in Figure 2. For efficiency reasons but without the loss of generality, this prototype was focused on the extraction, analysis and storage of Electrocardiogram (ECG) signals only, however, the modular nature of its architecture allows the retrieval of any vital signal from a dedicated sensor attached to the patient's body, by simply switching the particular sensor. The other elements of the application remain operational and the data flow smoothly continues.

The fundamental components of the Spark IoT Platform prototype can be categorized in three different groups, mostly due to their physical location and functionality namely (i) the Wearable Devices, (ii) the Mobile Application and (iii) the Spark IoT Platform Core. The Wearable Devices are depicted in Figure 2 as Personal Fog, the Mobile Application is categorized as Private Edge Cloud, closely related to the fundamentals of fog computing which constitute network edge computational efficiency and interconnection, while the Spark IoT Platform Core resides in the cloud and acts as the back-end that administers all resource-demanding tasks.

A. Wearable Devices

The first logical group of devices and services is composed by wearable devices attached to the patient's body. These non-intrusive devices record certain aspects of the physiological conditions of the user using an array of sensors. Some wearable devices can be used to provide feedback to the user in case they want to (e.g., via a small touchscreen, via audio or via vibration etc.) and are capable of analyzing the traces collected from the sensors. The original traces (raw data) along with the alerts (diagnostic metadata necessary for the operation of e-agents) are stored locally on the internal storage of the wearable device. This information constitutes the short-term historic health data and belongs to the patient. The data are stored in encrypted format and can become

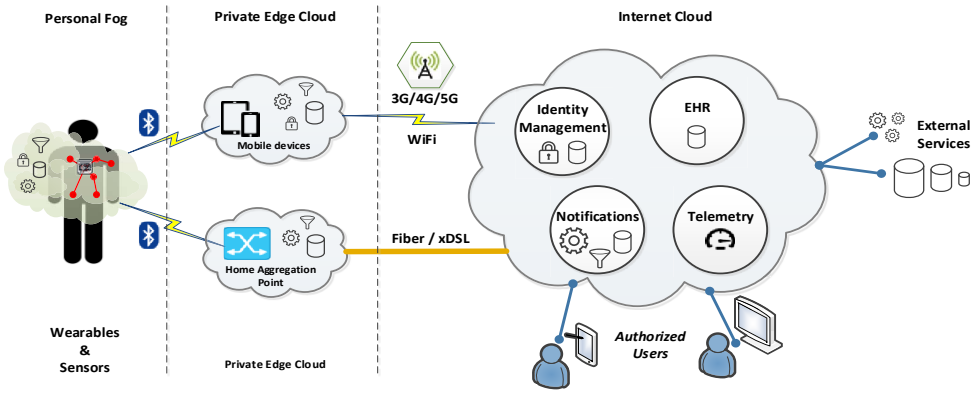

Fig. 2: Spark IoT Platform: A prototype implementation of a fog-enabled health monitoring system

available to specific services after the explicit authorization of the patient. In the specific prototype, an ECG device that provides high-definition 12-channel traces was implemented. Every 10 seconds, the signals are analyzed using a series of algorithms and a number of alerts are produced (e.g. possible ventricular hypertrophy, acute anterior myocardial infarction, arrhythmia, etc.). The traces and the alerts produced along with their timestamps are stored in the internal memory of the device.

B. Mobile Application

The mobile application is installed on the users (patients) mobile android device and is wirelessly connected to the wearable device in order to acquire the traces. Upon first installation of the mobile application, the user needs to pair the mobile device with the wearable devices following the standard BLE bonding process. As soon as the pairing process is complete, the mobile application locks the wearable device preventing it from being paired with another mobile device. The user can protect his information stored within the medical device from being accessed without permission even from his own device using all the available mechanisms from Android operating system.

The mobile application communicates with the wearable devices over a well defined API via the secure bluetooth wireless connection, can retrieve the traces and alerts either in small packages or in batch mode and the data received are stored within the mobile device's internal storage space. The mobile application can erase some or all of the data stored (a) on the wearable device and (b) on the internal store of the mobile device.

The application is capable of analyzing the data retrieved from the medical device by utilizing a series of algorithms available for Android OS or through tailor-made ones. As data is received from the wearable device, the Alert Handling component is activated to process and analyze the data and provide alerts. Data collected from the device and produced by the Alert Handling component is stored in the Data Handling component and complement those produced by the algorithms executed by the wearable device. Apart from the data transfer and management, the mobile application supports configuration/personalization tasks for the wearable device related to the memory (e.g., clean), alert generation and algorithm parametrization, battery configuration, sensors and synchronization functionality.

The user (patient) retains full control of the data retrieved from the wearable device. A user that wishes to share the information with a physician or personal health agent can use one of the following options: (a) hand-over the mobile device to the person in question to inspect the data via the mobile application, all the information stored within the mobile device become available to the third person, no data is transferred to another device; (b) produce a report including specific data (e.g., traces of a given period, specific alert categories, etc.) that are transmitted to the person requesting the data (e.g., via e-mail), only the data that the user has selected to share are transmitted; (c) unlock the wearable device and allow it to be paired with another mobile device, in this case all the information stored within the wearable device is shared with the third person.

C. Spark IoT Platform Core

The Spark IoT Platform Core offers certain features which facilitate the overall compliance of the prototype to the fog computing principles of security, efficiency and enhanced reliability, as described in Section II. In particular, it encompasses dedicated subsystems for Identity Management, Telemetry, Asynchronous Notifications and Electronic Health Records, together with the necessary interfaces for providing access to the platform to authorized users and external services.

1) Identity Management: The Identity Management service is used from the prototype system to provide authentication to all users and covers a number of aspects involving users' access to services and applications, including secure and private authentication from mobile devices and the web application or user profile management and privacy-preserving disposition of personal data. Generating a new identity requires providing a minimum set of information that is stored on an encrypted database. The information associated with the account is only accessible by the user and those users that are authorized

to do so. All other services (and users) cannot decrypt the information and thus all information accessed are anonymized.

2) Telemetry: The Telemetry service allows the user to provide access to other people (e.g. a personal health assistant) to remotely observe the measurements and alerts recorded from the wearable devices. The user needs to provide Internet connectivity to the mobile application, acquire a valid set of credentials and provide authorization to one or more users.

The Telemetry service collects the data generated by the wearable device via the mobile application that assumes the role of the so-called Fog Gateway. The mobile devices capability to transmit data to the Internet utilizing any available connection, allows the prototype platform to be fully functional both in static places such as the home/office environments and in moving environments such as an ambulance during submission.

3) Asynchronous Notifications: The Asynchronous Notifications service allows the user to provide access to other people (e.g. physician, caregiver) to remotely receive notifications related to alerts generated by the wearable devices and/or mobile application. Similar to Telemetry, the user needs to provide Internet connectivity to the mobile application, acquire a valid set of credentials and provide authorization to one or more users to use the Asynchronous Notification service.

A notification is triggered when one or more alerts generated by the wearable device and/or mobile application is above or below a threshold defined by the user (e.g., bpm above 150 or below 40). The Asynchronous Notification service allows the definition of complex criteria that require the computation of a formula involving the alerts generated by the device (e.g. arrhythmia detected and bpm increasing). The remote user receives notifications through a mobile application by utilizing the Push Notification mechanism.

4) Electronic Health Records: The Electronic Health Record (EHR) service provides the creation and maintenance of the electronic patient record. An electronic health record, is a systematic collection of electronic health information about an individual patient. It is a record in digital format that is capable of being shared across different healthcare settings. The EHR service provides support for the patient summary, both the basic and the extended versions.

The user can authorize other users (e.g., treating doctor, personal health assistant, caregiver etc.) to acquire direct access to all available EHR, or authorize the EHR to be exported (or even synchronized) with external medical record systems stored at hospitals, clinics etc. by utilizing innovative cryptographic mechanisms.

V. Evaluation

For the prototype evaluation process, measurements on the custom-made ECG device were conducted, under different operating parameters, in particular the sampling rate and the number of channels. The accepted values for the number of channels is 6 and 12, while for the sampling rate 500Hz and 1kHz. Table I summarizes the values for the maximum recording length using the integrated 4Gbit (8x512Mb) NAND Flash Memory available on the device. The device lifetime in terms of battery power is also evaluated, in particular the actual energy consumption of the most accurate sampling rate of 1kHz for 12 channels, in both operation modes as shown in Table II.

TABLE I: Maximum Recording Length

Channels	Sampling Rage	Duration
12	1kHz	14.9 hrs
12	500Hz	29.81 hrs
6	1 kHz	59.62 hrs
6	500Hz	119.42 hrs

TABLE II: Maximum Liferime Operation

Operation	Device Lifetime
Real-time data transfer, 12 channels, 1kHz sampling rate	7 hrs
Non-real-time data transfer, 12 channels, 1kHz sampling rate	19 hrs

The Mobile Application was evaluated by measuring the total memory size needed for storing an ECG trace of various channels and two distinct sampling rates (1kHz and 500Hz) using EDF+ format. All ECG traces had a total duration of 1 minute. It is possible to use compressing techniques for further decrease the memory size needed, however this would have an immediate affect on battery life. Data transmission rate was also measured for the ECG device in two distinct scenarios. The wearable ECG device can transmit data to a smartphone device either while in monitoring mode, with data of the current recording transmitted in real time, or in download mode, with data of a previous recording being transmitted from the ECG device's memory. While the device is transmitting data of a recorded ECG, the wearable ECG device can be either in idle state or in active recording mode. In the latter case, the wearables ECG devices software (firmware) limits the data transmission rate to the smartphone device to ensure the accuracy of the data sampling of the active recording.

TABLE III: ECG trace storage requirements

ECG trace parameters	Total Size
12 channels, 500Hz sampling rate	724.3KB
12 channels, 1kHz sampling rate	1.405MB
6 channels, 500Hz sampling rate	365.414KB
6 channels, 1kHz sampling rate	722.8KB

Additionally, the data transmission rate varies based on whether the ECG device stores the data while transmitting them or the data sent directly without storing in the internal memory. In the first case, the data transmission happens after the data is stored in the memory (one entry in the memory happens only after a whole data page is filled) and the data are read from their stored position to form the transmission packages with a predefined number of samples defined by the devices firmware. In the second case, the data transmission happens directly after the data are sampled by the ADC

converter and again a predefined number of samples is gathered. The predefined packet size for the data transmission is calculated based on the sampling rate, the number of channels used in the recording and the predefined maximum number of transmissions per second.

Tables IV and V show the data read rate from the ECG device to the smartphone application for different configurations of the ECG device as it was benchmarked from the smartphone's side. We need to note here that during the data transmissions we do not only transmit and count the data packets but the control and acknowledgement packets, battery level state report packets and reconnection packets (in the chance of disconnection). Those packets are sent rarely but can affect the theoretically available throughput in the communication between the ECG device and the smartphone.

TABLE IV: ECG trace direct transmission

ECG Parameters	Memory Storing	Average Throughput
12 ch., 500Hz sampl. rate	yes	35.733 kbps
12 ch., 500Hz sampl. rate	no	46.433 kbps
12 ch., 1kHz sampl. rate	yes	75.089 kbps
12 ch., 1kHz sampl. rate	no	96.597 kbps
6 ch., 500Hz sampl. rate	yes	10.510 kbps
6 ch., 500Hz sampl. rate	no	14.268 kbps
6 ch., 1kHz sampl. rate	yes	21.819 kbps
6 ch., 1kHz sampl. rate	no	26.072 kbps

TABLE V: Stored ECG trace transmission

ECG Parameters	Sampling/ Recording	Average Throughput
12 ch., 1kHz sampl. rate	yes	77.554 kbps
12 ch., 1kHz sampl. rate	no	639.087 kbps
6 ch., 500Hz sampl. rate	yes	12.047 kbps
6 ch., 500Hz sampl. rate	no	1.455 Mbps

Any authorized user with access to the web portal of the Spark-Heart prototype, available trough the cloud is capable of issuing commands to the Mobile Application as well as the Wearable Devices, which traverse through the dedicated interfaces of the Core. In order to assess the end-to-end response time of such a request, we calculated both the average and maximum response time of simple status requests (e.g. remaining memory, battery level, ECG trace counts) towards the ECG device along with the average and maximum time the system requires to issue a complete report of the current ECG device state. These findings are summarized in Table VI.

TABLE VI: Average and Maximum Response Time

Command Type	Average Resp.	Max. Resp.
Simple	21.388407 ms	48.809636 ms
Complete Report	106.974583 ms	128.340158 ms

VI. Conclusions

This paper has presented some of the basic issues a system architect must consider when designing, implementing and deploying an end-to-end healthcare application which includes IoT nodes and cloud computing backend services, that leverages the benefits of the Fog computing approach. Being an intermediate layer between end-user devices and the remote cloud datacenters, Fog alleviates a series of issues in the areas of security, scalability, bandwidth consumption reduction, latency decrease and seamless operation. However, one should focus on the actual problems derived from the fundamentals of IoT applications. The huge increase of interconnected devices indicated that holistic solutions are needed for efficiently solve the problems of colossal data transfer between the network nodes. As shown by the evaluation of a functional end-to-end application prototype designed by the latest trends of IoT and Fog computing, presented in Section V, a single ECG device requires significant amounts of storage and throughput to deliver adequate ECG traces to the system. Considering that these numbers will increase linearly to the number of interconnected ECG devices and users in general, maybe an even more radical approach than Fog computing is advised.

Acknowledgement

This work has been partially supported by the H2020 Grant Ageement 644090 EuroCPS/IE-7-CARDIO and the research project Designing Human-Agent Collectives for Sustainable Future Societies (C26A15TXCF) of Sapienza University of Rome.

References

[1] Cisco Systems Inc., "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are," Available: http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, [Online].

[2] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[3] C. Tselios and G. Tsolis, "On QoE-awareness through Virtualized Probes in 5G Networks," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on*, 2016, pp. 1–5.

[4] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[5] A. Papageorgiou, B. Cheng, and E. Kovacs, "Real-time data reduction at the network edge of internet-of-things systems," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 284–291.

[6] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiaraviglio, P. Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni *et al.*, "Superfluidity: a flexible functional architecture for 5g networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1178–1186, 2016.

[7] Y. Chen, W. Shen, H. Huo, and Y. Xu, "A smart gateway for health care system using wireless sensor network," in *SENSORCOMM 2010*, July 2010, pp. 545–550.

[8] S. Mohapatra and K. S. Rekha, "Sensor-cloud: a hybrid framework for remote patient monitoring," *International Journal of Computer Applications*, vol. 55, no. 2, 2012.

[9] S. Yang and M. Gerla, "Personal gateway in mobile health monitoring," in *2011 IEEE PERCOM Workshops*, March 2011, pp. 636–641.

[10] T. N. Gia *et al.*, "Fog computing in healthcare internet of things: A case study on ecg feature extraction," in *2015 IEEE CIT/IUCC/DASC/PICOM*, Oct 2015, pp. 356–363.

[11] D. Yi, F. Binwen, K. Xiaoming, and M. Qianqian, "Design and implementation of mobile health monitoring system based on mqtt protocol," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Oct 2016, pp. 1679–1682.