Design of a RESTful Middleware to Enable a Web of Medical Things

N. Philip, T. Butt,
D. Sobnath, R. Kayyali, S. Nabhani-Gebara and B. Pierscionek
Digital Media for Health Research Group, MINT center, Faculty of Science, Engineering and Computing, Kingston University, Surrey, UK,
n.philip@kingston.ac.uk

I. Chouvarda, V. Kilintis, P. Natsiavas and N. Maglaveras
Lab of Medical Informatics, Aristotle University Thessaloniki, Greece.
ioanna@med.auth.gr

A. Raptopoulos
Exus
Athens, Greece,
arap@exus.co.uk

The WELCOME Consortium*

***Abstract*— In this paper, we consider the design methodology of a mobile patient hub for the remote self-management of COPD patients. The patient hub design forms a part of the WELCOME system. WELCOME is a current EU project that aims to design and develop a new mobile health system to provide integrated care for COPD patients with comorbidities. The approach adopted for this research is based on the Web of Things architecture with RESTful principles as the enabler of communications. The proposed patient hub architecture design is based on three layers: an application layer, a middleware layer and the sensors layer. This paper presents the detail of the initial design of the middleware and an analysis of the architecture in the context of the system's requirements.**

Keywords-component; COPD; Patient-hub; mhealth; Web of Things; RESTful; Integrated care; WELCOME project; PAN; BAN;

I. INTRODUCTION

Chronic obstructive pulmonary disease (COPD) is a chronic condition that requires integrated care and a collaborative multidisciplinary approach to its management. WELCOME is a European project to design and develop a mobile health system that provides such a care approach for COPD patients. In particular, WELCOME is based on wearable sensing and cloud computing to provide proactive and patient-centered remote self-management for COPD patients with co-morbidities (www.wecome-project.eu) [1].

One of the main building blocks of the WELCOME system is the patient hub, which represents a mobile platform that includes all of the software and hardware components that interact with the patient and the sensors, including: a) a unified data-acquisition application to collect the medical data from the medical sensors and then format, prepare and submit those data to the WELCOME cloud, and b) interactive applications to collect information and present users with the results, including alarms, motivation messages, and guidelines for the patient in intuitive ways with a capability of providing feedback, etc.

Figure 1 depicts the overall patient hub functionality, including the foreseen connections with the different types of sensing devices. The connection between the patient-hub resources (e.g., the patient's applications and the medical sensors) to the cloud will be based on the RESTful paradigm [2]. In such an architecture, all of the acquired data from the medical sensors and the applications will be treated as Web resources and, hence, a Web of Things architecture will be created. As shown in Figure 1, the WELCOME Application Hosting Device (WAHD) will host the patient's hub software platform that will be able to access the patient's applications on the WELCOME cloud. On the other hand, communicating with the medical sensors (e.g., the glucose meter, BP, weight scale and the temperature sensors) via the Bluetooth (BT) API, while communicating with the Vest and the Inhaler via the Peer-to-Peer (P2P) Wi-Fi and Bluetooth Low Energy (BLE) APIs.

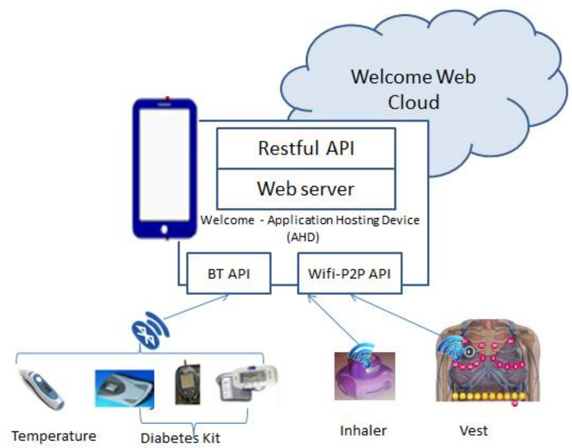

Figure 1. Patient hub of the WELCOME system.

This paper presents the initial design of the patient hub for the WELCOME system with a great emphasis on the proposed middleware layer. Driven by this Web-of-Things (WoT) vision, the proposed platform architecture can be divided into three layers: an application layer, a middleware layer and a sensors layer. The remainder of this paper is organized as follows: Section II presents a summary of related work. Section III identifies the user and system requirements. Section IV describes the proposed system architecture for the middleware. Finally, section V concludes the paper and summarises the ongoing work.

II. RELATED WORK

Existing literature contains a plethora of research efforts offering middleware solutions with varied scopes [3]. The main goal of most of the existing middleware systems and research projects was to build a unified adaptation layer framework to support the important needs of the respective

MOBIHEALTH 2014, November 03-05, Athens, Greece

DOI 10.4108/icst.mobihealth.2014.257408

domains. One group of middleware solutions aims to use data from only the internal sensors of smart phones to obtain and analyse data about user behavior and the surrounding environment [3]. However, the focus of these solutions is only limited to the usage of the internal sensors of a mobile phone. On the other hand, the solutions with much broader scope targets fulfill the middleware requirements for the Internet of Things (IoT) [3]. Some of these efforts focus on providing integration to the existing middleware. For example, Mosden [4] is an middleware application that collects sensor data from external and mobile phones' internal sensors and uploads it to the cloud IoT middleware, but it does not support any local mobile phone application.

There are many solutions that focus on the patient health monitoring [5]. Wagner et al. [6] propose an Android-based BAN for tele-medical systems, which is responsible for data acquisition from a ZigBee-based BAN and analysis before it is forwarded to the medical server. However, this solution assumes that a smart phone will solely perform all of the analysis of the sensor data and will store it locally, which is not possible in WELCOME's context where BAN data is generated at the rate of 2Mbps and consists of 26 sensors. Lifeware [7] proposes a framework for body sensors and has been demonstrated for use in monitoring sportsmen/women. The framework is effective, but it is optimised to be used in a particular domain and, thus, has limited applicability. Morón et al. [8] propose a smart-phone-based tele-care system that uses Bluetooth-enabled body sensors for the monitoring and management of chronic diseases. The body sensor data is collected by a node coordinator (i.e., a smart phone) before forwarding it to the central server. However, this effort does not address the data requirements of any local patient applications. Moreover, this effort focuses more on analysing the impact of different implementations of the solution in Java and Python. MyHealthAssistant [9] is an event-based middleware that runs on an Android smart phone to monitor a user's daily activities. It is strongly coupled with the Android operating system to enable an event-based publish/subscribe paradigm for multiple applications. However, it does not describe any data-forwarding approach that is required in the event of an integrated healthcare system.

Our middleware solution is focused on collecting medical data from a personal area network (PAN), which consists of different sensors, as well as a body area network (BAN) with 26 sensors, and provides their abstractions for supporting multiple mobile health applications while following the RESTful paradigm. The local mobile applications benefit from the locally available RESTful web server. The middleware also distributes the data processing task with the cloud and, hence, offers more scalability.

III. Requirements and Challenges

Based on the outcomes of the user requirements phase of the WELCOME project, in this section we identify the main non-functional requirements of the proposed patient hub's middleware layer architecture.

- Heterogeneity: The design should be extensible to support heterogeneous wireless protocols, sensor devices and BAN.
- Mobility: The patient hub application must run unobtrusively on a mobile device to monitor different health-related attributes while supporting the patient's mobility at home (e.g., engaging in simple home activities such as gardening).
- Interoperability: It must be fully interoperable with the cloud, and must synchronise the recorded medical data to the cloud.
- Adaptability: (handling changing sensor configurations): The architecture should be modular and extensible to accommodate future devices and changes.
- User Experience: The reliability and timeliness are the most important performance requirements identified by the WELCOME user, in addition to measures to meet high accessibility and usability of the patient hub.
- Security: As typical requirements in any e-health platform, considerable security and privacy are required to attain confidentiality, integrity and availability.

IV. Proposed RESTful Middleware

The central role of the middleware, which is hosted by the patient hub, is to act as a mediator between the cloud and sensor devices by collecting the sensed data and uploading the information to the cloud platform. The cloud platform does most of the processing of the complex data recorded by the vest and enables the applications to request data using RESTful [10] Web services. However, it is clear that there will not be a single mobile application that will monitor patients' conditions. Instead, the system will be modular to support multiple, very specific applications for patients depending upon their profiles. This fact hints at the inherent resource wastage involved in scenarios where multiple applications on a single mobile phone repeatedly request the same data from the cloud.

The proposed middleware is based on RESTful Web services that enable the WoT paradigm [11] at the mobile device to cope with the interoperability challenge by offering a standard service interface. This paradigm treats all of the available services as resources and provides URIs to interact with those resources using HTTP, which significantly reduces the required application development time. Furthermore, abstraction of sensors as resources can be further useful to enable the creation of physical mashups [12]. Mobile applications request the data from the locally available RESTful Web server, which fulfils the request either by using local storage or a cache, or by acting as a proxy to get the data from the cloud. Consequently, it eliminates the need for redundant requests from multiple applications that require the similar data on a single mobile device.

A. Architecture

Figure 2 shows the layered architecture of the proposed middleware. The details of its modules are covered in the rest of this section.

Communication Manager: This module manages the communication with heterogeneous sensors, BAN and the central cloud. The communication with the sensing layer is carried out for data collection to provide an abstraction of the

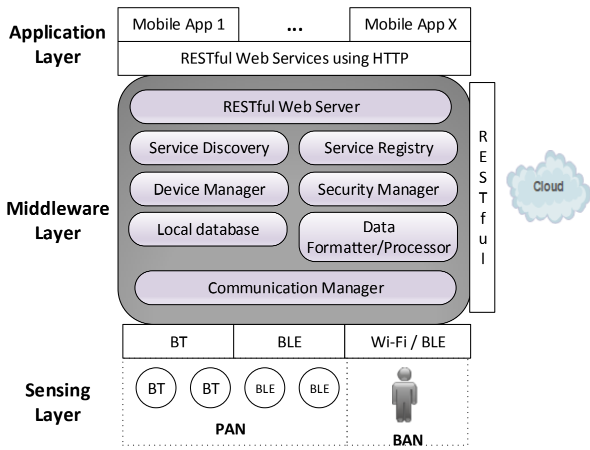

Figure 2: Architecture of the middleware

sensing layer, regardless of the underlying protocols that are being used. The sensors can use different wireless protocols, including Bluetooth (BT), Bluetooth Low Energy (BLE) and Wi-Fi, etc. The communication manager uses different protocol APIs to communicate with the respective devices. The process is further divided in to two modules: the transporter starts communication by calling a particular API to use the wireless medium transport to gather data, and then the translator applies higher-level knowledge to parse the data for a specific device type. This causes the communication manager to be extensible, as new devices can be integrated to the middleware by updating the device profiles of the translator. The gathered data is then made available to the upper layers while abstracting the heterogeneous devices. This abstraction helps the higher-level applications to perceive any type of device in the same manner, even if the underlying communication protocol and implementation are different.

In the context of WELCOME, we have medical sensors that use Bluetooth and Bluetooth Low energy (BLE), and BAN that requires a mixture of Wi-Fi and BLE communication for data collection. Most of the traditional Bluetooth-based medical sensors implement Serial Port Profile (SPP) and few of the continua-based sensors also offer a health device profile (HDP) stack for interoperability. Moreover, there is a variety of new devices that are trending towards the support of BLE or Bluetooth Smart because of its energy efficiency. On the other hand, the communication with the BAN, which is hosted by the patient's vest, requires special treatment because of its unique characteristics. The BAN generates data at the rate of 2Mbps and for storage in a local Wi-Fi SD card. The card is capable of creating an ad-hoc Wi-Fi network to deal with the inherent mobility and high data-rate requirements. However, it is impractical to keep the radio of Wi-Fi SD card constantly switched on for a battery-operated vest. Therefore, the vest is initially paired with the communication manager by using BLE. After recording the data for a pre-determined amount of time, the vest notifies the patient hub by using energy-efficient BLE. The radio of the Wi-Fi SD card is only switched on to create an ad-hoc Wi-Fi network after receiving an acknowledgement from the communication manager. The whole process is based on a PUSH-based design to minimise the energy cost of Wi-Fi direct. The communication manager acts as a transporter by creating listening threads for different protocols and then as a translator by deciphering the received content through parsing.

Another important job of the communication manager is to sync the acquired data with the cloud where data is further processed and made available to the middleware and other applications, including social carergivers, doctors and hospitals.

Device Manager: This component stores device-related information, including its ID, name, MAC address and other context attributes. The device manager is updated whenever any successful communication with a device is completed. This information is useful for the communication manager to schedule the listening for different devices.

RESTful Web Server: This module enables the WoT paradigm by following the RESTful Web architecture. It makes devices an integral part of the existing Web and its infrastructure by using HTTP as an application layer protocol. All of the functionalities or services of a device are made available as resources with unique identifiers (URIs) and offer a standard interface for interaction. The RESTful interoperable interface has simple operations, including GET (retrieve the current value of a resource), PUT (change a resource value), POST (create or update an attribute of a resource) and DELETE (delete a resource attribute). Furthermore, the Web server is capable of serving requests for resource representations in different formats including XML, SOAP and JSON.

In case of WELCOME, the BAN data is actually an aggregated format of 26 sensors; thus, it will need further processing at the cloud before it becomes available for applications. Therefore, the Web server will only have data of a few medical sensor devices available locally and will need to get other data, such as ECG, from the cloud. In that case, the Web server will act as a proxy to serve a request. However, the acquired data from the cloud is stored in the cache to serve similar future requests. The Web server will be built upon Restlet[1] Engine, which offers comprehensive capabilities to leverage REST and has a mobile platform-specific version as well.

Service Registry: This module consists of a service registry where all of the service information is stored. In WELCOME's context, the offered services are blood pressure, SpO2, etc. The service registry is updated in two ways: the communication manager updates it when it reads a new service from a device, and the Web server updates it when a

[1] http://www.restlet.org/

new service request is served after downloading data from the cloud. It also holds the cached values that are downloaded by the Web Server from the cloud.

Service Discovery: Mobile applications at the application layer require a URI to access resources at the RESTful Web server. The process of service discovery helps applications to find the URI for a resource; that is, an application that needs a blood pressure reading will use the service discovery to get the URI of the available resource at the server that offers such a blood pressure reading.

Data Formatter/Processor: This module serves the data formatting and processing requirements of the middleware. In the context of the WELCOME project, medical data needs to be encoded/decoded such as, for example, in JSON schemas that are based on the specified cloud communication API, whenever communication with the cloud takes place. Furthermore, the data processor is useful for understanding and sometimes segregating the received aggregated medical data from the BAN before converting it to a specific format (e.g., EDF) of equal time segments. Moreover, this module is responsible for performing the data compression before forwarding the data to the cloud.

Security Manager: It manages the security aspects, including authentication, authorisation and encryption. It is used to define whether or not an application is allowed to request a resource and determine what permissions it is granted. It also provides encryption keys to establish secure communication channels between applications and the middleware.

V. CONCLUSION

Driven by the future vision of WoT architectures, the RESTful approach and the requirements of integrated COPD management, an initial design of a multi-tier patient hub for the WELCOME system is presented in this paper. The detail of the initial architecutral design of the multi-tier architecture is described. It covers several requirements, including interoperability, heterogeneity, security, adaptability, interoperation, device and service discovery and management and management of data volume and user experience. This is ongoing work and the final detailed design specification is still underway. The future work will introduce improvements in the current design of the architecture by analysing the feasibility of a publish/subscribe scheme by using application-specific profiles in which applications provide a list of required sensor data and middleware can proactively download the data based on the requirements. The middleware, in this case, will notify the applications and exploit the data processing to serve multiple applications using the same data. Then, the detailed design of the final architecture will be completed. Finally, the solution will be implemented and its performance will be analysed in the different scenarios of the WELCOME project.

ACKNOWLEDGMENT

This work was supported by the European commission (Call Identifier FP7-ICT-2013-10). *The WELCOME consortium consists of Exus, Centre Suisse d’Electronique et de Microtechnique, Kingston University London, Aristotle University of Thessaloniki, Inventya Ltd, Christian-Albrechts-Universität zu Kiel, Universitätsklinikum Schleswig-Holstein, Royal College of Surgeons in Ireland, Smartex S.R.L., Ciro+ B.V., Kristronics GmbH, Universidade de Coimbra and Croydon Healthcare NHS Trust.

REFERENCES

1. I. Chouvarda, N. Y. Philip, et, al., ‘Innovative Integrated Care platform using Wearable Sensing and Smart Cloud Computing for COPD patients with Co-Morbidities’, The 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC’14) Chicago, USA.
2. D.Uckelmann, M. Harrison, F. Michahelles, “Architecting the Internet of things”, 1st ed., Springer-Verlag, 2011.
3. S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, “Role of Middleware for Internet of Things: A Study,” Int. J. Comput. Sci. Eng. Surv., vol. 2, no. 3, pp. 94–105, 2011.
4. C. Perera, P. P. Jayaraman, and P. Christen, “MOSDEN : An Internet of Things Middleware for Resource Constrained Mobile Devices,” in 47th Hawaii International Conference on System Sciences (HICSS).
5. A. Pantelopoulos and N. G. Bourbakis, “A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis,” IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev., vol. 40, no. 1, pp. 1–12, Jan. 2010.
6. M. Wagner, B. Kuch, C. Cabrera, P. Enoksson, and A. Sieber, “Android Based Body Area Network for the Evaluation of Medical Parameters,” in Proceedings of the Tenth Workshop on Intelligent Solutions in Embedded Systems (WISES), 2012, pp. 33–38.
7. L. Rodr’iguez-Molina, Jes'us and Mart'inez, Jos'e-Fern'an and Castillejo, Pedro and L'opez, “Combining Wireless Sensor Networks and Semantic Middleware for an Internet of Things-Based Sportsman/Woman Monitoring Application,” Sensors - Multidiscip. Digit. Publ. Inst., vol. 13, no. 2, pp. 1787–1835, 2013.
8. M. J. Morón, A. Gómez-Jaime, J. R. Luque, and E. Casilari, “Development and Evaluation of a Python Telecare System Based on a Bluetooth Body Area Network,” EURASIP J. Wirel. Commun. Netw., vol. 2, pp. 1–10, 2011.
9. C. Seeger, K. Van Laerhoven, and A. Buchmann, “MyHealthAssistant: An Event-driven Middleware for Multiple Medical Applications on a Smartphone-mediated Body Sensor Network.,” IEEE J. Biomed. Heal. informatics, pp. 1–9, May 2014.
10. R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” University of California, 2000.
11. D. Guinard and V. Trifa, “Towards the web of things: Web mashups for embedded devices,” in in Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM), in proceedings of WWW , Madrid, Spain, 2009.
12. D. Guinard, V. Trifa, F. Mattern, and E. Wilde, “From the Internet of Things to the Web of Things : Resource Oriented Architecture and Best Practices,” in Architecting the Internet of Things, Springer Berlin / Heidelberg, 2011, pp. 97–129.