AppA: Assistive Patient Monitoring Cloud Platform for Active Healthcare Applications

Venki Balasubramanian
Centre for Informatics and Applied Optimisation
Federation University, Ballarat, Australia
+61 3 53276407
v.balasubramanian@federation.edu.au

Andrew Stranieri
Centre for Informatics and Applied Optimisation
Federation University, Ballarat, Australia
+61 3 53279283
a.stranieri@federation.edu.au

Ranjit Kaur
Centre for Informatics and Applied Optimisation
Federation University, Ballarat, Australia
+61 3 53276407
ranjitkaur@federation.edu.au

ABSTRACT

Continuous, remote monitoring of patients using wearable sensors can facilitate early detection of many conditions and can help to manage the growing healthcare crisis worldwide. A remote patient monitoring application consists of many emerging services such as wireless wearable sensor configuration, patient registration and authentication, collaborative consultation of doctors, storage and maintenance of electronic health record. The provision of these services requires the development and maintenance of a remote healthcare monitoring application (HMA) that includes a body area wireless sensor network (BASWN) and Health Applications (HA) to detect specific health issues. In addition, the deployment of HMAs for different hospitals is not easily scalable owing to the heterogeneous nature of hardware and software involved. Cloud computing overcomes this aspect by allowing simple and easy maintenance of ICT infrastructure. In this work, we report a real-time-like cloud based architecture known as Assistive Patient monitoring cloud Platform for Active healthcare applications (AppA) using a delegate pattern. The built AppA is highly scalable and capable of spawning new instances based on the monitoring requirements from the health care providers, and is aligned with scalable economic models.

Categories and Subject Descriptors

D.2.11 [**Domain-specific architectures**]: Healthcare Cloud – *initialization, maintenance and scalability.*

General Terms

Performance and Design

Keywords

Cloud based computing; health condition; early detection, BAWSN.

1. INTRODUCTION

The early detection of medical conditions can lead to their prevention, more effective treatment and cost savings. For instance, the early detection of diabetic neuropathy was found to reduce complications and save substantial cost and suffering [1]. Remote monitoring of fluid buildup can lead to early detection of cardiovascular conditions with de-fribulated patients [2]. Sudden cardiac death often does not come ‘out of the blue’ and patients have symptoms for as long as two hours before cardiac death occurs [3]. Further, changes in heart rate variability have been associated with sepsis in hospital settings [4] and sensors have been developed to detect biomarkers for prostrate cancer [5].

Continuous monitoring of health conditions with the use of wearable sensors that stream data via wireless networks to repositories that are accessible by health care providers is emerging as a technology that promises to lead to new ways to realize early detection of conditions [6]. The approach combines a body area wireless sensor network (BAWSN) [7] with healthcare applications (HA) that are customized to monitor for general health or specific diseases into a Healthcare Medical Application (HMA).

Ultimately, we can expect the emergence of a multitude of condition specific applications, each using different subsets of each patient’s health data commissioned by diverse healthcare practices. For instance, a rehabilitation clinic may be interested in tracking a patient’s gait, while a counselling service may be interested in tracking heart rate variability to detect suicidal depression and a hospital may be interested in detecting sepsis. In addition, the HA can be expected to send alert messages to the patient as well as to the doctors in case of emergency.

Deployments of HAs with a BAWSN are still in their infancy owing to challenges that include:

- The selection and configuration of wearable wireless sensors and smartphones devices that will securely and robustly capture and process streams of health data [8]
- The design of an architecture that enables health care providers and patients to sign up, start and terminate applications with minimal overhead or cost
- The establishment of economic models that ensure a BAWSN/HA approach is scalable to millions of patients in cost effective ways
- The provision of high levels of security over patient’s private health data

- The provision of inter-operability standards for the collection of real time data for integration into electronic medical records
- The development of new analytics and data mining algorithms that identify insights while data is streamed

Pilot systems including a maternal pregnancy application [9] and cloud based diabetes detection using a BAWSN by [10-13] have emerged that demonstrate technical feasibility for in small scale settings.

Health care providers including hospitals and health care practices are unlikely to be in a position to maintain the data repositories required to receive, process and store streams of patient data. This represents an additional overhead to their existing ICT operations and stretches already thin budgets. In addition to technical challenges including connectivity and security issues inherent in the HA with BAWSN approach, the health economics case looms as a large obstacle.

In this article, we outline requirements for a scalable, secure, cloud based approach to the implementation of a HMA. The business model is assumed to be a pay-per-use model of the cloud services and repositories that will enable convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services). These can be rapidly provisioned and released with minimal management effort or service provider interaction. The hospital can make use of this business model to host the HAs in the cloud and have service level agreement (SLA) for the usage. This paper not only emphasis and applies this business model in healthcare application but also proposes an innovative way to leverage the cloud resources based on the healthcare monitoring needs.

The remaining of this paper is organised as follows: The requirements for a HMA are described in the next section. Following that, a Design pattern, for the implementation is presented in Section 3. Before describing the implementation results of AppA using Shimmer sensors and the Australian Nectar research cloud in Section 4. The architecture design of AppA is detailed in Section 5. The formulation of the cost and the justification are given in Section 6 and 7 respectively. The conclusion and future work are described in Section 8.

2. REQUIREMENTS FOR A HMA, BAWSN

Simple estimates illustrate that the provision and maintenance of a HMA and BAWSN facility by a single Health care provider is unrealistic. For instance, the Royal Melbourne Hospital in Victoria Australia discharges around 60,000 patients every quarter [15]. In a context of widespread use of remote patient monitoring, we may assume that 10% of patients discharged are provided with wearable sensors for an average duration of 30 days. If the sole data collected is heart rate variability with an eMotion device [16] that typically generates a 7Mb text file per hour of continuous monitoring and we assume monitoring for 20 hours per day, then we can expect 280,000 Mb of data to be generated per quarter for the hospital from patients wearing this one sensor. If a single health application is running for each patient as it would be for example with remote monitoring of heart rate variability, then over 60 HMA applications will be running in parallel processing the data as it streams in, for this one hospital alone. This requires considerable computational resources and built in redundancy that would severely challenge in-house IT services. In addition, other health care providers are likely to emerge with a desire to provide ancillary services such as emergency fall alert services with the same data streams.

Commercial analyses by IBM [17] identify consumer groups that are not chronically ill but desire real time additional information about their health status and suggest that consumers are willing to pay for this additional service so long as the data can be secure and accessible to a range of health care providers. A pay-as-you go model for Cloud services in UK health care has also been described by [18] as the most viable model.

A high level requirement for a HMA includes:

- the facility for diverse HA's managed by different providers to execute on different subsets of the data
- the ability for each HA to start and stop with very limited overhead resources or cost
- the capacity to ensure that each HA accesses only the data it is authorized to access
- the capacity for any health care provider to spawn a new HA with limited overhead or cost
- the capacity to define or ramp up to ensure a high level of security
- technologies that support business models include 'pay as you go' and other models that ensure sustainability

In the next section, we describe a Design Pattern as a key element of the architecture that realizes the requirements.

3. DESIGN PATTERN

In software engineering, a design pattern represents a solution to a problem or class of problems that one can put to work at once in the programming code. Design patterns were not invented based on theories. Rather the problem situation occurred first based upon the requirement context, some design solutions were evolved. Design patterns often provide design solution for common design problems faced by the application developer. There are many design patterns available related to object oriented programming world, and in time, a particular set of them has become accepted as the standard set [19, 20]. One of the standard design patterns is the delegation pattern in object-oriented programming where a class, instead of performing one of its stated tasks, delegates that task to an associated helper class. There is an Inversion of Responsibility in which a helper class, known as a delegate, is given the responsibility to execute a task for the delegator. The delegation pattern solves the common design problem for a class with excessive functionalities. If the class is highly loaded with different functionalities the delegation pattern creates a delegate to do some of its functionalities. A simple pseudo code is given in Fig. 1 that shows the client program calls the delegator Printer class, which in turn calls the

IMCOM (ICUIMC)'15, January 8–10, 2015, Bali, Indonesia.

RealPrinter class to print the output. To the client it appears that the Printer class is doing the print, but the RealPrinter class is the one actually doing the print. In this way, the delegator (Printer class) delegates the print function to the delegate (RealPrinter class).

```
class RealPrinter { // the "delegate"
   void print() {
        print "something" ;
     }
}
class Printer { // the "delegator"
     // create the delegate
     RealPrinter p = new RealPrinter();
     void print() {
          p.print(); // delegation
     }
}
public class Client {
      void main() {
         Printer printer = new Printer();
         printer.print();
      }
}
```

Figure 1. Pseudo code for Delegation Pattern

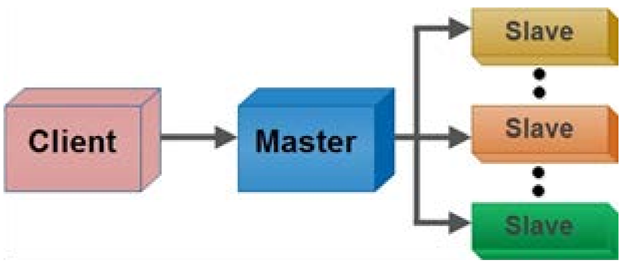

Figure 2. Delegation Pattern for HMA

The conceptual idea of the delegation pattern is transferred across in designing the HMA. The HA in the HMA is heavily loaded with various functionalities – based on our previous work [7, 10] the functionalities such as registration and authentication of patients, remotely monitoring of intrinsic health data from wireless sensors and storage of health data are considered to be generic and also requires high maintenance and resources. Therefore, the HA in the HMA is designed as shown in the Fig. 2, a master module (delegator) capable of registering and authenticating a patient and the slave modules (delegates) capable of remotely monitoring the patient's health data. The master module spawns (or delegates) as many slave modules for as many requests from the client (or health care providers). The spawned modules are terminated once the functionalities of that module are complete. In this way, the operational load of the master can be minimised owing to the delegation of functionalities to the slave modules. In the following section, we describe the architecture design of the proposed Assistive Patient monitoring cloud Platform for Active healthcare applications (AppA).

4. ARCHITECTURE DESIGN FOR APPA

In HMAs, the health data gathered by the BAWSN from the patient is delivered to the HA, which can be defined as a sophisticated application assisting the doctors/care staff to monitor the patients' health condition and consult with the patient 'on the fly', regardless of where they are located. Although the BAWSN achieves the critical function of gathering trustworthy health data from the patient, the HA provides the visualisation of the patients' progress to the doctor and can have many functionalities. Examples are maintaining the electronic medical records in the database, alerting the concerned clinicians about the condition of the patients, the ability to provide a common ground for the patients and the care staff to discuss their needs in detail and also in private; it can also have an intelligent algorithm to predict any forthcoming emergency situation. The general functionalities mentioned above are under the perspective of the user of this application. However, from the application developers' perspective, the specific implementation of these functionalities differs considerably depending on the health care requirements. For instance, the design of the electronic medical records differs considerably for patients who are suffering from lymphoma [1] and heart disease, and for those with other functionalities associated with an intelligent algorithm to predict any situation.

To comprehend the above claims in a pragmatic way, one of the objectives in our research was to build an HA to realize the complete end-to-end HMA. And also, in building an HA, it is imperative for our work to have a real-time test-bed to determine how the data generated and gathered from different BAWSNs is used at the other end of the HMA in a real-time environment. Hence, the AppA is built to maintain the electronic medical record and simultaneously monitor two types of patients for medical conditions - those who have undergone knee procedures and recovering at home and elderly residents at home at risk of falling. These two types of patients were selected because:

1. Knee surgery recovery patients typically have rehabilitation plans that involve sequences of exercises that should be stepped up as the knee recovers mobility. The challenge for most patients is to know when their knee has recovered sufficient mobility to step up to the next rehabilitation level. This is currently performed on advice from Orthopaedist and/or physiotherapist. However, gaining timely access to specialist is difficult and so rehabilitation is delayed, leading to problems being detected too late to correct.

2. Observing the physical condition of elderly people or patients in personal environments such as home, office, and restroom has special significance because they might be unassisted in these locations. The elderly have limited physical abilities and are more vulnerable to serious physical damages even with small accidents, e.g. fall. The falls are unpredictable and unavoidable. In case of a fall, early detection and prompt notification to emergency services is essential for quick recovery [21, 22].

The two medical conditions of the patient provides a real-time test-bed to explore the concurrent continuous monitoring using AppA as shown in the below Figure 3. Also, both the mentioned medical conditions have efficient algorithm defined in the literature [23, 24] for early detection and prompt notification with off-the-self wearable sensors.

[1] Lymphoma is a cancer in the lymphatic cells of the immune system

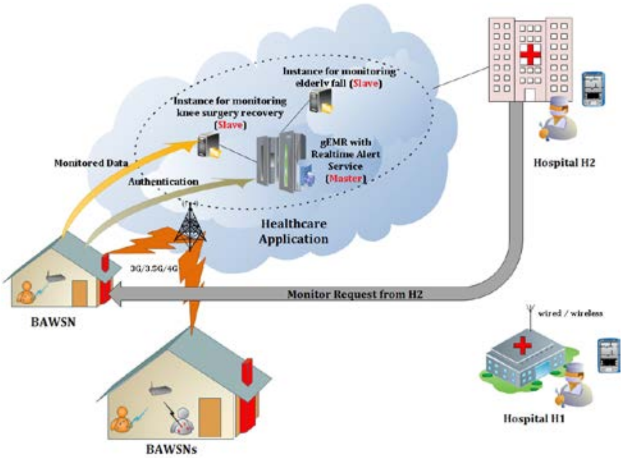

Figure 3. Architecture Design for AppA

A generic Electronic Medical Record (gEMR) is devised capable of storing necessary health data for patients. Using the gEMR a master instance is built. The master module can be used as a framework to monitor a variety of medical conditions of the patient. However, in this work two slave instances are spawned to monitor the knee surgery recovery and elderly fall conditions, as shown in Figure 3, in the cloud. Any HA would require health intrinsic data only through BAWSN. The BAWSN uses smartphones to transfer the health data from the body sensors to the slave instance. Therefore, an intelligent mobile application was developed to authenticate the patient as well as to collect the sensors data specifically for those medical conditions.

The architecture design depicts a real-time scenario that might happen when a patient requires continuous monitoring. As shown in Figure 3, the doctors/clinicians can request the patient to start the monitoring process. The patient with the wearable sensors attached to their body uses a smartphone application to start the monitor after authenticating their credentials with the master instance. In this way, the hospital ICT can eliminate the onus of authenticating the patient for monitoring. The master instance once authenticates the patient, spawns the slave instance with the required sensor data to be monitored based on the patient conditions and their requirements. The slave instance will be spawned if and only if there is no instance running for a particular hospital for that particular medical condition – this can be identified by using the hospital identification, for example H2 for Hospital2, as shown in the Figure 3. Please note the hospital identification will be entered by the patient along with their credentials during the authentication process. This authentication can be scaled up to large numbers of users.

Once spawned, the slave instance will be ready to receive data from the patient's smartphone. The functionalities of the slave instance are to receive the monitored data from the patient and to store the sensor data temporarily. The temporarily stored data can be transferred either to the master instance or directly to the hospital based on the service level agreement (SLA) between the cloud service provider and the hospitals. In addition, doctors from the hospital can login into the slave instance in order to visualize the patient health data and trigger any alert message based on the gravity of the health data.

One of the main aims of the AppA is to reduce the burden for the hospitals in deploying and maintaining the HA with their major functionalities such as patient authentication, continuous health data monitor, health data storage and real-time alert messages. The proposed AppA architecture achieves this by delegating the functionalities among the master and slave instances. The hospital responsibility is to send the request for the patient to start the monitoring and to have a required SLA with the cloud service provider. In the following section we detail the implementation of AppA.

5. IMPLEMENTATION DETAILS FOR APPA

Healthcare monitoring application is made up of emerging technologies for various heterogeneous components, therefore, the hardware and software used to build the HMA is also very much heterogeneous. In this section we present in-depth details of the hardware, software and communication infrastructure used in each of the components used in HMA. One of the major components of the HMA is BAWSN, which consists of wearable sensors to monitor the sensor data and the smartphone that acts as a gateway to transfer sensor data to the internet (or to HA). In this work, we used Shimmer sensors [25] and Samsung Tab [27] as wearable sensors and the gateway respectively.

We selected the following sensors Accelerometer, Electromyogram (EMG) and Electrocardiogram (ECG) because these variables are used in fall detection algorithms [21-23] and accelerometer data is useful for knee rehabilitation monitoring [26]. The Samsung Tab 2 was used as a gateway to the Australian Nectar Research Cloud (www.nectar.org.au). The following sub-sections present in-depth details of the hardware, software and communication infrastructure used in each of the components in HMA.

5.1 Shimmer Sensors

To monitor the elderly fall we selected the following sensors Accelerometer, Electromyogram (EMG) and Electrocardiogram (ECG) because these sensors does influence of the detection of elderly fall as given in [19, 22, 23]. Shimmer is a small sensor platform well suited for wearable applications. The Shimmer3 Bootstrap Loader (Shimmer3 BSL) application allows for the Shimmer unit to be programmed with the appropriate firmware solution when docked in the Shimmer Dock. Shimmer units arrive with BtStream and SDLog firmware. BtStream is a general purpose, fully configurable application to be used with the Shimmer platform. A Shimmer unit programmed with BtStream firmware will stream data via a Bluetooth (BT) connection to a PC, mobile or other Bluetooth-enabled device [25]. The SDLog Firmware is a firmware image which allows logging of data from a Shimmer to the on-board Secure Digital (SD) card. The firmware allows full user configuration of the Shimmer via a configuration file, stored on the SD card. The central element of the platform is the low-power MSP430F5437A microprocessor, which controls the operation of the device with 16Kbyte RAM and 256Kbyte Flash. The CPU has an integrated 16-channel 12bit analogue-to-digital converter (ADC) which is used to capture sensor data. Shimmer is powered by a rechargeable Lithium Polymer battery. The Shimmer battery is 3.7V, 450 mAh and contains a safety circuit board with over-current protection, which can trigger if a component short is created by a faulty peripheral or if components are bridged while the enclosure is open [25]. The detailed specification of the selected shimmer sensors are given below:

5.1.1 Accelerometer

By default the shipped Shimmer comes with 3-axis Low Noise Accelerometer and Wide Range Accelerometer array [25]. In addition, the capabilities such as 3-axis Gyroscopes (Angular Rate sensors) and Magnetic Sensor can be enable as well. However, the Shimmer are capable of sending only parameter at any given time. In other words, the Set in the sensor hardware is used to send either accelerometer, magnetometer or gyroscope sensor data. By default, the application will sample the 3-axis accelerometer at 51.2 Hz and send the data over a Bluetooth connection.

Figure 4. A patient wearing ECG in the chest, EMG in the forearm and a Accelerometer in the wrist.

5.1.2 Electromyogram (EMG)

The Shimmer EMG measures and records the electrical activity associated with skeletal muscle contractions and can be used to analyze and measure the biomechanics of human movement. The Shimmer EMG is non-invasive (surface EMG) and therefore the activity it measure is a representation of the activity of the whole muscle or group of muscles who electrical activity is detectable at the electrode site. The Shimmer ECG offers a wireless solution to a host of muscle, gait and posture disturbances in an easy to integrate and ergonomically valuable arrangement [25]. Each EMG board connects to three electrodes, namely, positive, negative and neutral. As shown in Fig. 4, the EMG is attached in the forearm of the patient.

5.1.3 Electrocardiogram (ECG)

The Shimmer ECG records the pathway of electrical impulses through the heart muscle, and can be recorded on resting and ambulatory subjects or during exercise to provide information on the heart's response to physical exertion [25]. In case of elderly fall detection ECG may give some indication to confirm atrial fibrillation and conduction defects where there is a prolonged PR interval, inferior ischaemia or bundle branch block. The three lead, two channels Shimmer ECG connects to the internal connector pin on the Shimmer main board, and is enclosed within the Shimmer unit, with the application to the skin via four conventional disposable electrodes as show in the Fig.4. In the below sub-section we detailed how the sensors data is captured using the smartphone monitoring application.

5.2 Gateway

A Samsung Galaxy Tab 2 [27] is used as the gateway in this work. The monitoring application is build using Android SDK [28]. The gateway connects with the selected Shimmer sensors using the Bluetooth. The doctor who wants to monitor the patient sends request from the hospital website, as shown the Fig. 5. The patient once receives the request from the hospital as shown in the Fig.6 starts the monitoring application installed in the gateway. The monitoring application prompts the user to enter the credentials along with hospital identification, as shown in the Fig.7, once authenticated by the master instance; the patient would be able to connect the sensors.

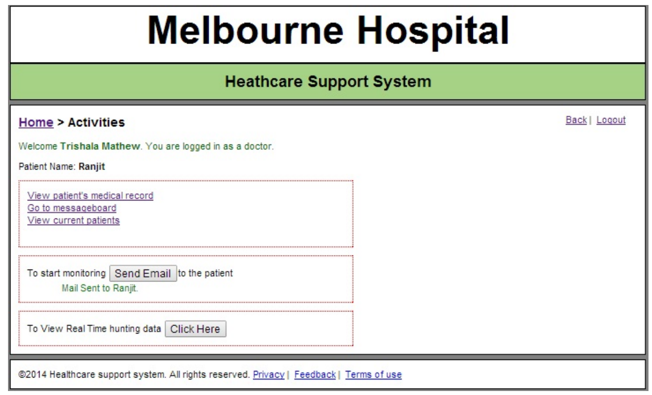

Figure 5. The doctor sends the request to the patient to start the monitoring process

Figure 6. Received request email from the doctor to start the monitoring process

(a) **(b)**

Figure 7. Login and Authenticated screenshot for the patient in the monitoring application.

Figure 8. Real-time streaming of sensor data

The sensors are connected using the connect button and can stream the data of all the three sensors simultaneously as shown in the Fig 8. The attached EMG in the patient converts input analog skeletal muscle contractions to a digital representation of this signal by assigning a value between 0-4095 to each sample and this will be output from the shimmer EMG module. Whereas, the attached ECG in the patients converts the input analogue signal, from four leads right arm (RA), left arm (LA), left leg (LL) and right leg (RL), in channel 1 and 2 to a digital representation of this signal by assigning a value between 0-4095 to each sample. The accelerometer gives the current position of the subject as x, y and z coordinates. The sensed data from the monitoring application is sent directly to the slave instance spawned by the master instance. Below sub-section details the cloud instances in AppA.

5.3 Cloud Instances

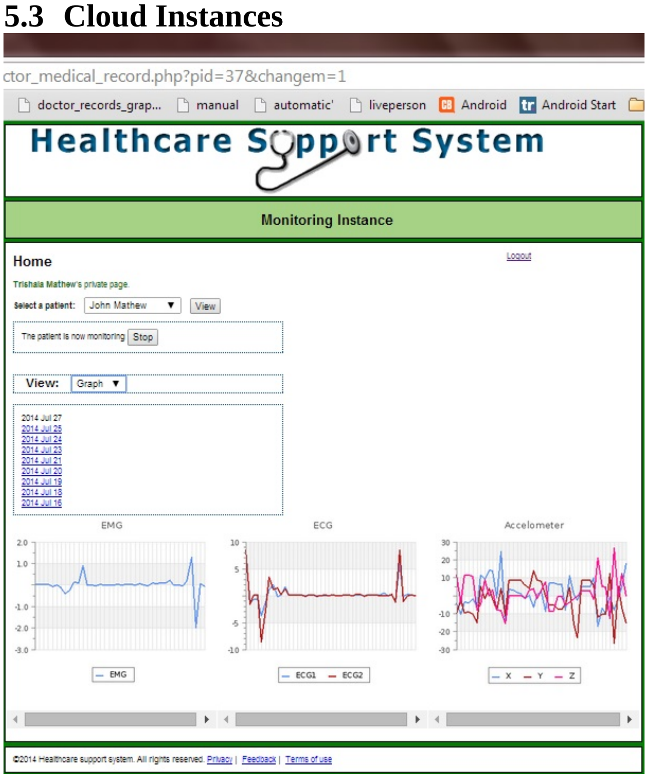

Figure 9. Slave instance shows the EMG, ECG and Accelerometer graph for a patient

The master and slave instances are shown in the Fig. 11 and Fig. 9 respectively. The slave instance not only facilitates the doctors to visualize the condition of the patient in real-time but also allows the doctor to send alert message manually to the patient in case of any abnormalities.

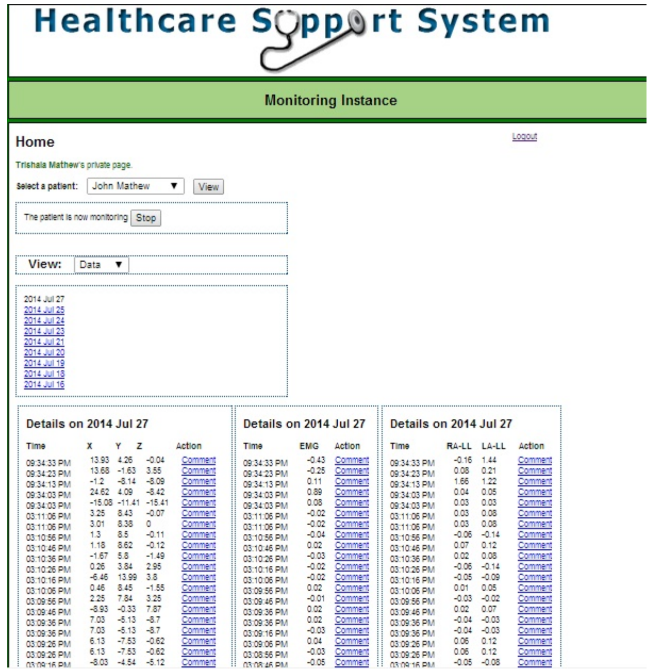

Figure 10. Slave instance shows the EMG, ECG and Accelerometer data for a patient

The patient were also given access to their data and can visualize them in real-time in the slave instance. In addition, they can collaborate with the doctors using the message board. In this way, the hospital minimize the overhead owing to maintenance of such heavily loaded functionalities of a HA. The slave instance are spawned dynamically based on the request from the hospital via patient by using Boto Applicaton programming interface (API) – a python interface to cloud services [29]. Please note, the slave instance is spawned one per hospital. If the patient from the same hospital makes another request – he/she will be added in the existing running instance.

Figure 11. Master Instance shows the active slave instances

The master instance have authentication only for the administrator and have the capacity to view, delete, stop and to take snapshot of the running slave instance. The master instance shown in the Fig.11 shows the two slave instances running for the hospital capable of monitoring two conditions simultaneously. Currently, the system has been implemented and running in real-time for planned field trials.

6. FORMULATION FOR COST EVALAUTION

The AppA is designed in such a way that it can leverage the cloud's pay-per-usage model. The master instance will spawn the slave instance for the hospital only if there is a request from the patient's from the hospital. If there is a slave instance already running for the hospital – any further request from the patient from the same hospital will be added in the existing instance. In this way, the Cloud service provider can charge the hospital only based on the number of hours the instance was running. The formulation below shows the cost estimation for AppA and in-house implementation of healthcare application for hospital H.

Let us consider a hospital H that has n number of patients to be monitored in real-time for a particulate disease. The total amount of time T for which the monitoring system was running in order to monitor the n patients is given by the following equation:

$$T = \sum_{i=1}^{n} T_i \quad (1)$$

Where, T_i is the time duration for monitoring a patient.

It is apparent that the time duration for monitoring n patients will have overlaps because several patients will be monitored at the same time for H. Therefore, the actual time T_A for which the monitoring system would be running is given by:

$$T_A = \sum_{i=1}^{n} T_i - \sum_{i=1}^{n} T_i \cap T_{i+1} \quad (2)$$

The total cost C for running the monitoring system for H using AppA model is given by the following equation.

$$C = T_A \times C_H + C_R \quad (3)$$

Where, C_H is expressed in dollars per hour ($/h) as we assume T_A is expressed in number of hours and C_R is the initial registration cost in master instance for H. In in-house model, the H has to incur the following cost for running the same monitoring services.

The capital cost for the hardware components C_C, maintenance cost for the hardware components C_M, which includes any on-going upgrades and minimal running cost for the hardware and support C_S. Please note the hospital H should have 24 x 7 services to replicate the master instance in AppA in order to service the requests from the patient. Therefore, the H should have proper infrastructure to host the server whose cost is represented by C_I. Based on the costs involved in running an in-house model, the total cost is given by the following equation:

$$C = C_C + C_M + C_S + C_I \quad (4)$$

It should be noted that in the above equation, the costs C_M and C_S are variable and depend on the amount of load the system receives in a given duration of time. Moreover, the hospital H has to incur the capital cost C_C again after certain duration owing to wear and tear of the hardware. Therefore, from equations (3) and (4), it is evident that for in-house model, the H has to incur overheads in running the monitoring system that has to be investigated further before deployment and also, the in-house model may not be scalable easily.

7. ARCHITECTURE IS APPROPRIATE FOR REMOTE MONITORING

The AppA approach addresses three of the challenges identified above: the selection of sensors, the design of an architecture that enables health care providers to sign up, start and terminate applications with minimal overhead, the alignment of technologies to viable economic models, and the provision of security.

The Master/Slave SOA approach enables each health care provider to spawn a slave instance configured to perform only the action relevant to that provider. The fall detection instance takes ECG, EMG and Accelerometer data and implements a fall detection algorithm whereas the Knee Rehabilitation instance solely uses Accelerometer data.

Security is maintained by having the provider request (by SMS) that the patient permit the initiation of a monitoring request by responding to an SMS. The health care provider is not required to set up or install any customized software but merely to subscribe to the HMA provider. Once commenced, the provider clinicians access the data stream through conventional web browsers.

The AppA approach is aligned with a scalable economic model because it provides the technology for start-up companies to provide HMA services. HMA suppliers can readily be imagined to offer a range of payment models including sourcing the full payment from the client on a per use basis or a combination of per-use and subscription plans from each health care provider and patient. The architecture envisages a plethora of HMA startups ensuring viable competition and also ensuring single vendors do not dominate the market place.

The AppA envisages that the data in the Master instance is not intended to be part of the patient's medical record per se. The data monitored by a HA is a subset drawn from the Master instance by a health care provider (or patient) so once downloaded, is ideally required to be integrated with other digital data related to the patient. Data integration has not been attempted in the current prototype but the design places responsibility for the integration in the realm of each HA. This is also intended to facilitate scalability in alignment with a viable economic model.

8. CONCLUSION

Cloud computing allows simple and easy maintenance of Information Technology (IT) infrastructure. In essence, cloud computing can be considered as a set of internet services provided by a third party who owns the IT infrastructure and offers its functionality over the Internet through various innovative business pricing models. In this proposed model, the healthcare monitoring application leverages the advantages of the cloud service to the extend that minimizes the operational and monetary cost overhead for the hospitals. Currently, the system has been implemented and running in real-time for planned field trials.

9. REFERENCES

[1] Samira H. Habib, Salima Akter, Soma Saha, Fahmida B. Mesbah, Mosaraf Hossain, Liaquat Ali, Cost-effectiveness analysis of medical intervention in patients with early detected of Diabetic Nephropathy in a tertiary care hospital in Bangladesh, *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, Volume 4, Issue 3, July–(Sep. 2010), Pages 123-127

[2] L. Blair, J. Harrison, D. Pahal, C. Clark, J. Allen, Sensing Fluid: The Use of Remote Monitoring Early Detection of Heart Failure to Avert Hospital Admission, *Canadian Journal of Cardiology*, Volume 29, Issue 10, Supplement, October 2013, Page S395,

[3] R. Agrawal and H.R. Arntz. 2006 ."Sudden cardiac death does not always happen without warning", *American Heart Association journal report*, pp. 93-98.

[4] Chen, W.L., Chen, J.H., Huang C.C., Kuo, C.D., Huang, C.I., and Lee, L.S. 2008. Heart rate variability measures as predictors of in-house mortality in ED patients with sepsis. *American Journal of Emergency Medicine* (2008) 26, 395–401.

[5] E. Ergezen, R.W. Hart, R. Philip, and R.M. Lec. 2008 "A high frequency thickness shear mode (TSM) sensor for detection of biomarkers for prostate cancer," Frequency Control Symposium, *IEEE International*, 19-21 May. 2008, pp. 341-345.

[6] Yo-Ping Huang, Chao-Ying Huang, Shen-Ing Liu, Hybrid intelligent methods for arrhythmia detection and geriatric depression diagnosis, *Applied Soft Computing*, Volume 14, Part A, (January 2014), 38-46, ISSN 1568-4946.

[7] V Balasubramanian and Hoang, D.B., "SOAP-based Assistive Care Loop using Wireless Sensor Networks", *Proceedings of the 1st IEEE International Symposium on IT in Medicine and Education*, IEEE Computer Society (Xiamen, China, 2008), 409–414.

[8] Akram Alomainy, Raffaele Di Bari, Qammer H. Abbasi and Yifan Chen. 2014. Chapter 1 - Introduction to Body Area and Wireless Sensor Networks, *In Co-Operative and Energy Efficient Body Area and Wireless Sensor Networks for Healthcare Applications*, Academic Press, Oxford, 1-4,

[9] Marie Chan, Daniel Estève, Jean-Yves Fourniols, Christophe Escriba, Eric Campo, Smart wearable systems: Current status and future challenges, Artificial Intelligence in Medicine, Volume 56, Issue 3, November 2012, Pages 137-156

[10] V Balasubramanian V, and Hoang, D.B. 2010. "Availability Measure Model for Assistive Care Loop Framework using Wireless Sensor Networks", *In Proceedings of the 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Brisbane, Australia, 281-286.

[11] K. V. Laerhoven, B. P. L. Lo, J. W. P. Ng, S. Thiemjarus, R. King, S. Kwan, H. W. Gellersen, M. Sloman, O. Wells, P. Needham, N. Peters, A. Darzi, C. Toumazou, and G.Z. Yang. 2004. Medical Healthcare Monitoring with Wearable and Implantable Sensors, *3rd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, Nottingham, England.

[12] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. 2004. CodeBlue: An ad hoc sensor network infrastructure for emergency medical care, *In International Workshop on Wearable and Implantable Body Sensor Networks*, April.

[13] P. Iso-Ketola, T. Karinsalo, and J. Vanhala. 2008. HipGuard: A wearable measurement system for patients recovering from a hip operation, *2nd International Conference on Pervasive Computing Technologies for Healthcare*, (Jan. 30 – Feb 1, 2008), 196-199.

[14] Pankaj Deep Kaur, Inderveer Chana, Cloud based intelligent system for delivering health care as a service, *Computer Methods and Programs in Biomedicine*, 113, 1 (January 2014), 346-359

[15] State of Victoria 2014. Royal Melbourenn Hospital Performance Report. http://performance.health.vic.gov.au/Home/Report.aspx?ReportKey=7&HospitalKey=79, accessed 12 Aug 2014

[16] eMotion LifeShirt from VivoMetrics, Available: http://www.vivometrics.com/lifeshirt/about-lifeshirt, accessed 8th April 2009.

[17] Heather Fraser, YangJin Kwon and Margaret Neuer. 2011. IBM Global Services. *The future of connected health devices Liberating the Information Seeker*

[18] Nabil Sultan, Making use of cloud computing for healthcare provision: Opportunities and challenges, *International Journal of Information Management*, 34, 2 (April 2014), 177-184

[19] A. T. van Halteren, R. G. A. Bults, K. E. Wac, D. Konstan- tas, I. A. Widya, N. T. Dokovski, G. T. Koprinkov, V. M. Jones, and R. Herzog, Mobile patient monitoring: The mobihealth system, *The Journal on Information Technology in Healthcare*, 2,5 (October 2004), 365–373.

[20] Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.

[21] Delegation pattern, http://en.wikipedia.org/wiki/Delegation_pattern, accessed 24 January 2014.

[22] Woon-Sung Baek; Dong-Min Kim; Bashir, F.;Jae-young Pyun. 2013. Real life applicable fall detection system based on wireless body area network, *IEEE Consumer Communications and Networking Conference* (Jan 11-14, 2013). 62-67

[23] Prevention of falls in the elderly, http://www.patient.co.uk/doctor/prevention-of-falls-in-the-elderly-pro, accessed 12 July 2014.

[24] Bisdorff AR, Bronstein AM, Wolsley C, Gresty MA, Davies A, Young A. 1999. EMG responses to free fall in elderly subjects and akinetic rigid patients. *Journal of Neurol Neurosurg Psychiatry*, 447–455.

[25] Shimmer sensor : http://www.shimmersensing.com/, accesssed 24 January 2014.

[26] Atallah, L.; Jones, G.G.; Ali, R.; Leong, J.J.H.; Lo, B.; Guang-Zhong Yang. 2011. Observing Recovery from Knee-Replacement Surgery by Using Wearable Sensors, *International Conference on Body Sensor Networks* (23-25 May 2011), 29-34.

[27] Samsung Galaxy Tab, http://www.samsung.com/global/microsite/galaxytab/10.1/index.htm, accessed 12 August 2014.

[28] Android SDK, http://developer.android.com/sdk/index.html, accessed 24 January 2014.

[29] Boto, https://boto.readthedocs.org/en/latest/, accessed April 8th 2014.