

University of Wollongong
Research Online

University of Wollongong in Dubai - Papers

University of Wollongong in Dubai

1-1-2014

Novel cloud and SOA-based framework for E-health monitoring using wireless biosensors

Abdelghani Benharref
University of Wollongong in Dubai

Mohamed Adel Serhani
United Arab Emirates University

Follow this and additional works at: <https://ro.uow.edu.au/dubaipapers>

Recommended Citation

Benharref, Abdelghani and Serhani, Mohamed Adel: Novel cloud and SOA-based framework for E-health monitoring using wireless biosensors 2014, 46-55.
<https://ro.uow.edu.au/dubaipapers/550>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Novel Cloud and SOA Based Framework for E-health Monitoring Using Wireless Biosensors

Abdelghani Benharref, University of Wollongong, Dubai, abdelgha@uow.edu.au
Mohamed Adel Serhani, United Arab Emirates University, UAE, serhanim@uaeu.ac.ae

Abstract— Various and independent studies are showing that an exponential increase of Chronic Diseases (CD) is exhausting governmental and private healthcare systems to an extent that some countries allocate half of their budget to healthcare systems. To benefit from the IT development, e-health monitoring and prevention approaches revealed to be among top promising solutions. In fact, well-implemented monitoring and prevention schemes have reported a decent reduction of CDs risk and have narrowed their effects, on both patients' health conditions and on government budget spent on healthcare. In this paper, we propose a framework to collect patients' data in real time, perform appropriate non-intrusive monitoring, and propose medical and/or life style engagements whenever needed and appropriate. The framework, which relies on Service Oriented Architecture (SOA) and the Cloud, allows a seamless integration of different technologies, applications, and services. It also integrates mobile technologies to smoothly collect and communicate vital data from a patient's wearable Biosensors while considering the mobile devices' limited capabilities and power drainage in addition to intermittent network disconnections. Then data is stored in the Cloud and made available via SOA to allow easy access by physicians, paramedics or any other authorized entity. A case study has been developed to evaluate usability of the framework and preliminary results that have been analyzed are showing very promising results.

Index Terms— e-Health, Body Sensor Network, Non-intrusive monitoring, chronic disease risk prevention.

I. INTRODUCTION

THERE is an alarming and growing number of people with chronic diseases in many countries all over the world. For example, according to the statistics from the UAE Ministry of health published in 2010 [1], over 25% of deaths in the country are caused by cardiovascular diseases. Risk factors that cause these diseases are multiple and include, but not limited to: cholesterol, diabetes, high blood pressure, physical inactivity, smoking, overweight and obesity.

To address the rising incidence of chronic diseases and their associated complications, well-designed monitoring and prevention approaches can help elucidating the main reasons of CDs and reducing the risk of their occurrence. These approaches, benefiting from the tremendous advances in healthcare services, are helping in revolving many of the earlier

mortal diseases into chronic diseases. This has led to an increasing demand for long-term continuous care, especially for elderly. Consequently, this has triggered growing costs for both the society and caretakers to an extent that, at the end, may lead to potential undersupply of health care services.

Following this trend, western healthcare authorities and industries are undergoing fundamental changes. They are shifting from hospital-centric services to a more ubiquitous and ambulatory system (with homecare, day care clinics, remote healthcare); whereby the treatment of chronic diseases is actively involving patients as well as their surrounding environment. The emergence of Internet-based e-Health services for instance, allows patients and healthcare professionals to have easy access to important health information anytime, anywhere, and in efficient ways.

In chronic diseases, continuous monitoring of subjects is a very important element in detecting diseases' symptoms as soon as they occur. Therefore, it can mitigate the impact and the consequences that these diseases may cause. Geo-localization tracking is also an essential mechanism in locating subjects while they are away mainly when emergency actions are to be taken (e.g. hospitalization).

This paper presents a novel electronic healthcare system, called Service Oriented and Cloud-Based e-health System (SOCBeS), for tracking, monitoring and prevention of chronic diseases and reducing their impact. The system relies on the established standardized concepts of Service Oriented Architecture (SOA) and cloud environment. On one hand, SOA has been proven to be an adequate solution for integrating heterogeneous systems and technologies, allowing application-to-application communication on the Internet, reducing cost of integration, and making data and services available to different stakeholders. On the other hand, using cloud infrastructures, platforms, and services maximizes data and system availability and reliability while reducing infrastructure costs. Healthcare organization adopting Cloud-based solution will be very beneficial in terms of massive patient's data storage, processing and management. It will allow these organizations to provide cost effective, faster, reliable, available and scalable services. SOCBeS will allow continuous (semi) automatic data gathering, on the fly monitoring, and taking proactive measures to identify risk factors and prevent the subject from unfortunate severe health

consequences. Being developed around the Cloud and SOA open technologies, the system can be easily integrated with other healthcare systems, which allows high interoperability and dynamic integration between heterogeneous systems, low cost, and high scalability in managing growing number and size of patient's data and services.

Monitoring patients in SOCBes serves three purposes: 1) early detection of CDs, 2) follow up patients who are confirmed to have CDs, and 3) prevent and coach patients that are prone to have some sort of CDs. All these cases involve the usage of sensing technologies along with communication infrastructure to sense and collect important physiopathology measures of a subject under observation. It addresses topics such as collecting sensory information, refining sensory information, and making intelligent decisions made available through the SOCBes services. Discrepancies between measured data and normal thresholds trigger immediate analysis and assessment of the subject's data by physicians to find out whether the subject has, or not, a chronic disease and if additional measures are to be taken.

Remaining sections of the paper are organized as follows: the next section presents a problem formulation of e-health systems for monitoring and tracking CDs. Section III presents the state of the art and discusses their strengths and limitations and positions the main innovative contributions of our proposed framework. Section IV identifies the fundamental requirements that a continuous automatic monitoring solution should satisfy. Section V presents and discusses SOCBes Architecture. In section VI a prototype and proof of concept is presented and results are discussed. Finally, section VII concludes the paper and highlights ongoing and future works.

II. PROBLEM FORMULATION

Chronic diseases monitoring process starts by initializing monitoring activities such as what to monitor, when to monitor, and threshold values. Initialization can be started by the patient, physician, nurse, or associated mobile application. If initialization is successful, the system enters a repetitive loop that will only terminate when the monitoring period expires or if the patient is hospitalized. Inside that loop, each measured value is compared to the normal thresholds range: if within range, the system waits for sometime before resuming monitoring. If out of range, there are three different actions: 1) if the discrepancy between measured value and thresholds is high to an extent that the subject needs hospitalization, the subject should be geo-localized, then the position will be sent to paramedics so that ambulances can reach him/her as quickly as possible. 2) If the discrepancy is high but the subject can get herself/himself to a physician, they will be instructed and guided to do so. 3) In the case where the discrepancy is low and can be levitated just by following an appropriate lifestyle, customized lifestyle's advices are sent to the subject. In the three cases, monitoring activities keep going until the end of desired monitoring period or hospitalization of the patient.

In the following subsections, we present a formulation of the health-monitoring problem based on state of the art mobile technologies including sensors, mobile applications, SOA, and Cloud. A sensor usually measures a single health parameter. If

a sophisticated sensor can measure two (or more) different parameters, it will be considered as two (or more) separate sensors even though they are contained within the same device. Each sensor is represented by a couple $\langle S_i, HP_i \rangle$, where S_i represents the sensor's ID and HP_i defines the health parameter that the sensor can measure. A health parameter is defined by a quadruple $\langle HP_i, min, n, max \rangle$ where HP_i is the parameter ID, min is the minimum value tolerated, n is the normal value as measured on a disease-free person, and max is the maximum tolerated value. For parameters without min (or max) value, the quadruple will be noted $\langle HP_i, \infty, n, max \rangle$ (or $\langle HP_i, min, n, \infty \rangle$).

From a patient's point of view, a Patient Under Monitoring (PUM) is represented by a tuple $\langle V_S(S_j, HP_j), M_i, D_i \rangle$ where V_S is a vector of $|V_S|$ sensors measuring $|V_S|$ parameters, where $|V_S|$ is the cardinality of V_S , M_i is the application connected to those sensors, and D_i is the assigned doctor (physician).

From a hospital point of view, a hospital has n physicians and nurses handling m patients under monitoring $\{V_h(PUM_i) / 0 < i \leq m\}$. Each physician or nurse p_j is responsible of V_{pj} patients $\{V_{pj}(PUM_i) / 0 < i < k \text{ and } V_{pj} \subset V_p \text{ and } V_p = U(V_{pj}(1 \leq j \leq m))\}$. If we denote Health Measurements of all parameters of a subject i during a monitoring period from time ts to time te by $HM_i^{ts \rightarrow te}$, we then need to find the *soonest* and *smallest* HM_i that detects risks R (chronic diseases or minimizes their impacts on subject i). HM_i has to satisfy the following property:
If $HM_i^{ts \rightarrow te}$ detects a chronic disease or a risk with patient i , then:

$$(1) \quad \nexists HM_i^{ts \rightarrow tb} / HM_i^{ts \rightarrow tb} \rightarrow R \text{ and } tb < te$$

Where $HM_j^{ts \rightarrow tb} \rightarrow R$ means $HM_j^{ts \rightarrow tb}$ detects R

The above property should be read as: there is no $tb < te$ such that $HM_i^{ts \rightarrow tb} \rightarrow R$. That is, there is no sequence of measurements performed between ts and tb (i.e. before te) inclusively that detects the chronic disease or its risks for patient i . To detect R , measurements have to be taken from ts until te , beyond tb .

For all m subjects (i.e. for a hospital), we need to find HM that is defined as:

$$(2) \quad HM = \{U\{HM_i\}, 0 < i < m\}$$

The process of monitoring and tracking m patients with chronic diseases can be formally specified as in Figure 1.

III. RELATED WORK

Nowadays, several e-health centers emerged and many initiatives are launched in USA, Canada, UK, EU as well as in many developing countries in order to improve healthcare services and optimize medical resources. At the core of all these initiatives is the investment in modern ICT infrastructures to connect hospitals, clinics and healthcare organizations, patients, and professionals so that the exchange of medical data can be possible, accurate, and reliable. This has also been made possible due to tremendous development in wearable body sensors capable of reading various health parameters directly from patients' body with acceptable levels of accuracy [2].

```

Input: startTime, endTime : Date ; // monitoring start and end times
Input: PUM: <Vs(Sj,HPj), M > ; // A Patient Under Monitoring
Input: Vs = {PUMi, 1 ≤ i ≤ m} ; // Patients monitored by a hospital
Input: waitTime = {wTimei, 1 ≤ i ≤ m} ; // Waiting times during monitoring
Data: reading : real ; // reading value from a sensor
Data: healthCondition: string ; // health condition of a patient

// the following procedure monitors one patient, given as an argument
procedure monitor(Patient PUMi, int waitTimei)
begin
  initialize a. startTime;
  while (actualTime ≤ endTime) do
    for j ← 1 to Vs do
      reading ← reading from sensor Sj;
      log(reading);
      healthCondition ← discrepancy between reading and min/normal/max
      values of HPj;
      switch (healthCondition) do
        case Critical
          get geo location from M;
          send paramedics;
          return;
        case Serious
          Ask patient to visit a physician ASAP;
          break;
        case Warning
          Send appropriate lifestyle to patient via M;
          break;
        Default // everything is fine, wait before next reading
          wait(waitTimei);
  /* The main method below starts monitoring of all patients, parbegin
  initiates monitoring of m patients concurrently */
void main()
for i ← 1 to m do
  parbegin(monitor(PUMi, waitTimei));

```

Figure 1. Chronic Diseases Tracking and Monitoring Process

The increasing demand for e-Health services has led to many research efforts [3]. Few, yet immature, architectures have been proposed for e-Health or other healthcare related systems' purposes. Some of them are used in special areas, such as trauma [4], cardiology [5], and neurosurgery, pathology treatment. Some are used with special purposes, such as emergency and patient monitoring [6]. With the advances in wireless and mobile technologies, there are also few wireless-based e-Health systems (i.e. m-Health system) developed in [7]. The common feature of these systems is that they only provide limited or special services to end-users. They are not transparent to involved parties, difficult to integrate with existing healthcare systems, and do not cater varying monitoring parameters.

Related works can be classified into two categories: solutions and architectures that emerged from healthcare industry and those that have been developed within academia. In the rest of this section, we describe the main characteristics of these solutions, conduct a thorough comparison based on relevant criteria of importance to e-health architectures, and finally we highlight how a comprehensive e-health based on WBSN should be designed and implemented while highlighting the main challenges that are involved in its design and development.

Under the umbrella of industrial solutions, many governmental initiatives led to the development of large-scale e-health based systems to help mainly elderly and people with chronic disease to be continuously monitored from home. Most of these solutions however, were developed to enable vital signs measurements, processing and real time decision support

(e.g. increase a medication dose) [15-18].

From academia, several research works and initiatives have studied the issue and challenges of building somehow comprehensive e-health solutions. These solutions differ in many aspects (1) the solution used to tackle integration issues, (2) the heterogeneity of different systems, and (3) the type of middleware, framework or architecture used to build the integrated e-health system. They mostly used WBSN to collect real time patient's vital signs, generate health status feedbacks, and send alert in case of severe health threatening situation. Indifferently, these solutions consider some key challenges related for instance, to the capabilities of sensing platform being used, network and communication interruption, power consumption, preservation of biosensors, mobility management, data processing and management.

For research solutions that tackle the integration and the heterogeneity issue in e-health systems, we classify solutions based on the implementation standard on which they rely for instance CORBA [19, 20], SOA [21, 22], and Health Level Seven (HL7) [23] used as the underlying integration infrastructure. CORBA [24] has been known as not flexible solution to allow the integration of non-CORBA systems and HL7 [23] also is renowned as message exchange standard that does not provide a complete integration solution at lower layers of different sub-systems. SOA has been proven to be an appropriate solution that allows varying organizations leverage their shared services to automate multiple business processes and reinforce overall interoperability [25].

The authors in [26] proposed a distributed framework of a Web-based telemedicine system which addresses two types of servers namely Web servers and data servers. This framework is based on CORBA technology and a distributed database fragmented on different sites. Unfortunately, the system is not flexible to allow the integration of non-CORBA systems; therefore, it requires an intermediary middleware to handle the heterogeneity between different health systems as well as a huge development effort to adapt the system to the integrated system requirements.

In [27], the authors proposed a multi-layer SOA-based e-Health services architecture. It consists of six main components responsible for defining interactions among different layers. Unfortunately, the system has not been developed; it states an architectural design without detailing the implementation and its complex challenges. Kart, F. et al. in [22] described a distributed e-healthcare system that uses SOA as a mean of designing, implementing, and managing healthcare services. The system includes a clinic module, a pharmacy module, and patient's interfaces, which are implemented as Web services. Various devices can interact with these modules, including desktop and server computers, Personal Digital Assistants, smart phones, and even electronic medical devices, such as blood pressure monitors.

The authors in [28] investigate how healthcare entities, using SOA, can leverage their common services to automate multiple business processes and strengthen the complete interoperability. The authors in [22] designed and developed a SOA-based platform for home-care delivery to patient with

chronic diseases. This work shares some of the goals with our project with regards to monitoring patients with chronic diseases. However, it differs in many aspect related to data dissemination though Cloud environment, power conservation, performance consideration, and value added services including prevention service (e.g. lifestyle adjustment, diet program).

Both categories of e-health based WBSN architectures that emerged from industry and academia described above exhibit some combinations of the following weaknesses:

- Less transparent to different healthcare stakeholders.
- Difficult to configure, deploy, and integrate with other healthcare systems.
- Lack clear separation between the data and the business logic implementation.
- Less flexible to handle the heterogeneity of different healthcare systems.
- Demand huge development effort to adapt the system to the integrated system requirements.
- Unable to fully integrate different biosensors for different vital signs and from different vendors in one solution/architecture.
- Resilient to mobile technologies integration because of the limited processing capabilities of the mobile devices (make use of REST WS).
- Slightly make use of varying Cloud services (SaaS, IaaS, PaaS) to benefit from storage, and processing capabilities of the Cloud infrastructure and services.

A reliable and self-configurable architecture for efficient health monitoring architecture based on WBSN must address the above challenges. Therefore, it should support data collection, perform non-invasive monitoring, and propose medical and/or lifestyle adjustment whenever needed. The following are some of the fundamental benefits our proposed architecture is developed to offer, which are lacking in related works presented above:

- Allow non-invasive sensing that does not burden the subject under monitoring and makes him/her feel that nothing is taking place at his/her private environment.
- Easy to integrate with any other healthcare system and easy to augment with new services as it is fully based on Web services.
- Follow the logic of decomposition-based layers to separate between core business logic, the detail of the implementation, the underlying sensing platform, and the upper application layer services.
- Rely on the usage of Cloud services while guarantying high Quality of Service (QoS), data privacy and Security, and management.
- Integrate different sensing technologies by means of creating interfaces that allow the adoption of different protocols including for instance Bluetooth, 3G, and Wi-Fi.
- Use mobile device for sensing and data gathering while optimizing (minimizing) device power consumption and the limited processing capabilities by using RESTful Web services for the business logic and the Cloud for data storage and processing.

- Allow data synchronization among different storage repositories and Cloud data centers.
- Provide some value added services that include for example automated preventive plan, monitoring services, and QoS guarantee.

IV. USER AND ARCHITECTURAL REQUIREMENTS

Collecting and sharing private health information have many sensitive requirements. These requirements can be classified under two interrelated categories: 1) user requirements and 2) architectural requirements.

A. User requirements

From a user point of view, collecting and sharing health information should satisfy at least the following properties:

- Non-intrusive sensors: traditional gathering of health information requires very intrusive approaches that are mainly centered on samples (e.g. blood sample and urine sample) that are taken directly from the subject's body. However, some users might highly prefer to use non-intrusive ways.
- Context aware: the collected health data as well as the frequency at which these data are collected should be clever enough to consider various contexts (e.g. type of vital monitored signs, the situation of the subject to be monitored).
- Data Accuracy: as all kind of measurements includes error margins, health/body information measurement should be very accurate as a slightly small variation might lead to fatal health issues. Readings' errors margins have to be controlled.
- Availability: the platform is used by different actors who have different access rights and requirements (time, location) thus should be available to all users anytime and anywhere.
- Performance: the platform should offer satisfying performance to end-users. Response time in all operations should be as low as possible.
- Data Privacy: health data privacy of all actors especially users under monitoring should be protected. The platform should provide proven mechanisms to protect privacy.
- Confidentiality: any information collected or transiting by the platform should be confidential.

B. Architectural requirements

User requirements discussed in previous section have straightforward implications on architectural requirements. However, the later includes in addition, extra requirements related to the underlying architecture that are completely transparent to end-users. Architectural requirements include:

- Sensor-based: non-intrusive, transparent measurement and congregation of health information requires efficient use of state-of-the-art body sensors capable of reading appropriate parameters at high precision (e.g. temperature, blood sugar, and blood pressure).
- Wireless connectivity: in order to be available anywhere and to all users in addition to be non-intrusive, the platform should consist of wireless components as much as possible mainly those components that interact directly with users under monitoring.

- Cloud-based: in a completely wireless environment where subjects under monitoring are located all around; availability is an important issue, distributing data and business logic over multiple sites (data centers) is required.
- Responsiveness: the platform should be designed in such a way that response times are always very low.
- Security: health data should be protected all over the path from measurement to storing and sharing.
- Support for heterogeneity/platform-independent: different users are using different health systems and should be able to integrate seamlessly the platform within their various systems.
- Scalability: the architecture should be scalable to support an increasing and high number of actors and applications.
- Robustness: ability to work in extreme conditions and to quickly and safely recovers in case of failure.

V. THE “SOCBeS” FRAMEWORK

A. Description

In this section, we describe the key constituents of our framework along with a summary of communications and interactions among them to accomplish the main functionalities of the framework namely: e-health data sensing, filtering, storage, monitoring, and visualization.

The framework is composed of a set of modules each of which exhibits a set of functionalities and services. Figure 2 illustrates the framework structure and the main components of each module. These modules include: Non-invasive sensing module,

SOA and integration sub-module, the Cloud environment module, and e-health services and applications module. The value added feature of our proposed framework is the full integration of diversity of technologies and solutions at all levels from low level infrastructure to high level services and applications. This will allow smart non-intrusive monitoring of subjects’ vital signs while maintaining a high level of performance relying on Cloud value-added services, efficient data gathering and management based on sophisticated sensors and gateways, smooth integration and interoperability between different technologies using SOA.

The rest of this section describes each of these modules along with their main components’ roles and responsibilities within the framework.

Non-Invasive Sensing Module: this module is divided into two layers:

(1) Non-intrusive sensing devices (e.g. sensors, mobile devices): these are used to sense one or more health parameters (e.g. blood pressure, blood sugar, body temperature, and oxygen saturation) without disturbing the patient’s normal life as such not being aware that sensing activities is taking place. It also includes gateways that serve in collecting data from sensors, process (e.g. filter), and store these data on the Cloud data centers. Many sensing technologies are available nowadays and used to retrieve vital information, and then relay these to the nearest device capable (e.g. gateways) to process these health data. These intermediate relays generally provide an interface to access and retrieve these data.



Figure 2. SOA & Cloud enabled Architecture for Smart Non-Invasive Health Monitoring

(2) Communication protocols: these include a wide range of protocols that are used to transport monitored health data from sensing devices into the storage module namely the Cloud data center or the monitoring repositories that might reside at the SOA and integration module.

Cloud Environment Module: this module provides the underlying infrastructure and platform that host e-health data and applications. This includes connectivity management, device management, e-health data processing, runtime engine, and other Cloud based services that support monitoring activities and their related processes. In fact, Cloud

environment offers the three common categories of services that are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The IaaS incorporates the physical infrastructure including for instance servers, networks, databases, and block storage. However, the PaaS incorporates other type of entities such as operating systems, DBMS, runtime engines, and many other tools. The SaaS represents the pillar of our framework as it integrates different components that are used to allow a full integration of heterogeneous healthcare systems with the monitoring framework. Moreover, it provides the necessary interfaces to enable an easy access and manipulation of health data. The Enterprise Service Bus (ESB) is the main component of this sub-module that allows interoperation and exchange of data among different sub-systems.

In addition, to the above components, the SaaS is linked with local monitoring repositories in which sensed data are stored and managed. These repositories are an intermediary storage environment that implement synchronization mechanisms to synchronize data stored with those stored on Cloud data center when required (e.g. at each data collection or update).

E-health Services and Applications Module: it consists of a set of applications and services that are used for the collection, processing, and visualization of monitored health data. Type of services might include report generation, pattern extraction, data mining tools, data visualization applications, hospital information systems, mobile applications, and social health.

In addition, to the above framework's modules, two silos in the framework of Figure 2 characterize some key important aspects of our proposed framework. These include: Quality of Service-Service Level Agreement (QoS-SLA) and management, and the Security/Privacy. These knowledge areas are involved at all levels/modules of the framework involving all the four previously described framework modules. QoS-SLA and management area involves (1) management processes from the lowest-level modules of the framework (e.g. device management) to the highest-level modules of the framework (e.g. visualization services), and (2) service level agreement to guarantee a high level of QoS and detect any violation of these QoS parameters. Security and privacy area exhibits a very important aspect of the framework that should be maintained at every single entity of the proposed solution. Protecting data gathering, storage, communication, and manipulation by applications and services is very crucial as it deals with private, very sensitive patients' health data.

B. Monitoring processes and main involved entities

In this section, we illustrate the main processes and operations of smart health monitoring. The monitoring lifecycle involves a set of roles and components that include the WBSN built on the monitored subject (e.g. a patient), a data collection and processing that includes the data collection and processing that involves an IaaS module and/or monitoring repositories, SaaS module, and varying end user applications. The following are the operations executed within the monitoring process.

- The Wireless Body Sensor Network (WBSN) is used to integrate different biosensors that sense couple of vital signs

including Blood pressure, Blood Sugar, Heart Rate, and Temperature.

- Different communication protocols can be used to transmit the collected data to different storage and processing locations. These protocols are based on the sensors used and the protocols they support (e.g. Bluetooth, Wi-Fi, 3G, and Infrared).

- Data can be transported and processed using a Cloud IaaS in which abundant storage and processing services and resources are available. It can eventually be sent to monitoring repositories (e.g. Databases). Data synchronization might be needed to keep track of all updates in both locations.

Couples of Web services constitute the SaaS to integrate couple of interfaces that are developed to hide the implementation detail of the monitoring. These Web services implement business logic to deal with data collection, processing, and clean up (e.g. filters).

End user applications can be a mobile application, a web portal, a Hospital Information System (HIS), a surveillance center, social networks, or any application that can make use of monitored data for different purposes, such as decision-making, mining, and pattern recognition. These applications invoke related Web services to get customized data, generate reports and statistics, visualize these data, alert physicians in emergency and severe situations, share statistics and socialize to distribute information about disease awareness and prevention, and eventually use data mining schemes for better health diseases management and intervention.

C. Adaptive Monitoring Scheme

In this paper, health monitoring refers to the process of automatically and semi-automatically collecting patients' appropriate health information. The information to collect and frequency at which it is collected depend on many issues such as the purpose of monitoring, the patient's health condition, and the desired collection pattern. Collected data have to be communicated on the fly to the backend system. However, if there is no possible connection, data have to be saved on the mobile application and then communicated when a connection becomes available. In some serious circumstances, the system can explicitly ask the patient to re-initiate the connection. Data collection patterns can follow one or a combination of the following patterns:

1) **Continuous:** this type of monitoring offers the highest level of granularity of collected health data. It consists of a set of sensors continuously gathering relevant data. This heavy monitoring pattern is usually used in intensive monitoring situations as, for instance, severe disease monitoring.

2) **On-demand:** in this pattern, monitoring is initiated following a request from a controller. A controller is any entity authorized to trigger monitoring and includes: patient, doctor on duty, and assigned nurse. This pattern is usually considered when available readings are insufficient or unexpected and more detailed results are needed.

3) **Periodic/regular:** in this case, monitoring is performed at regular patterns or intervals. This is usually an automated and preplanned activity that starts and ends at predefined dates and times. These are defined by the controller and can be changed at any time. Periodic monitoring is used in normal monitoring of

patients with known stable conditions and non-health-threatening situations.

VI. IMPLEMENTATION AND EXPERIMENTATION RESULTS

To prove the applicability of our framework in real situations, we have implemented its main components, and conducted some monitoring scenarios. In this set of experimentations, we studied the monitoring of two diabetic patients (e.g. blood sugar). These patients are equipped with sensors and mobile applications for one patient, the application is developed and deployed on an iPhone mobile and, for the second patient, the application is developed and deployed on an Android-based mobile. Both users are being tracked by a physician using a portal-based application.

A. Experimentation Test bed

The test bed in this phase of experimentation is depicted in Figure 3 and consists of the following components:

- **Server #1:** Core 2 Duo desktop; this server is running the SOA platform in addition to database. Due to its simplicity, efficiency, and open source, we are using Glassfish [29] as the main Web Service container. For similar reasons, we are using MySQL [30] as our database system.
- **Server #2:** high-end third generation i7-based desktop; this server is mainly running the portal platform and container. This consists of two heavily tested and widely used containers: Tomcat [31] and Liferay [32]. Moreover, this server is running apache web server [33] and twiki wiki server [34]. The web site and wiki site offer plenty of information required to use the framework, extend the framework, or integrate new components into the framework.
- **Microsoft Azure slice:** this is cloud storage where health data are synchronized with the local data on the MySQL database. Web Services can read and write data to and from the MySQL database or from the Azure
- **Physician view:** this consists of a MacBook Pro with Internet browser (Safari) connected to the portal platform and able to display various information and portlets. In this set of experimentation, and through the portal, the physician is able to monitor patients' information and to send them appropriate advices or direct them to appropriate sources of medication or lifestyle activities.
- **iPhone 4:** this iPhone is connected to the iBGStar sensor (Blood Sugar based sensor) and running the iBGStar application [35]. Patients use the sensor's strips to get their blood sugar readings. The blood sugar readings are then entered into our mobile applications.
- **iPhone 4S:** it is running the iOS version of the monitoring application. A more detailed description of the mobile application is provided in section VI.A.1).
- **Samsung Note (Client Application):** this Android-based smart phone is running the android version of the monitoring application.

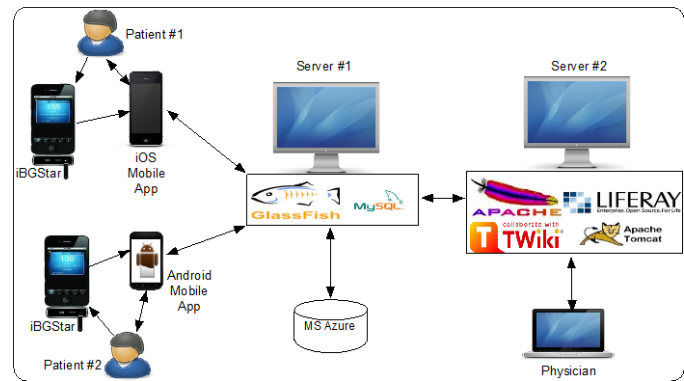


Figure 3. Test bed configuration

To facilitate information exchange and messages parsing, communications between mobile applications and servers are all encoded in JavaScript Object Notation (JSON) [36]. JSON is a lightweight encoding scheme that is now widely preferred over XML for message exchange on the Internet. Moreover, communication between mobile applications and the Web services platform (i.e. server #1) are based on Representational State Transfer (REST). REST [37] has been preferred over Simple Object Application Protocol (SOAP) [38] for communication due to its simplicity and lower overhead compared to SOAP [39].

1) Mobile app overview

This application exhibits 4 graphical interfaces' views two of whom are depicted in Figure 4: the first screenshot is from the iOS application while the second is from the Android application. Both applications are developed while keeping in mind the User eXperience (UX) on iOS and Android so that users of both platforms benefit from the ease of use and look and feel that are used too.



Figure 4. iOS and Android monitoring screen shots

We have experienced the solution with couple of users under monitoring. These experiences revealed that all monitored subjects were not disturbed while being monitored; sometime they are not even aware that monitoring activities are taking place as long as they are wearing the sensors. However, there was a need for a quick introduction and training at the first use of the sensing kit and the mobile application (how to wear the sensing kit (belt, sensors), how to initiate/stop monitoring, how

to navigate through the mobile application to read the monitoring results, advices, and graphs).

a) My Data: this view shows series of three types of blood sugar readings: 1) readings taken early morning while fasting, 2) readings taken two hours after meals, and 3) readings taken before going to bed, late evenings. Readings show up in the view as soon as they are available. Readings' sources and delay between readings can be configured in the settings view.

b) Chart: this view dynamically plots a chart of the three types of readings' values displayed in the My Data view. Whenever a new reading is available, it is then plotted. Three thresholds of relevance to blood sugar are represented by horizontal grids: 100 mg/dl, 200 mg/dl, and 300 mg/dl.

c) Advices: this view shows the latest advices based on recent readings. The physician suggests new advices, which are then pushed through the backend servers to this mobile application view. The patient can also explicitly request latest advices by pressing the icon on top-right of the view.

d) Settings: this view is used to customize data collection and synchronization between different data sources including the cloud. It consists of a key to start/stop readings and has two sub-views: 1) for reading configuration and 2) for data synchronization. Readings can come from three different sources: sensor to acquire new readings, backend database, or local repository to review previous readings. Although time between two consecutive readings is solely relevant when getting readings from sensors, it can also be used to illustrate the evolution of previous readings. If the patient would like to see the whole plot at once, this value should be then set to zero.

In the data synchronization sub view, the user can instruct the mobile application to send new readings as soon as they are available (real time option). The user can also overwrite the remote readings database with data from the mobile app (sync to backend option), can overwrite local data by readings stored in the backend (sync from backend option), and can also merge local data with remote data. In the later case, resulting remote data is coherent with local data with no duplicates.

2) Web Services

End user mobile apps and the portal make use of different services offered by the monitoring framework using a set of Web services deployed on server #1. Table 1 summarizes the deployed Web Services' operations. The input and output parameter "Reading" of some of these operations is a new user-defined type that consists of a value of type float, a string that describes the type of readings, and a date that indicates when (date and time) the reading was taken. The notation $\{Readings\}$ denotes a *JSON* set of readings.

Table 1. Some Web Services Operations

Operation	Description
<i>Reading</i> <i>getFirstReading()</i>	This operation returns the first reading of the patient.
<i>Reading</i> <i>getNextReading()</i>	This operation is usually invoked after the previous operation and returns the next reading based on the last reading.
<i>Reading</i> <i>getLastReading()</i>	It returns the last reading of a patient.
<i>Reading</i> <i>getPreviousReading()</i>	It returns the previous reading based on the last retrieved reading.
<i>JSON{Reading}</i>	It returns <i>JSON</i> encoding of readings, from

<i>getReadings(int x, int y)</i>	index <i>x</i> to index <i>y</i>
$\{Readings\}$ <i>getReadings(Date d1, Date d2)</i>	It returns a <i>JSON</i> encoding of readings, between date <i>d1</i> and date <i>d2</i>
<i>bool insertReadings({Readings})</i>	This operation accepts a <i>JSON</i> encoding of readings from the mobile application and inserts its elements as the last readings into the database.

B. Monitoring overhead

Although components of the monitoring framework can be located anywhere on the Internet, in our experimentations, they are grouped into four groups; each group is located in a single Local Area Network (LAN). The first group consists of the mobile infrastructure and includes sensors and mobile applications. The second group consists of server #1 third group consists of server #2, and the MS Azure cloud is group four. All groups are connected as follows: groups #2, #3, and #4 are usually connected through wired networks; group #1 is connected to group #2 through wireless technology (WiFi, 3G, Bluetooth, etc). The network overhead due to communication between groups #2, 3, and #4 can be ignored since exchanged messages are usually small and network bandwidth in and between nowadays LANs is very high. However, the network overhead introduced by communication between groups 1 and 2 has to be considered. We quantitatively and qualitatively discuss the network overhead introduced by continuous monitoring in the following subsections.

When the mobile application is configured to send new readings as soon as they are available to the backend server (option Real time enabled via My Settings view), the number of messages sent everyday is the same as the number of daily readings. Our experimentation showed that the size of an HTTP message carrying a REST message containing one reading encoded in *JSON* is around 560 Bytes. Even over a slow network (e.g. 3G), this amount of data does not represent a substantial overhead, as most diabetic patients are likely to take few daily readings (on average, three/day).

When synchronizing data to and from the backend database (last three options in the My Settings view), the overhead is quite high. In fact, a patient might have readings for months or even years and synchronizing back and forth will induce a considerable amount of data. Figure 5 shows the volume of data exchanged between the mobile application and Web Services to synchronize a various numbers of readings. This figure shows that the load is linearly proportional to the number of readings. This is due to the HTTP/REST overhead, which is almost the same for one or many readings. The monitoring overhead, at the HTTP level, for any number of readings can be then estimated using the following formulae:

$$Overhead_n = Overhead_{req} + n * (Overhead_1 - Overhead_{req})$$

Where:

- $Overhead_n$: the overhead to synchronize n readings
- $Overhead_{req}$: the overhead generated by a single synchronization request (excluding the response)
- n : the number of readings to synchronize
- $Overhead_1$: the overhead to synchronize one reading (including request and response)

Our experimentations showed that $Overhead_{req}$ is around 320 Bytes and $Overhead_r$ is around 880 Bytes. So, the monitoring network overhead equation above can be rewritten in Bytes as:

$$Overhead_n = 320 + n * 560$$

Although the overhead equation is linear, the network overhead might look very high for an excessive number of readings. However, we do not expect users to synchronize data very frequently; and even when they do, the mobile application and backend Web Services will only send new data, the data that have not been sent synchronized previously.

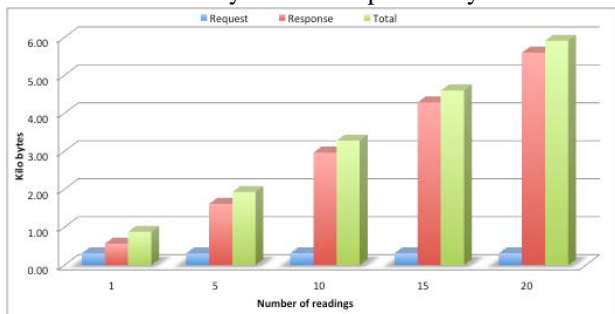


Figure 5. Monitoring network overhead.

C. Discussion

In this set of experimentations, we have executed many scenarios in which different readings for a couple of patients being monitored by a physician are involved. Normal and abnormal blood sugar readings have been used to check the responsiveness of the framework and the prompt correctness of interactions between its components. For the moment, we are basing decisions on simple comparisons between reading values and standard thresholds for different parameters. Our experimentation showed that as soon as an abnormal reading is observed, the physician is notified on the fly, through the portal, and requested to take appropriate actions. When the physician proposes new advices (e.g. lifestyles or requests patient to perform activities) based on last readings or ongoing treatment, the mobile application instantly prompts the patient to check physician's instructions, which are going to be presented in the My Advices view.

From a monitoring overhead point of view, our results reported that the network overhead generated by monitoring activities is somehow low even if data on mobile applications and the backend system are kept coherent. In fact, when synchronizing data, only new and small pieces of data are sent back and forth except in the exceptional case where the mobile application is willing to wipe out all its data and starts fresh with data stored in the backend database or the Cloud. This might incur some charges when on 3G/4GLTE network but does not have that much impact when connected through a Wi-Fi connection.

From a security perspective, data accuracy and privacy are key issues in health environment and without these, no solution will get any real success. For the actual version of the platform and applications, we are using HTTPS, the same protocol that is being used by banking institutions for data confidentiality during communications. On the backend server, we are mainly enforcing Access Control Lists. However, on the mobile side, it is up to the user to secure his/her mobile and/or data.

In terms of scalability, it is practically not feasible to experiment our prototype at this stage with a high number of real and live patients. However, to ensure that our architecture scales with varying and high number of patients, we implemented a patients' simulator that imitates traffic between the backend system and mobile applications.

Up to now, we have been able to collect around 200 health records from neighboring hospitals. We are looking for more data but most of the time we are faced with security issues mainly with patients' privacy. To have a relatively high number of patients, the patients' simulator is using the same record many times, up to 10 times. The simulator is instructed in such a way that the virtual number of patients is randomly variable, from few up to 2000.

Using the patients' simulator, we haven't seen any noticeable degradation of the performance of the architecture or any unexpected high traffic. All traffic follows the same formulae presented above. It is to be noted that our experiments are using local area networks. However, we do not expect that getting patients on 3G/4LTE networks is going to put that much burden on the architecture thanks to the same reasons discussed above.

On the same scalability aspect and to evaluate the architecture with sensors that generate a significant amount of data, we integrated an advanced sensor, Bioharness 3 [40], which collects data on ECG, blood pressure, and temperature. Initial integration showed that there are no issues except for a little increase in network overhead, which was logically expected. A data purge policy is being designed in mobile applications to limit the size of data resulting from ECG monitoring to keep just data of an added value.

VII. CONCLUSION

Chronic diseases are affecting lives of millions of people all around the world and are becoming the primary causes of mortality. The number is so high to an extent where countries are allocating good percentage of their budgets to patients with such disease. Especially, when they need to be permanently hospitalized which add a huge burden on health systems, patients, and their families.

Transforming chronic diseases' patients into outpatients is a great relief for health systems, patients, and their relatives. However, outpatients do not benefit from the same level of monitoring inpatients have. This might increase the risk of unexpected health deterioration. These risks would be drastically reduced if outpatients can make use of thorough and continuous monitoring out of hospitals.

In this paper, we presented a novel framework for outpatients' chronic diseases monitoring using wireless body sensors and based on the Service-Oriented Architecture and the Cloud environments. Each patient in this framework is assigned a set of sensors, depending on his/her chronic disease(s), and an appropriate mobile application. Sensors get readings of various health parameters of the patients and automatically communicate these to the mobile application. The mobile application performs an initial processing during which it decides if there is a need to communicate some information to and from the backend system. The later communicates with the

mobile application and various health care personnel (e.g. physician, nurse, and nutritionist) to keep personnel updated with patients' vital health data and pushes any advices the personnel might have for the patient. These advices might range from proposing a lifestyle change to an invitation to urgently consult a physician. A prototype implementing some components of this framework has been developed and promising results have been shown. In fact, the framework has been able to detect all anomalies and communicate appropriate messages and data to different stakeholders. These activities are performed while minimizing the network overhead generated by communications of sensors, mobile applications, and backend systems.

As ongoing and future works, we are developing remaining components of the framework and integrating more sensors. We are also working on the implementation of key knowledge area related to data security, and Service Level Agreement (Quality of Service) guarantee.

REFERENCES

- [1] BIM, "Business Intelligence Middle East", at <http://www.bi-me.com/main.php?id=46797&t=1&c=35&cg=4&mset=>, Visited on 2012.
- [2] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, pp. 1-12, 2010.
- [3] G. Kaur and N. Gupta, "E-health: A new perspective on global health," *Journal of Evolution and Technology*, vol. 15, pp. 23-35, 2006.
- [4] Y. Chu and A. Ganz, "A mobile teletrauma system using 3G networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, pp. 456-462, 2004.
- [5] J. Fayn, C. Ghedira, D. Telisson, H. Atoui, J. Placide, L. Simon-Chautemps, P. Chevalier, and P. Rubel, "Towards new integrated information and communication infrastructures in e-health. Examples from cardiology", *Proceedings of Computers in Cardiology, 2003*, pub: IEEE. 2003, pp. 113-116.
- [6] E. Jovanov, "Wireless technology and system integration in body area networks for m-health applications", *27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS*, pub: IEEE. 2006, pp. 7158-7160.
- [7] R. S. H. Istepanian, E. Jovanov, and Y. Zhang, "Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, pp. 405-414, 2004.
- [8] N. Ludvig, H. M. Tang, N. S. Artan, P. Mirowski, G. Medveczky, S. L. Baptiste, S. Darisi, R. I. Kuzniecky, O. Devinsky, and J. A. French, "Transmeningeal muscimol can prevent focal EEG seizures in the rat neocortex without stopping multilineuronal activity in the treated area," *Brain Research*, vol. 1385, pp. 182-191, 2011.
- [9] N. S. Artan, P. Mirowski, H. Tang, G. Medveczky, S. Baptiste, H. J. Chao, O. Devinsky, R. Kuzniecky, and N. Ludvig, "Detecting abnormally large-amplitude multi-neuron bursts before focal neocortical EEG seizure onset in freely behaving rats," *Epilepsia*, vol. 50, pp. 391, 2009.
- [10] N. S. Artan, X. Li, R. Patel, C. Ning, N. Ludvig, and H. J. Chao, "Multi-layer coils for efficient Transcutaneous Power Transfer", *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC*. 2011, pp. 3031-3034.
- [11] N. S. Artan, H. Vanjani, G. Vashist, F. Zhen, S. Bhakthavatsala, N. Ludvig, G. Medveczky, and H. J. Chao, "A high-performance transcutaneous battery charger for medical implants", *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2010, pp. 1581-1584.
- [12] "Neuropace Responsive Neurostimulator (RNS)", at <http://www.neuropace.com/product/overview.html>, Visited on 2012.
- [13] J. Viventi, J. Blanco, and B. Litt, "Mining terabytes of submillimeter-resolution ECoG datasets for neurophysiologic biomarkers", *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2010, pp. 3825-3826.
- [14] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: non-invasive security for implantable medical devices," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 2-13, 2011.
- [15] QualComm, "2net Platform", at <http://www.qualcomm.com/wireless-health>, Visited on 2012.
- [16] CardioNet, "Mobile Cardiac Outpatient Telemetry™ (MCOT™)", at https://http://www.cardionet.com/medical_02.htm, Visited on 2012.
- [17] CareMatrix, "Carematix Wellness System (CWS)", at <http://www.carematix.com/>, Visited on 2012.
- [18] Pinned, "Pelex-04", at <http://www.pinmed.net/products/index-7.html>, Visited on 2012.
- [19] J. Grimson, W. Grimson, D. Berry, G. Stephens, E. Felton, D. Kalra, P. Toussaint, and O. W. Weier, "A CORBA-based integration of distributed electronic healthcare records using the synapses approach," *IEEE Transactions on Information Technology in Biomedicine*, vol. 2, pp. 124-138, 1998.
- [20] K. Ohe, "A hospital information system based on Common Object Request Broker Architecture (CORBA) for exchanging distributed medical objects--an approach to future environment of sharing healthcare information," *Studies in health technology and informatics*, vol. 52, pp. 962, 1998.
- [21] G. Juneja, B. Dournaee, J. Natoli, and S. Birkel, "SOA in healthcare (Part II)," *SOA Magazine*, 2009.
- [22] F. Kart, L. E. Moser, and P. M. Melliar-Smith, "Building a distributed e-healthcare system using SOA," *IT professional*, vol. 10, pp. 24-30, 2008.
- [23] HL7 and OMG, "The practical Guide for SOA in Health Care: A real World Approach to Planning, designing, and deploying SOA [version 1.0]", at <http://hssp.wikispaces.com/PracticalGuide>, Visited on 2012.
- [24] OMG, "Common Object Request Broker Architecture ", at <http://www.omg.org/spec/CORBA/>, Visited on 2012.
- [25] W3C, "Service Oriented Architecture", at <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, Visited on 2012.
- [26] S. Hsieh, S. Hsieh, Y. Weng, T. Yang, F. Lai, P. Cheng, X. Ping, M. Jan, J. Lin, and C. Peng, "Middleware based inpatient healthcare information system", *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering, BIBE*, pub: IEEE. 2007, pp. 1230-1234.
- [27] W. M. Omar and A. Taleb-Bendiab, "E-health support services based on service-oriented architecture," *IT professional*, vol. 8, pp. 35-41, 2006.
- [28] C. L. Yang, Y. K. Chang, and C. P. Chu, "Modeling Services to Construct Service-Oriented Healthcare Architecture for Digital Home-Care Business", *International Conference on Software Engineering & Knowledge Engineering*. 2008, pp. 351-356.
- [29] "GlassFish - Open Source Application Server", at <http://glassfish.java.net/>, Visited on 2011.
- [30] "MySQL Open Source Database", at <http://www.mysql.com>, Visited on 2011.
- [31] "Apache Tomcat", at <http://tomcat.apache.org/>, Visited on 2012.
- [32] "Liferay: Enterprise open source portal and collaboration software", at <http://www.liferay.com/>, Visited on 2012.
- [33] "Apache Web Server", at <http://httpd.apache.org/>, Visited on 2012.
- [34] "TWiki Server", at <http://twiki.org>, Visited on 2012.
- [35] Sanofi, "BGStar and iBGStar: Developing Integrated Diabetes Solutions", at <http://www.bgstar.com/>, Visited on 2012.
- [36] "JavaScript Object Notation", at <http://www.json.org>, Visited on 2012.
- [37] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," *ACM Transactions on Internet Technology (TOIT)*, vol. 2, pp. 115-150, 2002.
- [38] W3C, "Simple Object Access Protocol", at <http://www.w3c.org/TR/soap>, Visited on 2012.
- [39] K. Mohamed and D. Wijesekera, "Performance Analysis of Web Services on Mobile Devices," *Procedia Computer Science*, vol. 10, pp. 744-751, 2012.
- [40] Zephyr, "Bioharness 3", at <http://www.zephyr-technology.com/products/bioharness-3/>, Visited on 2013.