Emergency Medical Data Management through an Enhanced Cloud-based Push Messaging Mechanism

Vassiliki Koufi, Flora Malamateniou, and George Vassilacopoulos

University of Piraeus, Department of Digital Systems, Piraeus, Greece
{vassok, flora, gvass}@unipi.gr

Abstract

With mobile-cellular penetration in developing countries and billions of subscribers around the world, mobile health (mHealth) has emerged as a core element of effective field-level public health interventions. Especially when built on cloud infrastructures, mobile applications hold enormous potential for transforming healthcare delivery systems which currently involve cumbersome processes that slow down care and decrease rather than improve safety. Emergency medical systems (EMS) stand to benefit from mobile cloud applications as they involve a variety of activities which are performed from the time of a call to an ambulance service till the time of patient's discharge from the emergency department of a hospital. This paper is concerned with NefeliMobile, an Android application that facilitates access to a PHR-based EMS, developed in a cloud computing environment and places special focus on an upgrade of the custom notification mechanism incorporated in it whereby notifications for new emergency incidents are routed to the caregivers bearing less load. Henceforth, caregivers are notified on critical PHR data updates of their patients in a way that top priority messages are getting immediate attention.

Keywords: *Android, Personal Health Record, Google Cloud Messaging*

1. Introduction

With the drive to achieve excellence in healthcare, the flow of information both within and across healthcare organizations needs to be seamless, simple and effective. Building on leading-edge technologies, such as mobile communications and cloud computing, mobile health (mHealth) holds enormous potential for realizing a fully integrated environment thus transforming healthcare delivery systems which currently involve cumbersome processes that slow down care and decrease rather than improve safety [[1]][[2]][[3]]. As mobile technology can offer unmatched convenience in accessing cloud-based electronic health records, it may assist in developing mission critical applications for healthcare delivery in all levels of care (e.g. emergency care). Thus, healthcare providers and organizations are increasingly considering moving towards mobile cloud solutions in an attempt to increase flexibility and agility of their healthcare systems and services and enhance quality of patient care. Emergency medical systems (EMS) stand to benefit from a potential move towards such solutions as they involve a variety of activities which are performed from the time of a call to an ambulance service till the time of patient's discharge from the emergency department of a hospital.

This paper is concerned with the development of a PHR-based EMS system, namely NefeliEMS, which provides ubiquitous access to medical information stored and exchanged in dealing with emergency cases through desktops and Android-enabled mobile devices [**Error! Reference source not found.**]. In the context of NefeliEMS, NefeliMobile has been developed, an Android native application that provides access to NefeliEMS and incorporates a customized asynchronous notification feature, whereby caregivers are notified on critical PHR data updates shortly after their occurrence. This feature is essentially a lightweight cloud-based push messaging mechanism, which builds upon and enhances Google Cloud Messaging (GCM) by providing message classification capabilities [**Error! Reference source not found.**].This paper presents an enhanced version of this mechanism where, in addition to the prioritization of GCM notification messages, load balancing capabilities have been incorporated in order to ensure optimal performance by emergency responders.

2. Motivating Scenario

The basic motivation for this research stems from our involvement in a recent project concerned with developing a prototype EMS system for the needs of the Greek National Health Service with the objective to improve quality of care while containing cost. To illustrate the main principles of the system architecture incorporated into the NefeliEMS, consider an EMS process scenario that shows an example of how an EMS may work. Figure 1 shows a broad view of the EMS process.

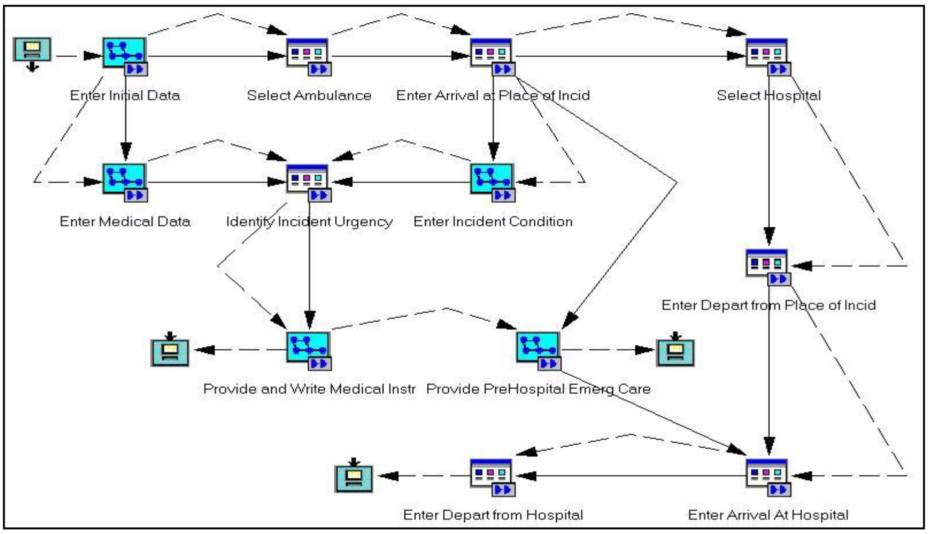

Figure 1. A high level model of an emergency care process.

In the EMS process of Figure 1, where the ambulance service and a hospital emergency department are involved, five user roles are identified: ambulance communication operators, ambulance service physician, ambulance paramedics, emergency department physician and emergency department nurse.

- *Ambulance communication operators* - Ambulance communication operators are located in the ambulance center premises and use an EMS application to write into a PHR system emergency case data which is provided by either the patient or another person and to pass this data to ambulance personnel.
- *Ambulance service physician* - Ambulance service physicians are usually located in the ambulance center premises and use an EMS application to read from and write into a PHR system relevant medical data of their current patients (e.g. medical history, patient allergies and other critical health factors) so that to give appropriate medical instructions to ambulance paramedics, regarding en route treatment, that are also recorded.
- *Ambulance paramedics* - Ambulance paramedics use an EMS application, via a smart handheld device, to read authorized portions of medical data from a PHR system and write data regarding the paramedic activities performed on the patient at the site of incident and en route.
- *Emergency department physician* - Emergency department physicians use an EMS application to read from and write into a PHR system relevant medical data of their current patients (e.g. medical history, patient allergies and other critical health factors).
- *Emergency department nurse* - Emergency department nurses use an EMS application to read authorized portions of patient data from a PHR system, to write a nursing assessment of patient's condition (triage) and to write data of the nursing activities performed on the patient.

3. System Architecture

In the context of NefeliMobile, information pertaining to emergency case management is transferred and stored to mobile devices owned by authorized personnel at the beginning of their shift [**Error! Reference source not found.**]. Henceforth, data are delivered to these mobile devices on the occurrence of certain events (e.g. new emergency incident, PHR data update etc), through the GCM service. This can be achieved by either a lightweight message notifying NefeliMobile that there is new data to be fetched from the server or by a message containing these data (e.g. medical instructions provided by the ambulance service physician to ambulance paramedics) so that the application can consume the message directly, extract the data it contains and store it in the local repository. The components participating in the notification service are the Android-enabled mobile devices running NefeliMobile, the NefeliEMS application server and GCM Servers.

By default, GCM is configured to broadcast notifications to every successfully registered device, irrespective of the active context and the current load of target devices. Moreover, prioritization of notification messages depending on certain criteria, such as the severity of the emergency case, is not supported, since all notification messages are handled by GCM as of equal importance. In an attempt to address these issues, a custom servlet has been developed, namely NefeliMediator, which is hosted on the cloud application server and serves as a mediation gateway, between NefeliEMS cloud infrastructure and GCM. In the context of NefeliMediator messages regarding an emergency incident are classified depending on the severity of the incident. Scheduling of messages, which is performed after their classification, depends not only on the severity of the case but also on the certain attributes possessed by the mobile device (e.g. message load, etc) and its owner (e.g. mobile user role, shifts, etc) and the context holding at the time a notification message is being placed (e.g. time, location etc.).

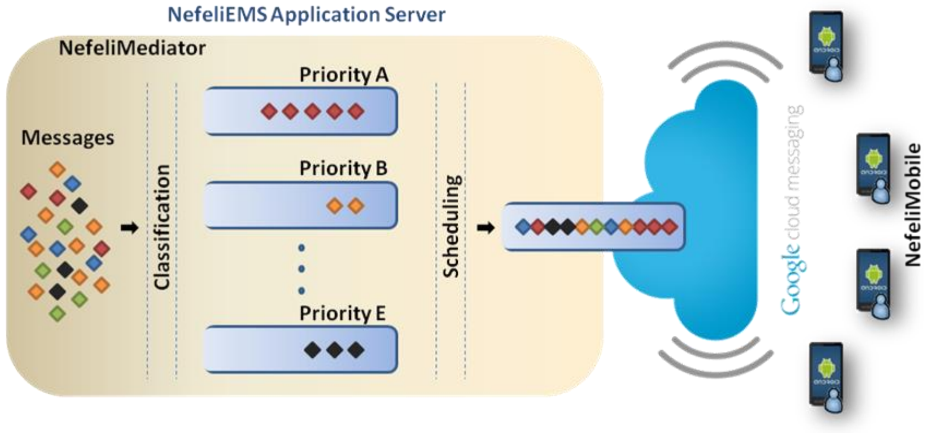

Figure 2. Routing Mechanism.

On the occurrence of a new emergency incident, the initial data regarding the incident are recorded. This triggers the submission of a notification message to the members of the staff (ED physician, ED nurse) who will treat the patient upon arrival to the hospital's ED. Each message of this type is marked as "the first message for this particular emergency incident" by means of a 0/1 flag {F}. Thus, the first message regarding an emergency incident is distinguished from all subsequent messages of the same or other incidents. As soon as the notification messages are generated, they are being placed to different priority queues depending on the severity of the relevant emergency case, which is indicated by the key/value pair "priority, *priority_value*" included in the GCM message payload (Figure 2). Five levels of severity have been identified for the classification of emergency incidents, namely A for maximum possible priority (e.g., witnessed cardiac arrests), B for high priority incidents, requiring advanced life support, C for incidents requiring advanced life support, D for medium priority incidents, and E for low priority incidents. Within each of the priority queues, messages are scheduled in First In First Out (FIFO) order. Messages are submitted from the head of a given queue only if all queues of higher priority are empty. However, if the amount of high-priority messages is not conditioned, the submission of messages concerning lower-priority incidents may experience excessive delay. In order to avoid such situations, a polling system has been utilized to switch between queues. In particular, a 7:4:3:2:1 approach has been adopted regarding the maximum number of continuous messages that can be transmitted from each queue, starting from the more urgent ones. If a queue is empty then the queue is skipped and the messages of the lower priority queue are transmitted.

Upon selection of a message from the head of a queue, NefeliMediator detects the target mobile devices where the message should be multi-casted depending on the current context (e.g. the mobile devices owned by ED physicians or ED nurses who are currently on duty and are less busy than other colleagues of theirs). For ensuring optimal distribution of the emergency incidents, a dynamic policy is utilized for balancing load among the number of physicians prior to message transmission. The number of physicians {N} and their specialties

{S}, along with the number messages {M} assigned to each physician and the number of messages {P} processed by him are crucial initial conditions in designing an efficient strategy. In particular, {S} determines the types of emergencies that a physician can handle, while {F} shows if a particular case is assigned to a physician and hence he is responsible for handling further messages. Moreover, {M} and {P} provide information on the status and available capacity for each physician during system operation. Hence, during the queue polling process, for each message exiting a priority queue a feature vector {N,S,M,P,F} is used to determine the physician whose mobile device should receive the relevant notification message.

Upon system startup, all messages are considered to have a zero flag value {F} and for each message arriving from its equal priority queue an optimal allocation is made to the set of physicians, taking into account the number of physicians {N} and their respective specialties {S}. During system operation the ratio of assigned to processed messages {M}/{P} for each physician is periodically updated using a notification process and compared against a threshold which in the ideal case should be close to unity. The {M}/{P} value is evaluated over time and the physician to be assigned the next message is determined based on the lowest value of {M}/{P}. It should be noted that prior to the selection of the physician based on the {M}/{P} value, features {F} and {S} are checked for being consistent with the constraints set initially. The system performance is continuously monitored and if a constantly increasing {M}/{P} value is observed for a large set of specialties a proper contingency action is used like increasing the number of physicians for a specific specialty {S}.

Once the GCM messages are triggered, all other queuing aspects are handled by GCM service along with message delivery aspects.

4. Implementation Details

The prototype implementation of the EMS system has been performed on a laboratory cloud computing infrastructure. Apache/Tomcat was used as Web/Application Server where ActiveBPEL is hosted. ActiveBPEL is an open source BPEL engine used for the execution of BPEL healthcare processes[[6]]. The platform used for the generation of sample patient PHRs is Tolven ePHR as it has been considered sufficient for the purpose of our research [[7]]. Acquisition of information regarding EMS processes deployed on the BPEL engine was achieved by means of web services. However, in order to enable access to PHR data stored on the cloud a number of web services had to be developed. NefeliMobile has been implemented in a prototype using Android SDK 4.0.3r2 API15 while Java SE 7 has been used for the implementation of NefeliMediator [[8]]. SQLite, the database platform supported by Android, has been used for replicating patients' PHRs locally on healthcare professional's Android device.

5. Concluding Remarks

By pairing mobile applications with cloud computing infrastructures, challenges and the diverse needs of global health stakeholders can be addressed through scalable, flexible, and sustainable solutions. Together, cloud computing and mobile communications can support consolidation of diverse and otherwise inaccessible sources of vital public health information into a single, secure, decision-oriented environment. NefeliMobile is an Android native application that provides mobile access to NefeliEMS, a PHR-based EMS implemented in a cloud computing environment. In order to ensure that occasional, time-critical data receive immediate attention by authorized personnel, a notification mechanism has been incorporated in NefeliMobile, which draws upon GCM, according to which every transaction for medical data acquisition is initiated on the server side on the occurrence of certain events. One important component of this mechanism is NefeliMediator, a custom servlet which prior to the initiation of notification message submissions it performs prioritization of these messages according to certain criteria such as the severity of the emergency incident. This paper presents an updated version of NefeliMediator which incorporated load balancing features in order to ensure maximum throughput of the system.

References

[1] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Gener. Comput. Syst. 25(6), 599–616 (2009)

[2] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A. , Zaharia, M.: "Above the clouds: A berkeley view of cloud computing," Technical Report (2009)

[3] Rosenthal, A., Mork, P., Li, M.H., Stanford, J., Koester, D., Reynolds, P.: "Cloud computing: A new business paradigm for biomedical information sharing," J. Biomed. Inform. 43(2), 342 – 353 (2010)

[4] Koufi, V., Malamateniou, F., Vassilacopoulos, G., Prentza, A.: An Android-Enabled Mobile Framework for Ubiquitous Access to Cloud Emergency Medical Services. In: 2nd Symposium on Network Cloud Computing and Applications (NCCA 2012), London, United Kingdom, December 3 – 5 (2012)

[5] Koufi, V., Malamateniou, F., Vassilacopoulos, G.: A Sophisticated Mechanism for Enabling Real-time Mobile Access to PHR Data. In: International Conference on Informatics, Management and Technology in Healthcare (ICIMTH 2013), Athens, Greece, July 5 – 7 (2013)

[6] Active Endpoints. Activebpel open source engine project, http://www.activebpel.org/

[7] Tolven ePHR, http://www.tolven.org/ephr.html

[8] Android, http://www.android.com/