2017 21st International Conference on Control Systems and Computer Science

Multi-agent based e-health system

Cosmin Gabriel Toader
University Politehnica of Bucharest, cosmintg@gmail.com

***Abstract*—This paper proposes the design of an intelligent integrated system for sending and processing personalized medical data and storing them in the cloud according to modern standards. A mobile hardware component will be used, that features a set of medical sensors (tonometer, EKG, pulse oximetry, temperature, accelerometer, respiratory rate, electromyography, GPS) and is able to serialize them in the standard HL7 format and send them through an internet connection (2G, 3G, 4G, wi-fi) to the cloud server. The equipment will allow the patient (or someone who is helping the patient) to initiate audio/video streams to the medical personnel. (*Abstract*)**

Keywords—multi-agent systems; E-health; Raspberry-pi; MLLP; HL7

I. Introduction

With the increase of calculation power and computational intelligence in various devices, a new trend arose called IoT (internet of things). IoT already made a strong presence in several applications in e-health, most of them in patient surveillance area. This means a huge number of data sources is now available for use, and those applications already take advantage of it [1].

Those devices are capable of processing data in real-time and adjust to changing conditions [2], not only to collect and stream.

Typically in MAS, a mobile agent is a piece of autonomous software, capable of migrating between nodes of a network. With IoT in place, we can now have other forms of mobility, being able to physically move the device.

Multi-agent systems have been quickly adapted for the e-health area, therefore more and more work has been done over the last few years. In a Multi-agent architecture for real-time Big Data processing has been introduced. It has been shown that classical lambda architecture can be extended as a Multi-Agent System. In some related work that we found [3], the idea is taken further, as we know how data is exchanged between the processing layers and devices located in the physical environment.

A multi-agent system approach was taken where the data is pre-processed on the client-side as a model and sent to the server, and the client would just keep that model updated[4, 5]. This approach is called "model-to-data" and it was proven very effective as it saves both server processing power and latency.

In [6] the authors have discussed the use of Mobile agent technology in communication between different remote locations. When an emergency occurs, especially in mass casualty incidents, lots of victims need medical attention as early as possible. This paper advocates the use of mobile agents for faster and accurate acquisition of data, where proper health care equipment or analysis of medical information is not possible.

Mobile agents have been also used in telemedicine [7]. This term refers to an application area that couples the technology of computer and communication with medical services. The telemedicine exists in different forms: tele-education, tele-consultation, tele-surgery, tele-monitoring. In this context, mobile agents are broadly perceived as aiding medical and telemedicine applications. That is why, the solution offered by this work is to design a safe agent-based telemedicine based on P2P networking architecture. When a mobile agent holding patient information travels from one place to another place through the Internet, it can be attacked by other malicious agents. So, the architecture proposed in [7] uses a two-layer safety mechanism for mobile agents to provide a solution for agent-based telemedicine services sustaining important aspects as confidentiality, reliability, integrity and non-repudiation.

Another related work about multi-agent systems in e-health consists in a mobile piece of hardware installed in an ambulance vehicle able to determine the best hospital to send a patient to [8].

This system involves context awareness [9] and based on the patient's data and the nearby hospitals will come up with a decision. The scenario is the following: a person is in dire need of an ambulance, having a certain condition. That person or some nearby help calls for an ambulance and communicates the problem and the patient's ID. This data will be immediately introduced into the proposed system. In this system, two agents are introduced:

- Data-assistant: this agent will contact the relevant medical database and retrieve the patient's medical records.
- Hospital-assistant: this agent will search nearby hospital and sort them by various criteria (facility type offered by the hospital, number of available doctors, number of available nurses etc.)

Those two agents then will negotiate to decide which

2379-0482/17 $31.00 © 2017 IEEE
DOI 10.1109/CSCS.2017.107

hospital is the best choice and after that, the hospital will be contacted to announce the patient's arrival.

In this context, the goal of this work is to take advantage of IoT and multi-agent systems and design the architecture of an intelligent system that sends, processes medical data and makes decisions based on it. It involves a mobile hardware part with various sensors that can collect data from a human being (ekg, pulse oximetry, tonometer, temperature etc.) and a server side that aggregates the data and presents it to the relevant medical authority.

The mobile component will have three modes of transmission:

- normal data transmission, where periodic data is collected from the patient and sent to the cloud through MLLP (minimal lower layer protocol) and serialized in the standard HL7 format
- Emergency data transmission, where only a set of minimal data will be transmitted (that can be sent even via 2G connections). The decision to enter this mode is based on some parameter thresholds, for example the O2 level being lower than the minimal admitted value.
- Audio/video stream, that the user can initiate and will be possible based on the connections

The mobile component's hardware consists of an e-health platform based on raspberry-pi with various sensors described earlier (Fig. 1). The e-Health Sensor Shield allows Arduino and Raspberry Pi users to perform biometric and medical applications where body monitoring is needed by using 5 different sensors. This information can be used to monitor in real time the state of a patient or to get sensitive data in order to be subsequently analyzed for medical diagnosis. Biometric information gathered can be wirelessly sent using any of the 6 connectivity options available: Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 and ZigBee depending on the application.

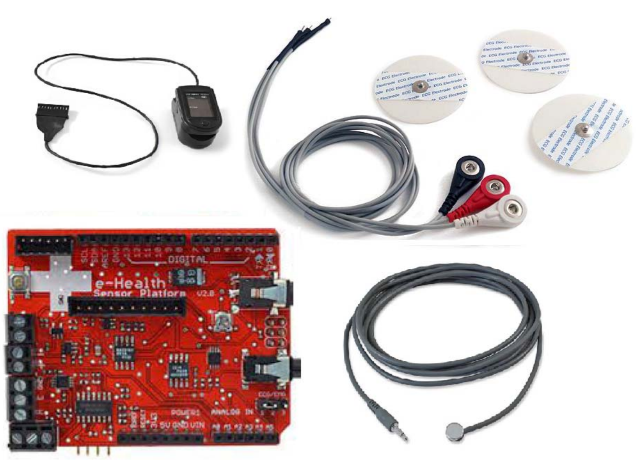

Figure 1: E-health sensor platform

The cloud component is a web system that stores the medical data, aggregates, processes them and allows the medical personnel to access them. This component will be composed of multiple services (patient's data, emergency, monitoring, medical Uber, configuration, etc.)

The paper consists of four sections. After introduction, the second section describes the mobile client architecture [10], including a diagram on how we plan to develop the client side. The third section presents the sensors that will monitor the vital signs, the fourth section will describe the server side, the services required and the proposed technology stack that will be used. The last section focuses on conclusions and future work.

II. MONITORING THE VITAL SIGNS

The vital signs or physiological parameters are the critical factors to determine the individual's health status. These signs measurement are very important assessment, which includes counting the number of pulses in one minute and checking forehead palpation for body temperature manually. Monitoring them become critical procedure to gain information about the health status of patients in any given scenario. That is why there have been continuous improvement and enhancement of the vital signs collection equipment, transmission protocols and graphical presentation for the doctors in an informative and easy to understand approach [11].

The values of all the bio-signals taken from different sensors of eHealth platforms influence the medical professional's interpretation of a patient's overall condition and affect the course of treatment for each patient individually. Pulse is defined as the palpable rhythmic expansion of an artery produced by the increased volume of blood pushed into the vessel by the beating of the heart. In most clinical circumstances pulse rate is very similar to heart rate. Some other factors such as irregular pulse or if the person is cold, play a significant role in the inaccuracy of reading. The Oxygen Saturation will be also monitored by our system.

Other vital sign is the body temperature. The balance between heat generated and heat lost is represented as the body's internal temperature. Core temperature is technically difficult to measure but in most clinical circumstances it is acceptable to approximate core temperature by measurement of peripheral body temperature. So the proposed platform will also use data collected by this sensor. The system will also pay attention to monitoring the blood pressure. This process is also important at home especially in the case of the persons having high blood pressure.

Other data gathered by the e-health sensor platform regard electrocardiogram, the respiratory rate and the glucose in blood.

III. MOBILE CLIENT ARCHITECTURE

The mobile client will be composed of three agents, each being responsible for collecting data from either sensors or the user and negotiating between them to decide which mode to use in order to send the data (Fig. 2):

- Periodic analyzer agent will gather data periodically from the sensors and if relevant, will serialize them in the HL7 format [12] and send them to the cloud.

- Emergency agent will gather data from the sensors, decide if there are critical parameters and send minimal data to the cloud.
- Manual handling agent can override the other two agents' decisions and will initiate the audio/video call to the medical personnel.

The HL7 standard is a framework for the exchange, integration, sharing and retrieval of electronic health information between e-health software applications. The first version of this standard, HL7 1.0 dates since 1987, and right now, the latest version of HL7 is v3 [13].

The HL7 (health level 7) refers to the highest level of International Standards Organization (ISO) communication model for Open Systems Interconnection (OSI) (it resides on the application layer). The latest version of HL7 is designed to be scalable, using a formal object oriented design methodology.

In order to send HL7 messages, a protocol is needed. The most common HL7 transport method is LLP (lower layer protocol), also referred as MLLP (minimal lower layer protocol, a standard for transmitting HL7 messages over TCP/IP. A less common protocol is the HLLP (hybrid lower layer protocol).

The automatically collected data will be sent using the MLLP protocol (minimal lower layer protocol), which is a standard in e-health and will make sure the traffic will stay low. The typical structure of an HL7 message being sent via MLLP contains four parts: a header (vertical tab character), the actual HP7 message, a trailer (field separator character) and a carriage return. Those separators are needed because of the TCP/IP being a continuous stream of bytes.

After the negotiation phase of the agents, the client will know exactly what data to send and who it is designated to (general patient history, emergency room, patient's doctor, etc.)

IV. CLOUD SERVER

The main server module will be called "Cloud Service" and it will be responsible with taking data from the mobile clients (fig-3). It will have different endpoints for each of the three modes of transmission: two endpoints for MLLP and one for audio/video streaming. Upon receiving data, the server will process it and decide to store it and if it should raise an event handler to notify the right module or make the audio/video streaming connection.

The main advantages of cloud computing [14] consist in a few strong pillars:

Reliability - having a well-managed service platform, cloud computing is much more reliable and consistent than in-house IT infrastructure.

Manageability - cloud computing provides simplified management and maintenance capabilities through central administration of resources. Cloud-based services are ideal for services with growing or fluctuating bandwidth demands. In case of disasters, unlike in-house IT infrastructure, the recovery is imminent, requires less resources, resulting in a more manageable problem.

Perhaps the most significant advantage is the cost savings pillar. Unlike other services, using the cloud is brings a revolutionary view to technology and it's low-cost

The chosen technology for this service is .Net Core, the main reasons being that it is an open source technology and cross-platform.

The second server module will be the web component, which will have the standard MVC architecture, also based on .Net Core technology. The web server will work closely with the cloud service and it will serve web content to the medical personnel.

This web component will use SignalR, a technology that will allow us to send real time notifications to the web browser in case of the emergency module making a call.

The event handler component plays an important role in this architecture. It holds the business logic for handling real-time events, like a patient's vital signs falling below a critical threshold, or multiple patients in the same area suddenly needing medical intervention for similar reasons.

This event handler is responsible with processing the critical situations and raising the event to the most relevant authority. This could mean sending alerts through SignalR, sending an e-mail, a text message, or even initiating a video-call automatically.

A scenario for this system would be the following: a patient suffered from a burning accident and is being monitored by the mobile hardware. The data gets to the cloud, but the parameters show that it is a minor situation, maybe only a bit over the threshold. Another patient from the same accident uses the mobile hardware, sending the same data to the cloud (more or less) and then maybe another one, since a fire usually produces multiple victims.

The cloud just got similar data from patients in the same area, using their geolocation. It can now decide to raise an event to the relevant authorities and personnel about this, so the staff will figure out it is a larger scale incident and take actions against it.

Both the cloud service and the web component will be able to store and read data from the database via a repository design pattern.

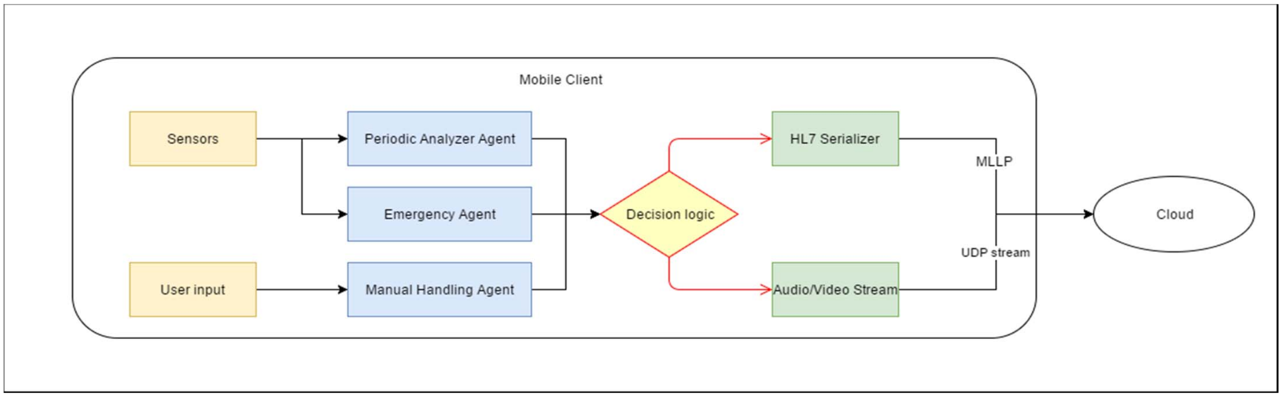

Figure 2: The mobile client architecture

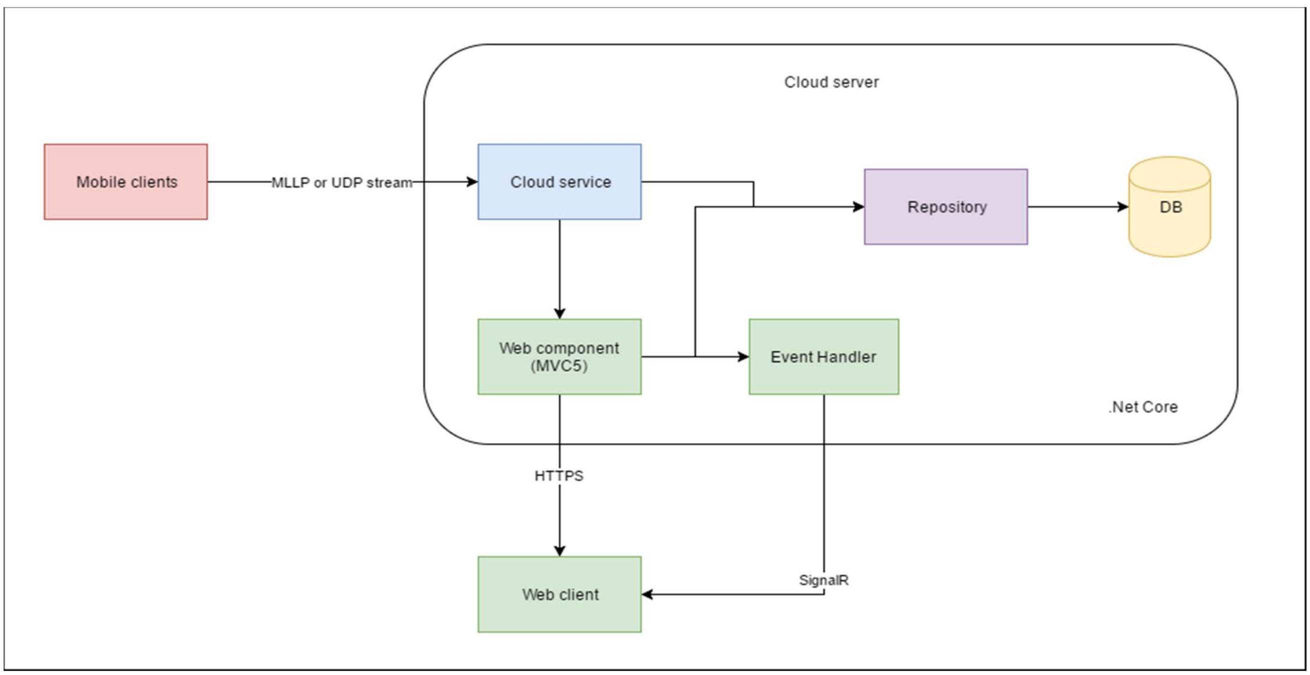

Figure 3: Cloud service architecture

V. Conclusions

The proposed project will help the medical system gather data in standard format and the intelligent system will be able to cluster the data in order to detect various incidents (mass accidents, virus spread, fire incidents, etc.)

The presented architecture is scalable, so it will allow us to add or change business logic without any major restructure.

The chosen technologies are open-source and cross-platform, so they will be easily integrated in most of the existing hardware systems and operating systems.

Now that we have a basic scaffolding of the architecture and it is scalable enough, in the future we will be able to build upon it. The aim of the project is to detect mass accidents and hazards and to decide how many resources to allocate (e.g. how many ambulance vehicles to send, how many medical personnel etc.). The mobile client aims to be a context aware system following the recommendations studied from related work [15].

The next step is to find a better way to process data in the cloud so the medical personnel that use the dashboard web application will be alerted as soon as possible with accurate statistics about hazards.

REFERENCES

[1] A. L. A. Lella, “The u.s. mobile app report (whitepaper),” 2014

[2] Nirvana Popescu, Vlad Ciobanu, Adrian Ghidel, Flavius Dinu, Decebal Popescu, “Predefined Recovery Exercises System for After Stroke Hand Rehabilitation”, EPE2016: International Conference and Exposition on Electrical and Power Engineering, Iasi, Romania, Oct. 2016, pp.61-67

[3] Bartłomiej Twardowski, Dominik Ryzko, "IoT and context-aware mobile recommendations using Multi-Agent Systems", 2015

[4] L.-C. WANG, X.-W. MENG, and Y.-J. ZHANG, “Context-Aware Recommender Systems,” pp. 1–20, 2012

[5] B. Hidasi and D. Tikk, “Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback,” inLecture Notes inComputer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7524 LNAI, no.PART 2, 2012, pp. 67–82

[6] M. V. Prem and S. Swamynathan,”Role of Mobile Agents in Medical Information Retrieval in Mass Casualty Scene –A Performance study in Web Environment‖”, WseasTransactions on information science and applications vol. 8, no.10, 2011, pp.407-416

[7] W. Hsu and J. Pan,”Secure Mobile Agent for telemedicine based on P2P networks”, Journal of Medical Systems, 2013.

[8] Burstein, F., Zaslavsky, A., Arora, N., "Context-aware mobile agents for decision-making support in healthcare emergency applications"

[9] A.Karatzoglou,X.Amatriain,andN.Oliver,“Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering,”Context, pp. 79–86, 2010. [Online].Available: http://portal.acm.org/citation.cfm?id=1864727

[10] B. Y. M. Yun, “Research on the architecture and key technology of internet of things (iot) applied on smart grid,”Advances in Energy Engineering (ICAEE), pp. 69–72, 2010

[11] K. D. Johnson, “ Patients’ Vital Signs and the Length of time between the Monitoring of Vital Signs during times of Emergency Department Crowding”, *PhD. thesis,* University Cleveland, United States, 2011.

[12] Elena Madalina Rac-Albu, Vlad Ciobanu, Marius Rac-Albu, Nirvana Popescu, „Interoperability of Medical Data Through eHealth service in Romania”, IESS 1.6: International Conference on Exploring Service Science, Bucharest, May 2016, pp 11-21

[13] Common HL7 transports, Interfaceware - https://blog.interfaceware.com/common-hl7-transports/

[14] Robert Basmadjian1, Hermann De Meer, Ricardo Lent and Giovanni “Cloud computing and its interest in saving energy: the use case of a private cloud”, J. of Cloud Computing, Vol 1, issue 1, 2012.

[15] Y. Shi, A. Karatzoglou, L. Baltrunas, and M. Larson, “CARS 2: Learning Context-aware Representations for Context-aware Recommendations,”2014