УДК 004.415.2:338.486.61

ABOUT DEVELOPMENT OF THE PROJECT "PERSONAL MEDICAL TELEMONITORING"

Rena Sultangazieva, *PhD docent, KSTU named after I.Razzakov, 720044, Kyrgyz Republic, Bishkek city, Ch.Aitmatov Avenue, 66, E-mail: renasultangazieva@mail.ru*
Miklos Kozlovszky, *PhD, John von Neumann Faculty of Informatics, Obuda University, H-1034, Budapest, Becsi str. 96/b., Hungary, E-mail: kozlovszky.miklos@nik.uni-obuda.hu*

Abstract. This work presents the next steps for the development of a mobile service for personal ECG monitoring, which was started with the financial support of the CAREN-EYR (Enlighten Your Research) program with the technical support of the CAREN network and Obuda University (Hungary). The purpose of this project is to develop the service "Personal ECG Monitoring", which allows to register patients ECG data using mobile medical sensors, then to send data by the Android application via a cellular network to the server for providing operational information to the doctors from National Center of Cardiology. Different ECG sensors from different vendors use different data formats: JPG, PDF, SCP, DICOM, MFER, HL7, binary. To develop a unified database of patients, it is necessary to resolve how to integrate ECG data of different formats. To protect medical data, we use the HTTPS protocol of CRENA, as well as Base64 coding and OAuth based secure authentication.

Keywords: ECG monitoring, telemedicine, mHealth, Savvy ECG, mySignals, Olimex

Introduction

This work presents the next steps for the development of a mobile service for personal remote ECG monitoring, which was started with the financial support of the CAREN-EYR (Enlighten Your Research) program with the technical support of the CAREN network and Obuda University (Hungary). The purpose of this project is to develop the service "Personal ECG Monitoring", which allows to register patients ECG data using mobile medical sensors, then to send data by the Android application via a cellular network to the server for providing operational information to the doctors from National Center of Cardiology in Bishkek (Kyrgyzstan).

To implement the ECG monitoring service, it is necessary to solve the questions: what mobile ECG sensors can be used and what is the price of these sensors to form the cost of the service.

Currently, the Institute of Cardiology uses mobile PC-based EDAN ECG (Китай) for Holter Daily Monitoring. ECG data is transferred to the computer of the medical professional with cable. Wired communication requires the physical presence of the patient and they have to repeatedly come to the hospital premise (Bishkek). Russian company Shvabe presented a mobile ECG to the National Center of Cardiology. The mobile ECG allows you to measure the ECG data of patients in rural area and send data via the Internet to the cardioserver for analysis and to the ECG Automatic Interpretation Server, that is a cloud Internet service of Russia company. But this system is not commercialized yet and fully owned by the Russian company.

Our project extensively relying on the OpenEMR software solution [5]. OpenEMR is an Open Source electronic medical record and medical practice management software. Crucial part of our project is to identify available market ready low-cost, portable, medical ECG devices. We are integrating these sensor devices into OpenEMR and furthermore our aim is to develop an integrate server solution to enable interoperability between the various ECG sensor devices using different ECG data formats.

The market of portable ECG sensors is not yet developed in Kyrgyzstan, while a wide number of wearable devices with heart rate measurement capabilities are now available, both stand-alone devices and integrated with Internet-of-Things infrastructures. In the last decades, many standardization efforts have been made with ECG data. Most of the ECG devices are using proprietary data protocols and the vast amounts of available standards makes true interoperability a difficult task..

To develop a unified common measurement database for the patients, it is necessary to resolve the existing interoperability issues between major standards and realize integrated, automatic ECG data format conversion. ECG signals can be stored in SCP-ECG, DICOM-ECG, HL7 aECG or HL7 FHIR formats just to name the most important ECG standards formats. In addition to the many ECG standards, we can still observe today's (mostly binary) format used by many equipment manufacturers. The existing many formats prevent the patient's ECG data from being simply unified (eg.: resting ECG, outbound ECG, ECG for clinical trials, ECG format for remote/mobile patient monitoring devices). The most commonly used ECG standards nowadays support a 12-channel ECG data, where the channels are from 30 seconds up to a one-minute length of data recording. Most of the wearable equipments used for remote patient monitoring use a single channel and can take hours, or a week-long recording.

Table 1. HL7 aECG vs. SCP-ECG vs. DICOM ECG standard formats - comparison

	HL7 aECG	**SCP-ECG**	**DICOM ECG**
Small file size	Big	Small	Medium
Human readable	Yes	No	No
Simple schema	No	No	No
Several modalities in a single format	No	No	Yes
XML vs. Binary	XML	Binary	Binary
Easy to use	Yes	No	No
Continuous ECG streaming data storage	No	No	No

The result is large set of proprietary and standard ECG file formats based on XML, JSON, JPEG, PDF or other solutions. The number of wearable ECG devices for remote patient monitoring is currently still limited, but tendencies are increasing. Some devices are sold as standalone portable ECG devices (eg.: Meditech CardioBlue, Savvy Pcard) and other wearable devices have additional features (eg.: Apple iWatch4 or v5, Sanatmetal WIWE, Beurer BM 95). The variety of available formats can cause problems, when trying to collect patient data for ECG reports, since we do not know at the server side, which ECG device the patient is using for the measurement in advance.

To collect and store ECG data (and patient data), we examined the following ECG sensors:

1) **Savvy ECG** (Ljubljana, Slovenija) is a single-channel ECG with long-term recording capability that communicates via Low Power Bluetooth with the mobile phone. This sensor was kindly provided by the Biotech Research Center at Obuda University for the implementation of our project. The Savvy ECG has a mobile application, which is capable to store and forward ECG measurements as pdf files to a pre-defined email address or archive it locally at the smartphone. At the first stage of the project, we developed the architecture shown in the figure 1. [1]

ECG data from the Savvy sensors were sent to FTP server installed in the information center at KRENA. The official site SAVVY provides the exe-file of VisECG program, which makes it inconvenient to use on Linux systems. VisECG is an analysis and visualization program for ECG measurements recorded by a SAVVY device via MobECG Android application. Doctors cannot view detailed data via the server's web interface; they have to copy the patient's s2- file to their computer for more detailed analysis.

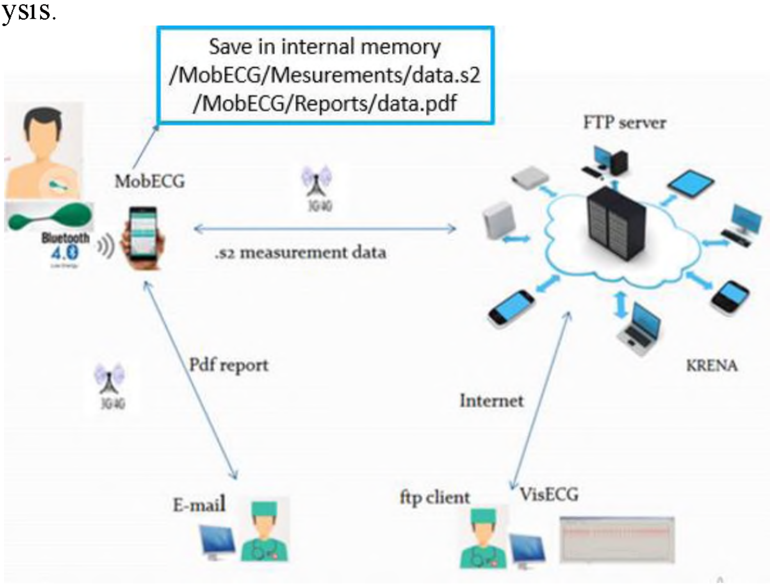


Figure 1. Architecture of the mHealth network

At the second stage, we developed mobile application, which found ECG files saved in the Android's internal memory, converts to json format and sends to the server (fig.2)

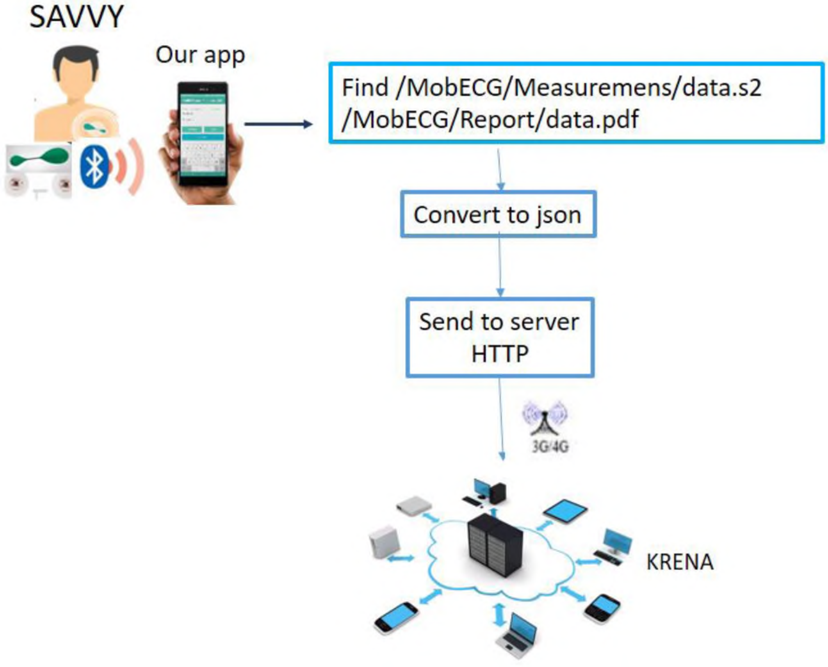
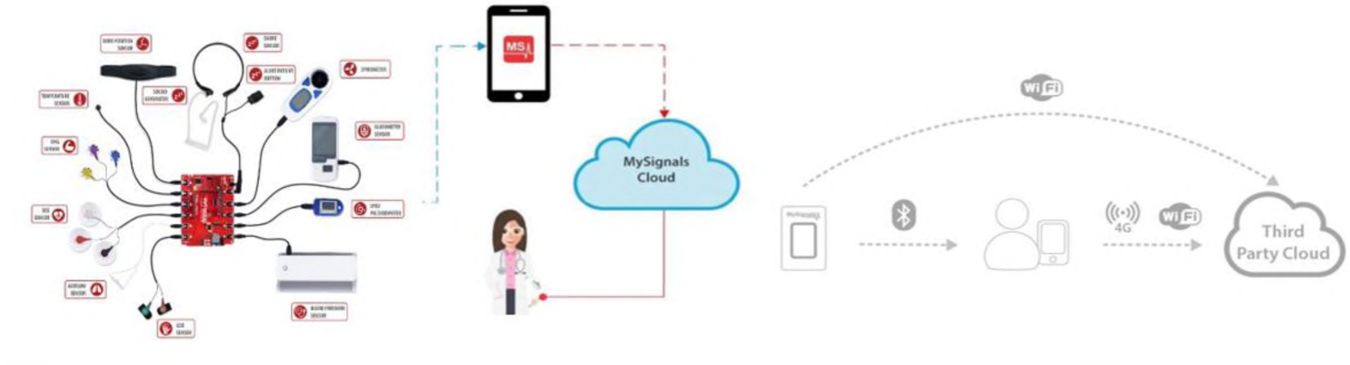


Figure 2. Mobile app for Savvy

2) **MySignals**, eHealth and Medical Development Platform for Arduino is an advancement stage for therapeutic gadgets and eHealth applications [2]. Data from MySignals is scrambled and sent to the Primary Health cloud database through WiFi or Bluetooth. Developers may send the information coming from MySignals to a third party Cloud server using directly the WiFi. (fig.3)



Figire 3. MySignals

3) **AliveCor Kardia** is a hand-held single-lead Electrocardiography (ECG) device that utilizes a smartphone to detect and monitor Atrial Fibrillation (AF).

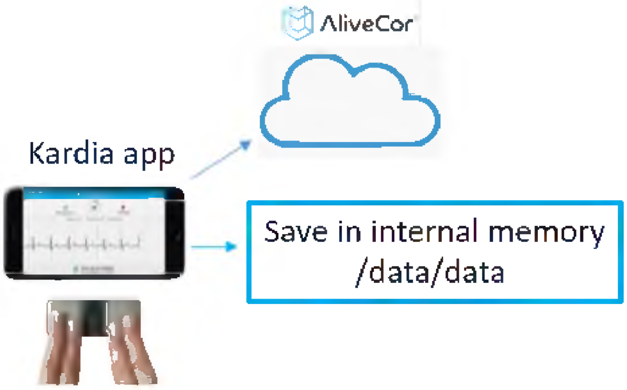


Figure 4. AliveCor Kardia

Our department has this ECG sensor, but the mobile application Kardia is not yet operational in Kyrgyzstan. The results of research work [3] demonstrate that smartphone applications that interact with medical devices provide an avenue for obtaining digital evidence from these devices. The recovered medical data (pdf, mp4) could be sent to our server.

3) **Olimex ECG/EMG shield** which allows Arduino like boards to capture Electrocardiography Electromiography signals [4]. The shield opens new possibilities to experiment with bio feedback. Our Android app saves the data received from the shield in the internal memory of the smartphone as a txt file, converts it to json format and sends it to the server.

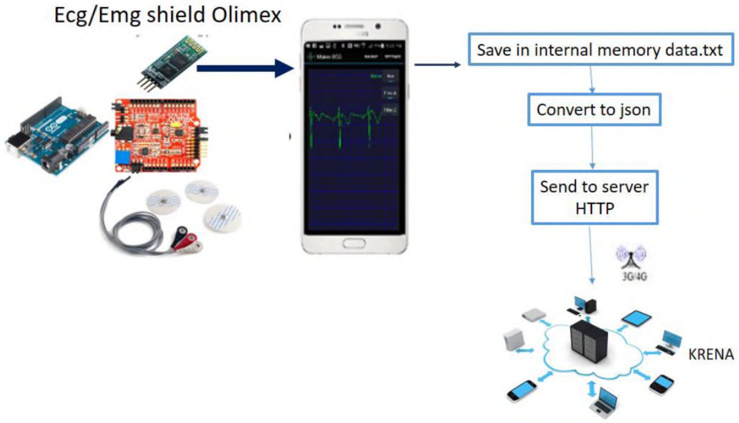


Figure 5. Olimex ECG/EMG shield

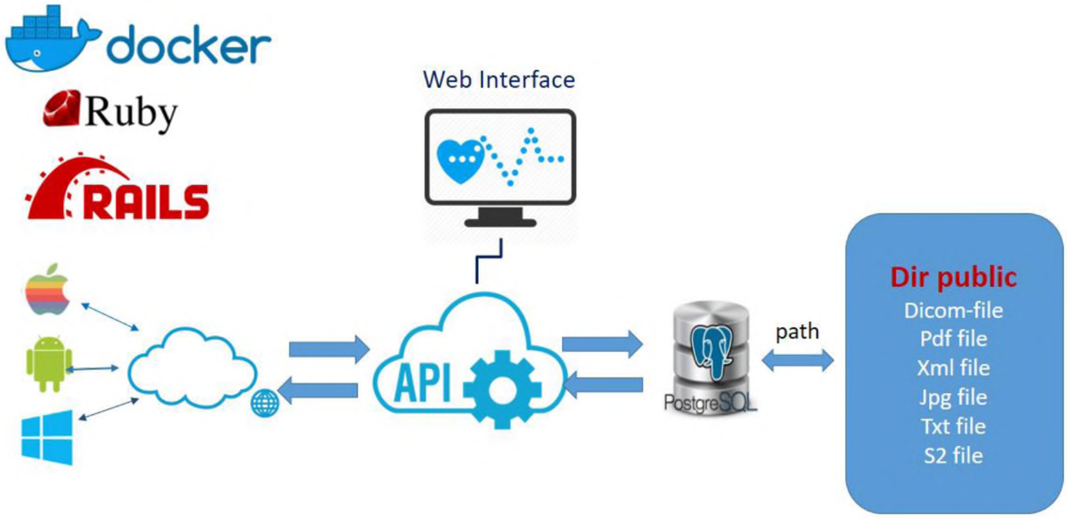


Figure 6. Architecture of the service

Communication

To receive and store data from the medical sensors, the RestFull server architecture is implemented on the server of Caren. The proposed service is realize RESTful architecture and is developed within Ruby on Rails Framework using Ruby language. The Database Management System is based on PostgreSQL and is responsible for data persistence. For the storage of the data, due to the dynamic and heterogeneity of medical data, we used the PostgreSQL databases. These databases are focused on dealing with high data volumes, with a good grade of scalability/flexibility and solving the traditional performances issues of relational databases with heterogeneous data. Patient medical files are stored in folder **Public** of our application, patient database contains a link to medical files.

The RESTful API implements the endpoint that allows Creating, Reading, Updating and Deleting remote resources and their relationships. It exposes APIs to help Android app communicate with the PostgreSQL database. RESTful is not dependent on any protocol and it makes use of existing well-known standards, such as HTTP, URI, JSON and XML.

Secure data exchange

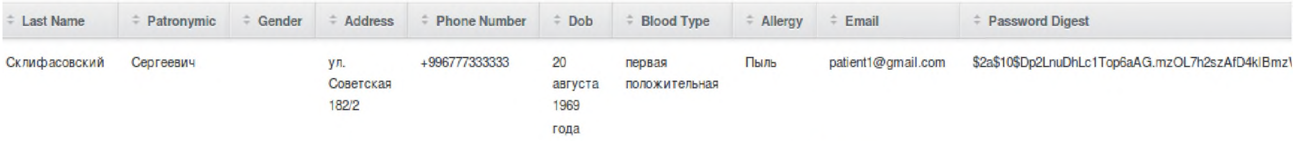
To protect medical data, we used the following solutions to support secure and reliable communication between the clients and server:

1) **HTTPS protocol**. HTTP is the standard protocol for client-server communication in the context of mobile apps and offers no security features. To ensure confidentiality and integrity of data sent through an untrusted medium, the Transport Layer Security (TLS) protocol with the protocol are used. TLS-secured HTTP connections are called HTTP Secure (HTTPS) connections. The KRENA association has a certified HTTPS protocol. Ruby on Rails Framework uses the force_ssl method in the controller to force the use of the HTTPS protocol. The SSL certificate and key are not handled by Rails but above it on a load balancer.

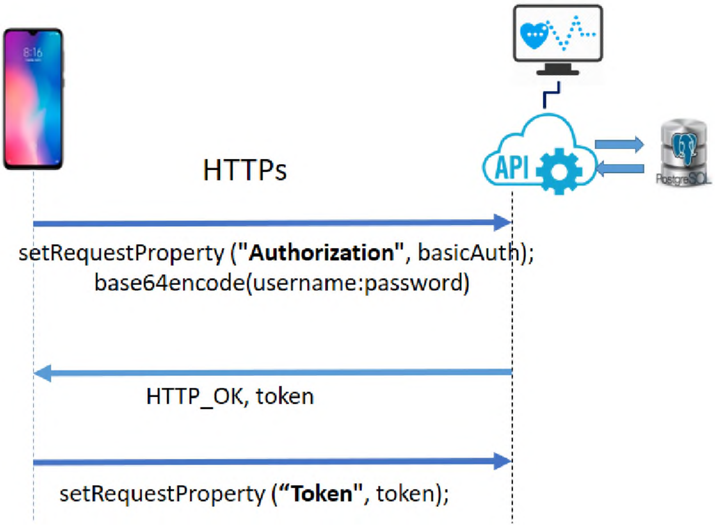
2) **Basic Authentication** - Rest client indicates its login and password to access the Rest service. The login and password are transmitted over the network as Base64 encoded text and can be easily decoded by any user. When using this method, it is advisable to use the HTTPs protocol.

In the application library «devise» deals with authentications. Devise is a flexible authentication solution for Rails. It is composed of 10 modules. For example, Database Authenticable — provides an opportunity to enter the system based on the encrypted and stored password in the database. Authentication is done by a POST request or using HTTP Basic Authentication. Devise uses «Bcrypt» to hash the password. «Bcrypt» library is used to encrypt and store passwords, «Bcrypt»-adaptive cryptographic hash function for key generation. Hash password is stored in the database instead of a valid password. During authentication, the hash of the user's password is compared with its hash in the database.

3) **Token Based Authentication** –after successful authentication **,** the server creates token with a secret and sends the it to the client. The client stores includes token in the header with every request. The server would then validate the token with every request from the client and sends response.



Last Name	Patronymic	Gender	Address	Phone Number	Dob	Blood Type	Allergy	Email	Password Digest
Склифасовский	Сергеевич		ул. Советская 182/2	+996777333333	20 августа 1969 года	первая положительная	Пыль	patient1@gmail.com	$2a$10$Dp2LnuDhLc1Top6aAG.mzOL7h2szAfD4kIBmz\



setRequestProperty ("**Authorization**", basicAuth);
base64encode(username:password)

setRequestProperty (“**Token**", token);

Figure 7. Token-based authentication

Token-based authentication is stateless - it does not store anything on the server but creates a unique encoded token that gets checked every time a request is made JSON Web Tokens carry information via JSON, is comprised of plain strings, so they can be easily exchanged in a URL or a HTTP header. Devise Token Auth- simple, multi-client and secure token-based authentication for Rails.

Authentication of a mobile application requires tokens, not cookies. Devise Token Auth gem updates tokens on every request and expires after a short time, so the application is protected. In addition, it maintains a session for each client / device.

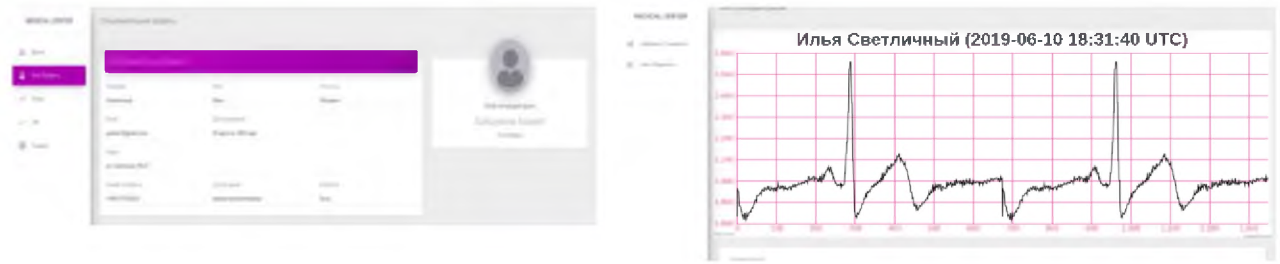


Figure 8. Web interface of the application

A web application has been developed, designed to provide real-time counseling and treatment assistance.

Conclusion

The tasks of integrating different ECG data formats into the server database were solved. Ready low-cost, portable, medical ECG devices available on the market have been reviewed. To receive and store data from the medical sensors, the RestFull server architecture is implemented on the server of Caren. Certified HTTPS protocol of the KRENA association was used for the interaction between server and mobile applications. Token-based cryptographic authentication and authorization systems are used.

REFERENCES

1. User's Manual Savvy, http://bodycontrolmt.cz/doc/UM-1.19.3-EN.pdf
2. MySignals eHealth and Medical IoT Development Platform Technical Guide
3. Grispos, George, William Bradley Glisson and Peter Cooper. "A Bleeding Digital Heart: Identifying Residual Data Generation from Smartphone Applications Interacting with Medical Devices." Proceedings of the 52nd Hawaii International Conference on System Sciences. 2019.
4. SHIELD-EKG-EMG bio-feedback shield USER'S MANUAL
5. OpenEMR https://www.open-emr.org/