An integrated eHealth solution for ECG examinations

G. D'AMATO, G. AVITABILE, G. COVIELLO
Dipartimento di Ingegneria Elettrica e dell'Informazione
Politecnico di Bari
Via Orabona n. 4, 70125 Bari
ITALY
giulio.damato@poliba.it

Abstract: - This paper details the building blocks of an integrated eHealth solution prototype for ECG examinations. The proposed solution is composed of a custom smartphone-based Bluetooth® electrocardiograph together with its companion app, and a set frontend and backend services deployed in the cloud allowing the definition of role-based collaborative workflows in a modern web-based environment. The aim of the discussion is to provide an insight on both the hardware and software design choices and trade-offs, as well as on the vision that inspired their implementation.

Key-Words: - eHealth; ECG; Electrocardiograph; Microcontroller; Bluetooth Low Energy; Smartphone; Cloud Computing; Web Technologies; Collaboration Tools

1 Introduction

Now more than ever, enterprise applications and IT strategies are integrating consumer technologies to implement solutions for easier and effective workflows on any device, a vision commonly referred to as "consumerization of IT" [1]. In fact, today's consumer applications are designed to deliver intuitive and engaging user experiences through modern web-based technologies, even before they reach the business market. As an example, social networks allow people to connect each other, share information and collaborate, three key actions even in modern enterprise processes.

Business performances can be enhanced with tools that deliver employees the same user-experience, participated environment and data fluidity they are used to at home. Traditional workflows based on sharing results of individual tasks, are now being replaced by real-time collaborative platforms designed over these trends.

Given this perspective, smart devices and web-based collaboration tools are expected to empower next-generation eHealth solutions. Previous research efforts reported the prototypal implementation of Bluetooth® electrocardiographs [1][3] and of cloud-based platforms for telemedicine [4][5]; however, few works investigated an integration of the two efforts or tried to take advantage of recent trends in consumer applications.

The main contribution of this research is the proposal of an integrated eHealth solution for ECG examinations that takes advantage of smart devices and modern web-based technologies. This has been done through: i) the implementation of a cheap, compact and lightweight smartphone-based Bluetooth® electrocardiograph; ii) the development of a *compation app* for smartphone and tablet devices; iii) the cloud deployment of a modern web application and of its underlying backend infrastructure.

2 Bluetooth® electrocardiograph

The first step towards the proposed solution prototype has been the design of a custom smartphone-based Bluetooth® electrocardiograph capable of 12-lead acquisitions. The electrocardiograph is based on a fully integrated analog frontend, a microcontroller and a Bluetooth® Low Energy (BLE) transceiver (Fig. 1).

The device has been developed for medical grade performances, but trying to minimize cost, size and weight. For this reason, being a smartphone-based device, the electrocardiograph intentionally lacks a hardware user interface, apart from a power button and a status LED, a choice that can also be exploited to enhance robustness in though environments.

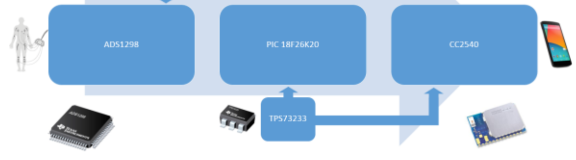

Fig. 1 – Bluetooth® electrocardiograph block diagram

The selected analog frontend is the TI's ADS1298. The IC complies with the AMI EC11, EC13, IEC60601-1, IEC60601-2-27 and IEC60601-2-51 standards; integrates eight simultaneous sampling delta-sigma ($\Delta\Sigma$) Analog-to-Digital

Converters (ADCs) with 24-bit resolution and eight Programmable Gain Amplifiers (PGAs); implements, among the others, the Right Leg Driver (RLD) and the Wilson Center Terminal (WCT) circuits.

A PIC 18F26K20 microcontroller runs the application firmware, a software component, written in C programming language, which configures the biopotentials' acquisition, collects digitalized samples and forwards them via a third-party BLE transceiver (based on TI's CC2540). Data exchange between the microcontroller and the ADS1298 is performed through SPI, with an interrupt-driven samples' acquisition. Instead, data exchange between the microcontroller and the BLE transceiver is performed through UART.

The analog and the digital section of the apparatus are configured for single supply operation (3.3V, 0V), a popular choice for battery-operated devices, though two separate voltage regulators are used to preserve noise characteristics. A third-party wireless-charging module, based on the Qi standard [6], has been integrated for future investigations (e.g. seal the apparatus to improve its robustness in tough environments).

The Printed Circuit Board (PCB) of the proposed apparatus (Fig. 2) (Fig. 3) (Fig. 4) measures $5x5cm^2$ and has been enclosed in a convenient plastic case. An adhesive NFC tag has been glued to one of the internal faces of the case and has been programmed to let the user pair with the apparatus when in close proximity, a feature that is commonly referred to as *tap-to-pair* [7].

The NFC payload is a non-NDEF formatted string including, among other data and metadata, device name and PIN code; actual pairing is performed through a proprietary evaluation of the above payload. The apparatus has a DB15 connector to allow connection for 10-lead ECG cables (HP, Philips and Agilent compatible) (Table 1).

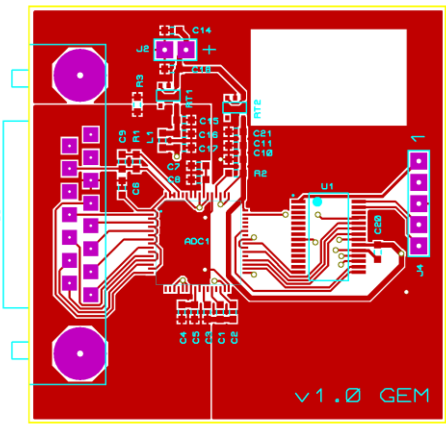

Fig. 2 – Actual layout of the apparatus (top layer)

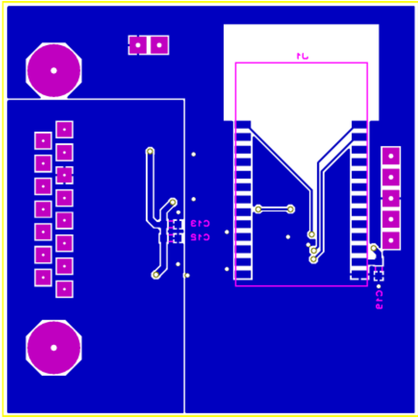

Fig. 3 – Actual layout of the apparatus (bottom layer)

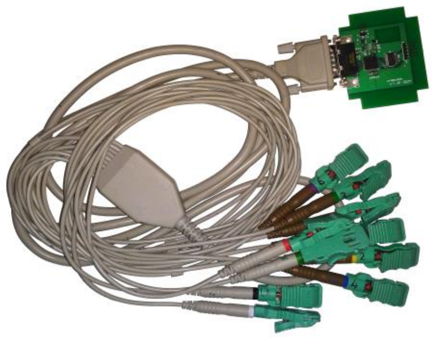

Fig. 4 – Actual prototype of the apparatus

ECG Electrode	DB15 Pin
V2	1
V3	2
V4	3
V5	4
V6	5
AVSS	6
RA	9
LA	10
LL	11
V1	12
RLD	14

Table 1 – 10-lead ECG cable pinout

The BLE transceiver has been configured to act as a *UART to BLE bridge*, emulating a Serial Port Profile (SPP) at a data rate of 115.200bps. A proprietary protocol, based on a binary packet exchange, has been implemented to ensure that a consistent stream of samples is received and interpreted, and that communication errors could be identified and, possibly, corrected [8] (Fig. 5). Given the above data rate and protocol, it is possible to calculate N, the maximum allowed sample rate:

$$\left(2+\frac{24}{8}\cdot N_{CH}\right)\cdot N+5\left(\frac{N}{256}\right)<\frac{115.200}{8}$$

where N_{CH} is the number of simultaneous ECG channels to be transmitted (Fig. 6).

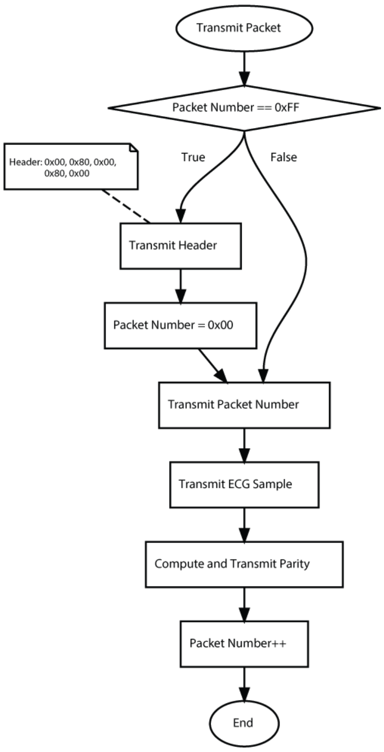

Fig. 5 – Transmit Packet routine's flowchart

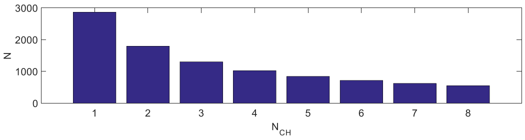

Fig. 6 – Maximum allowed ECG sample rate.

The selected analog frontend forms and samples eight ECG leads (Lead I to Lead II and Lead V1 to Lead V6). The remaining four leads are computed using the following formula (Table 2): the simultaneous transmission of two ECG channels is always sufficient to delegate calculations to the companion app.

Lead	Formula
Lead III	Lead II - Lead I
Lead aVR	- Lead II + 0.5 * Lead III
Lead aVL	Lead I - 0.5 * Lead II
Lead aVF	Lead III + 0.5 * Lead I

Table 2 – Computed leads formula

A MATLAB® script capable to test the whole electrocardiograph has been developed. The script configures the apparatus, and acquires and plots an internal test signal on channel 1 at a rate of 500sps; two timed loops perform the latter two tasks: the acquisition loop is executed every 100ms, the plot loop every 500ms. The acquisition loop (Fig. 7) is responsible of: i) reading the serial stream; ii) recomposing the incoming packets; iii) populating the sample buffer. The plot loop is responsible of updating a figure with a convenient set of the incoming data.

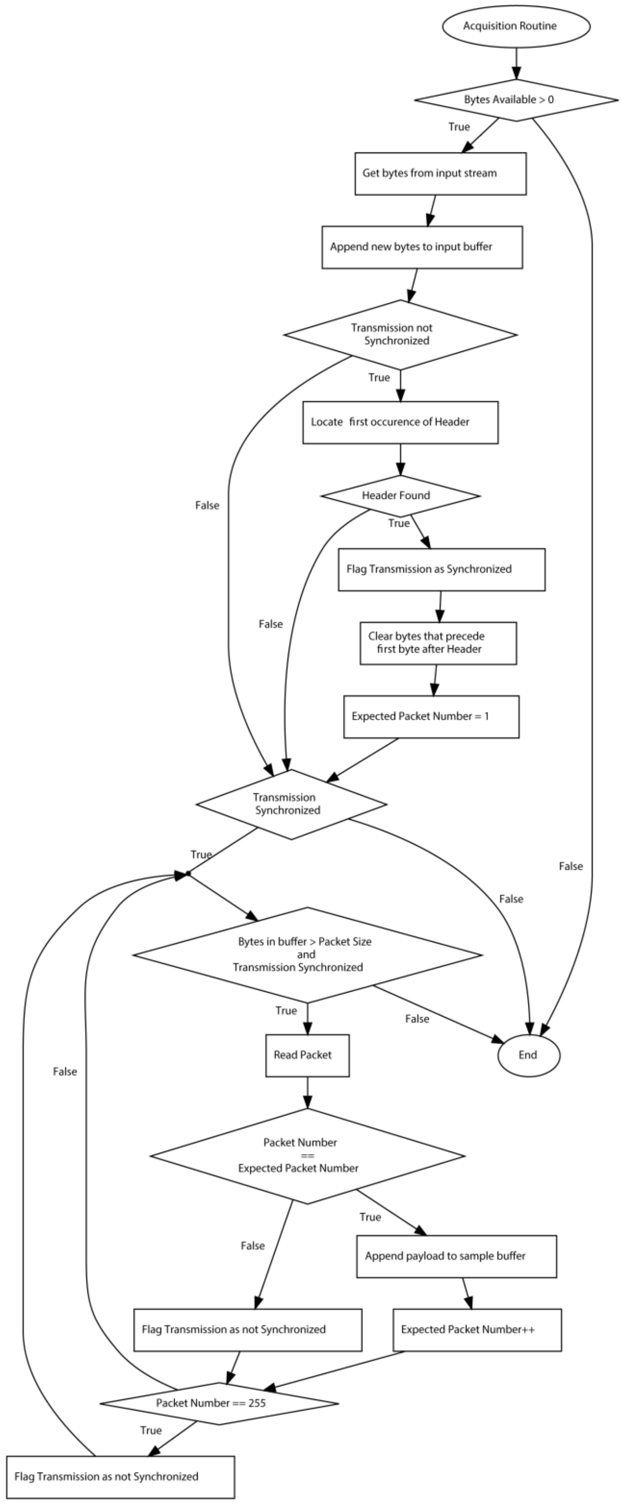

Fig. 7 – Acquisition routine's flowchart

3 Companion app

The companion app for smartphone and tablet devices is a key software subsystem for the acquisition of ECG signals and for the initiation of teleconsultation flows.

The app constitutes the main user interface towards the hardware; it is capable of: i) configuring the apparatus for acquisition; ii) displaying the digitized ECG samples on screen in real-time; iii) saving or sharing the signals in the cloud. Moreover, the app conveniently processes the incoming signals through a cascade of three digital filters (high-pass, low-pass and notch). In fact, the hardware and firmware subsystem is intentionally lacking this feature, thus relying, for this task, on the compute power of the *host* smart device; though, the implications of this tradeoff may further be investigated.

The app's interaction flow has been designed to minimize the number of steps separating initialization from acquisition. The main menu allows registered users to log into the cloud platform or to immediately browse for Bluetooth® devices in pairing mode (Fig. 8); if an already paired electrocardiograph is ready for connection, the app automatically shows up a convenient shortcut dialog box. After this step, the hardware, firmware and software subsystems are ready to perform the biopotentials' acquisition; additionally, a step by step tutorial mode can be activated, even assisting the correct electrode placement. Eventually, actual acquisition is performed, and the user is prompted a dialog box that allows to save further details about the exam (e.g. patient's name, age, …); then, the user can store the exam locally or in a remote workspace. As soon as an exam reaches the cloud, it becomes available for its further classification (e.g. it can be assigned to an existing patient's record) and in teleconsultation flows through web-based collaboration tools.

Current operative scenario is oriented towards acquisitions characterized by a limited duration; however, the Bluetooth® electrocardiograph could even operate as a Holter monitor if paired to a different software subsystem. This unprecedented modularity of the above discussed components proves that modern apparatuses, whose functioning is based on smartphone and tablet devices, can interestingly take advantage of updatable apps to adapt to not foreseen use cases or to provide newer features. Being the app the main user interface towards the hardware, further variants and customizations can also be implemented (e.g. less or more controls for beginner or advanced users, overlay captions, graphical themes, …).

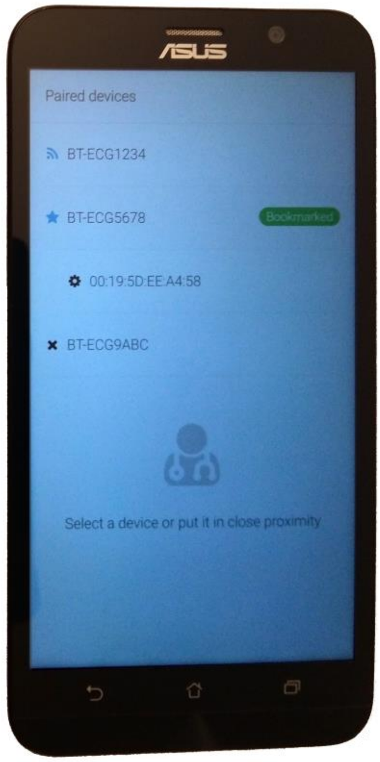

Fig. 8 – Actual graphical user interface (paired devices)

Two concurrent threads, that replicates the aforementioned MATLAB® script's behavior, manage the biopotentials' acquisition; however, in this case, the plot loop is also responsible of the digital signal processing. The signal processing class implements filters in Direct Form II (Fig. 9) and has been realized in managed code. A native code reimplementation will be investigated to increase throughput [9] and support a higher number of filter's coefficients.

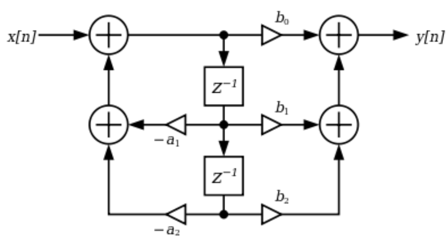

Fig. 9 – Direct Form II flow graph of a 2nd-order filter.

The normalized frequency response that has been implemented is represented in (Fig. 10). A typical diagnostic ECG requires the undistorted acquisition of frequency components in the 0.05–150Hz range [10][11], though further information exists beyond these limits. In this particular implementation, the high-pass filter's cut-off frequency is 0.01Hz and the low-pass filter's cut-off frequency is 150Hz; a notch filter centered at 50Hz has also been implemented to remove powerline interference. Furthermore, a sample rate of 500sps has been selected; it can be considered as acceptable, though higher sample rates can be implemented according to the former discussion.

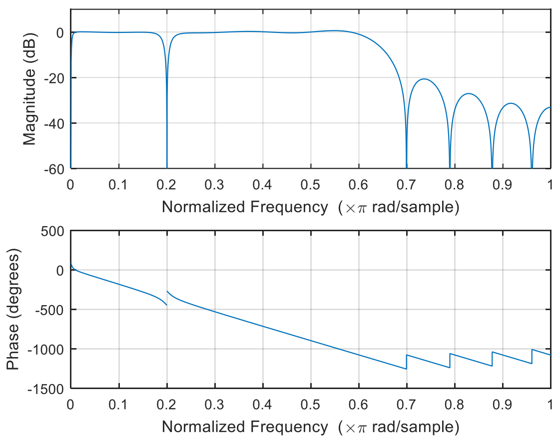

Fig. 10 - Normalized frequency response of the high-pass, low-pass and notch filters' cascade

The discrete-time high-pass filter's (HPF) transfer function is:

$$H_{HPF}(z) = \frac{1 - 2\,z^{-1} + 1\,z^{-2}}{1 - 1.999822284\,z^{-1} + 0.9998223\,z^{-1}}$$

The discrete-time notch filter's (NF) transfer function is:

$$H_{NF}(z) = \frac{0.963 - 1.558z^{-1} + 0.962z^{-2}}{1 - 1.558\,z^{-1} + 0.926\,z^{-2}}$$

The discrete-time low-pass filter's (LPF) transfer function is represented in (Fig. 11).

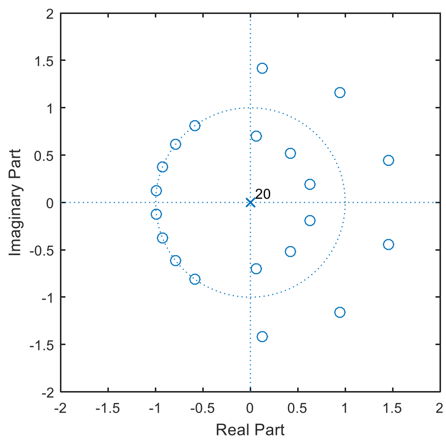

Fig. 11 – Zero-pole plot of the LPF

$H_{HPF}(z)$, $H_{NF}(z)$ and $H_{LPF}(z)$ are stable digital filters since, by design, all their poles lay inside the unit circle. However, it seems important to remark that particular care should be taken when implementing their coefficients, since coarse numerical representations may easily lead to considerable deviations in the frequency response or to potential instability. A CAL30000 beat type has also been constructed (Table 3) and processed. The output signal is essentially indistinguishable from the input, apart from a time delay of about ten samples. A more rigorous conformity assessment of the achieved performances shall be conducted according to international standards; the current design should be considered as a proof of concept and substantial revisions may be necessary to be ready for certification.

Parameter	Type	Value
P	Amplitude (µV)	150
Q	Amplitude (µV)	0
R	Amplitude (µV)	3000
S	Amplitude (µV)	-3000
ST	Amplitude (µV)	0
T	Amplitude (µV)	600
P	Interval/Duration (ms)	114
PQ	Interval/Duration (ms)	177
QRS	Interval/Duration (ms)	100
QT	Interval/Duration (ms)	398
Q	Interval/Duration (ms)	0
R	Interval/Duration (ms)	50
S	Interval/Duration (ms)	50

Table 3 – CAL30000 Reference Values

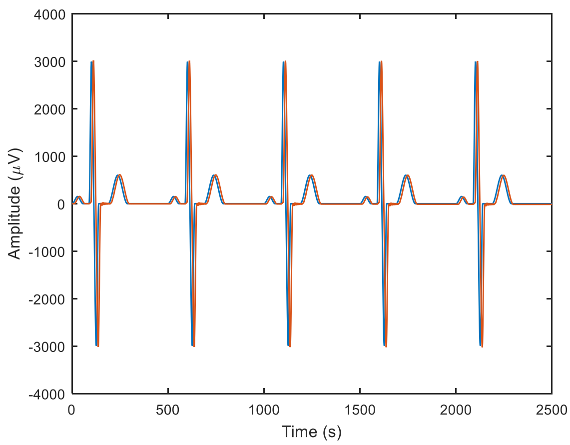

Fig. 12 – CAL30000 signal, before (blue) and after (red) processing

An ECG acquisition from a patient simulator is depicted in (Fig. 13). A standard ECG acquisition (Fig. 14) from an actual person has also been conducted; the Lead I trace has been imported in MATLAB® and is depicted in (Fig. 15).

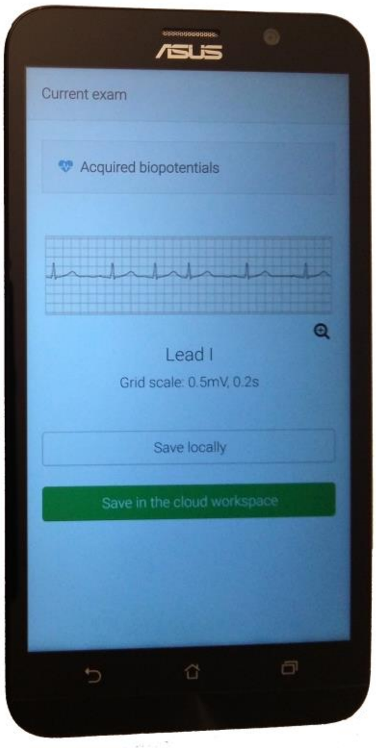

Fig. 13 – Actual graphical user interface (current exam)

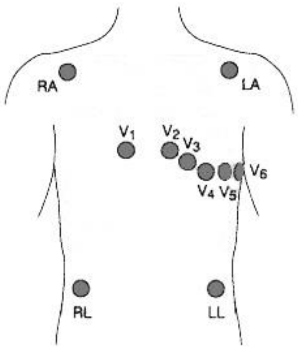

Fig. 14 – 12-Lead ECG standard electrode placement

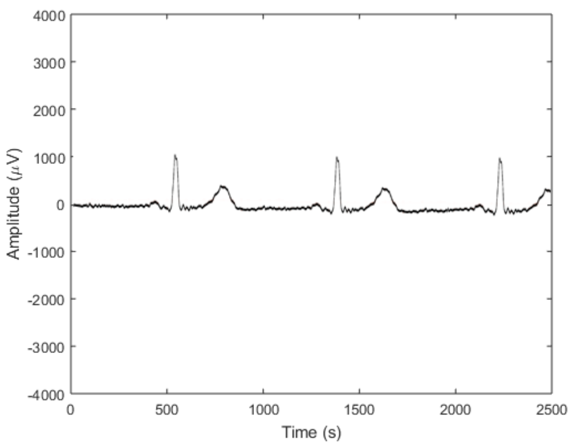

Fig. 15 – Lead I acquired from an actual person

4 Cloud services

A web-based collection of collaboration tools implements the remote platform services empowering teleconsultation flows. The overall system and software architecture (Fig. 16) comprises: i) a fault-tolerant Distributed Filesystem (DFS) based on a NoSQL Database Management System (DBMS); ii) a Relational DBMS (RDBMS); iii) an Identity Manager (IDM); iv) a RESTful API towards data; v) a Policy Enforcement Point (PEP); vi) a publish/subscribe message broker; vii) an integrated web application.

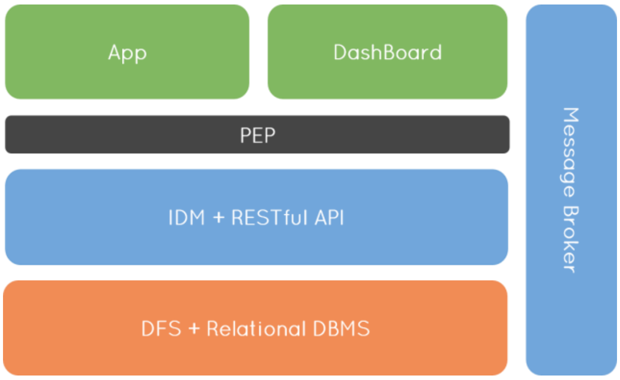

Fig. 16 – System and software architecture's diagram

In this particular implementation, the DFS is based on MongoDB, and is responsible for the acquired biopotentials' persistence; relevant metadata are stored along with actual samples, so as each document implements a self-contained piece of information. The RDBMS is based on PostgreSQL, and is responsible for structured data persistence. The IDM and PEP are based on OAuth 2.0, and are responsible for authentication and authorization procedures. The RESTful API is based on the Slim Framework; it implements the endpoint that allows Creating, Reading, Updating and Deleting (CRUD) remote resources and their relationships.

On top of the aforementioned components, a web application has been implemented, namely a platform component developed in HTML5, CSS3 and JavaScript (using jQuery and Bootstrap frameworks) that allows users to operate on private and shared workspaces from any modern web browser (Fig. 17).

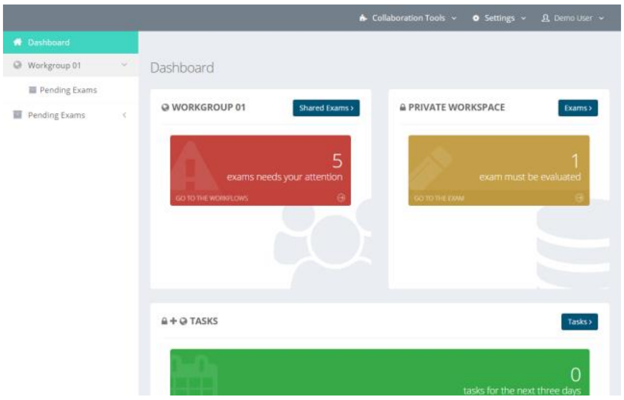

Fig. 17 – Actual web application (user's dashboard)

By logging into the cloud platform, registered users can: i) manage their own patients; ii) assign exams; iii) provide diagnoses or leave notes. Users can also join workgroups and collaborate with other professionals; under this scenario, a number of users can: i) manage a shared directory of patients; ii) operate on shared exams; iii) leave notes towards collaborative diagnoses. Moreover, the workgroup's owner can assign roles and define workflows, namely predefined procedures with role-based tasks.

An event-driven message broker for WebSockets, based on Node.js [12]: i) orchestrates teleconsultation flows; ii) allows users to receive notifications; iii) synchronizes client-side collections. The message broker asynchronously pushes notifications, towards any subscribed instance of the web application, about relevant CUD operations performed on observed data. Messages are JSON encoded objects that include: i) the name of the queried collection; ii) the preformed action; iii) the involved resource.

As an example, when a user flags a task as completed, a message (Code 1) is forwarded to any other workgourp's member involved in the procedure. In this particular implementation, the web application would react to the new message: i) updating its cache; ii) generating and showing a text to notify the end user (Fig. 18).

```
{
  collection: "tasks",
  action: "UPDATE",
  resource:
  {
    id: 123,
    workflowId: 456,
    status: "COMPLETED",
    comment: "..."
  }
}
```

Code 1 – Example of a message

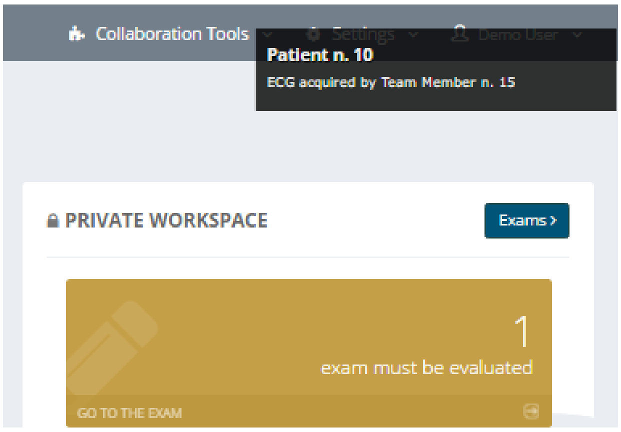

Fig. 18 – Actual web application ("on message" notification)

The above services have been deployed among three CentOS 7 virtual machines, namely a Web Application Server (for message broker and web application), a Web API Server (for authentication, authorization and RESTful endpoint) and a Database Machine (for DFS and RDBMS).

5 Conclusion

Modern eHealth solutions can take advantage of consumer technologies to implement easier and effective workflows to enhance performances in diagnostic processes.

Smartphone-based diagnostic devices, together with their companion apps, can act as convenient entry points for data towards hospital information systems. In fact, this new class of diagnostic devices allows to seamlessly produce, share and consume data in a wide range of clinical use cases.

Modern web-based technologies can be used to implement frontend and backend services that allow healthcare professionals to connect each other, share information and collaborate towards diagnoses. Information systems in healthcare should be designed to deliver the same user-experience, participated environment and data fluidity that characterize successful consumer applications.

The above considerations have been the starting point of this work, which has focused over the development of both a prototypal smartphone-based diagnostic device and a web-based information system, with a particular interest towards their implications in an innovative workflow design.

The key finding of this research is that future diagnostic devices can take advantage of the capabilities of smartphones and tablets to achieve cheapness, compactness and lightness, while delivering improved interaction schemes and functionalities, such as the opportunity to *connect* to web-based information systems in healthcare. Moreover, modern integrated eHealth solutions can be used to shape agile diagnostic processes through pleasant user interfaces towards data and real-time collaboration tools.

Our future research effort is focused towards: i) power efficiency for longer battery-life (e.g. improving the proprietary protocol); ii) higher sample rate (e.g. introducing compression); iii) in-firmware vs. in-app digital signal processing impact on overall system performances; iv) security of the implemented cloud services (e.g. assessment on the best practices employed in state of the art eHealth solution); v) cloud services interoperability with existing platforms and standard protocols.

It seems important to remark that, regardless of the underlying technological effort, the deployment of effective eHealth platforms eventually depends on cultural acceptance, and thus a careful multi-disciplinary and multi-professional co-design stage shall always be conducted.

References:

[1] S, Köffer, K. C. Ortbach and B. Niehaves, "Exploring the Relationship between IT Consumerization and Job Performance: A Theoretical Framework for Future Research," *Communications of the Association for Information Systems*, 2014, Vol. 35, Article 14.

[2] B. Yu, L. Xu and Y. Li, "Bluetooth Low Energy (BLE) based mobile electrocardiogram monitoring system," *Information and Automation (ICIA), 2012 International Conference on*, Shenyang, 2012, pp. 763-767.

[3] S. E. de Lucena, D. J. B. S. Sampaio, B. Mall, M. Meyer, M. A. Burkart and F. V. Keller, "ECG monitoring using Android mobile phone and Bluetooth," *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, Pisa, 2015, pp. 1976-1980.

[4] S. Ahmed and A. Abdullah, "Telemedicine in a cloud — A review," *Computers & Informatics (ISCI), 2011 IEEE Symposium on*, Kuala Lumpur, 2011, pp. 776-781.

[5] Z. Jin and Y. Chen, "Telemedicine in the Cloud Era: Prospects and Challenges," in *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 54-61, Jan.-Mar. 2015.

[6] D. van Wageningen and T. Staring, "The Qi wireless power standard," *Power Electronics and Motion Control Conference (EPE/PEMC), 2010 14th International*, Ohrid, 2010, pp. S15-25-S15-32.

[7] J. Morak, H. Kumpusch, D. Hayn, R. Modre-Osprian and G. Schreier, "Design and Evaluation of a Telemonitoring Concept Based on NFC-Enabled Mobile Phones and Sensor Devices," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, pp. 17-23, Jan. 2012.

[8] Texas Instrument, "ECG Implementation on the TMS320C5515 DSP Medical Development Kit (MDK) with the ADS1298 ECG-FE," *Application Report*, SPRABJ1, Jan. 2011.

[9] C. M. Lin, J. H. Lin, C. R. Dow and C. M. Wen, "Benchmark Dalvik and Native Code for Android System," *Innovations in Bio-inspired Computing and Applications (IBICA), 2011 Second International Conference on*, Shenzhan, 2011, pp. 320-323.

[10] P. Kligfield, et al. "Recommendations for the standardization and interpretation of the electrocardiogram: part I: the electrocardiogram and its technology a scientific statement from the American Heart Association Electrocardiography and Arrhythmias Committee, Council on Clinical Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society endorsed by the International Society for Computerized Electrocardiology." *Journal of the American College of Cardiology* 49.10 (2007): 1109-1127.

[11] G. D. Clifford, F. Azuaje; P. McSharry, "ECG statistics, noise, artifacts, and missing data. Advanced Methods and Tools for ECG Data Analysis," 2006, 6:18.

[12] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," in *IEEE Internet Computing*, vol. 14, no. 6, pp. 80-83, Nov.-Dec. 2010.