electronics

Article

Building Standardized and Secure Mobile Health Services Based on Social Media

Jesús D. Trigo [1,*], Óscar J. Rubio [2], Miguel Martínez-Espronceda [1], Álvaro Alesanco [3], José García [3] and Luis Serrano-Arriezu [1]

[1] Department of Electrical, Electronic and Communications Engineering, Public University of Navarra, Institute of Smart Cities (ISC), Navarra Institute for Health Research (IdiSNA), 31006 Pamplona, Spain; miguel.martinezdeespronceda@unavarra.es (M.M.-E.); lserrano@unavarra.es (L.S.-A.)

[2] Everis Assets, 50009 Zaragoza, Spain; orubimar@everis.com

[3] Department of Electronics Engineering and Communications, University of Zaragoza, Aragón Institute of Engineering Research (I3A), 50018 Zaragoza, Spain; alesanco@unizar.es (Á.A.); jogarmo@unizar.es (J.G.)

* Correspondence: jesusdaniel.trigo@unavarra.es

Received: 20 October 2020; Accepted: 18 December 2020; Published: 21 December 2020

Abstract: Mobile devices and social media have been used to create empowering healthcare services. However, privacy and security concerns remain. Furthermore, the integration of interoperability biomedical standards is a strategic feature. Thus, the objective of this paper is to build enhanced healthcare services by merging all these components. Methodologically, the current mobile health telemonitoring architectures and their limitations are described, leading to the identification of new potentialities for a novel architecture. As a result, a standardized, secure/private, social-media-based mobile health architecture has been proposed and discussed. Additionally, a technical proof-of-concept (two Android applications) has been developed by selecting a social media (Twitter), a security envelope (open Pretty Good Privacy (openPGP)), a standard (Health Level 7 (HL7)) and an information-embedding algorithm (modifying the transparency channel, with two versions). The tests performed included a small-scale and a boundary scenario. For the former, two sizes of images were tested; for the latter, the two versions of the embedding algorithm were tested. The results show that the system is fast enough (less than 1 s) for most mHealth telemonitoring services. The architecture provides users with friendly (images shared via social media), straightforward (fast and inexpensive), secure/private and interoperable mHealth services.

Keywords: mHealth; privacy; security; social media; standardization

1. Introduction

In the last decade, the usage of mobile and smartphones has grown exponentially. According to a mid-2019 report [1], the vast majority of Americans (96%) own a cellphone of some kind, while the share of Americans that own smartphones is 81%. This fact has spurred the scenario of mobile health (also referred to as mHealth), which, as stated by the World Health Organization, is the "medical and public health practice supported by mobile devices" [2]. The coverage of mHealth includes several aspects, such as the acquisition, manipulation, classification and transmission of health-related information [3]. Thus, the users become the center of the action, being producers and owners of their own biomedical data and signals, which can be ubiquitously gathered with their own personal health devices and transmitted by means of their personal mobile devices. Furthermore, the users can analyze their biomedical data locally and share them with both formal and informal caregivers. Ultimately, such data can be either stored in personal data vaults for future consultations or sent to other systems or services, depending on the mHealth application.

All said, although the feasibility and availability of traditional telemonitoring mHealth services have been thoroughly described in the literature [4,5], there are still challenges to be solved, mainly related to security and privacy concerns. Moreover, the fixed structures of such centralized architectures traditionally promoted by manufacturers could lead to lack of engagement, motivation or connections among their users, who may decide to create their own solutions [6]. Indeed, many patients and communities today are demanding to become co-producers and, in the end, holders of their own data. If they do not get to have control, they may take over and create the necessary tools to gain self-empowerment themselves. An epitome thereof is the Nightscout movement [7], an open-source, Do-It-Yourself project allowing real-time access to a glucose monitor, as well as data persistence in the cloud. This project can be seen as an illustration of the difference between "patient-centered care" and "real patient controlled care", where it is the patients, the relatives and the communities they live in who finally "make sense" of things, define the goals and stress coherence [8].

Although the Nightscout project has provided valuable lessons for the new paradigm of empowered citizens, one logical step forward is the use of social-media-based health systems, since they could foster engagement, empowerment and community building [9]. Besides the predominantly ludic character of social media, new uses in different domains are being investigated and developed nowadays. They are driven by the attracting features of social media as well as their remarkable mass of users (e.g., as of September, 2020, Facebook claimed to have 3 billion users [10], which represent nearly one-third of the world's population). Reciprocally, attracting new users to social media, e.g., those coming from mHealth scenarios, would help social media to reaffirm their leading position in today's internet panorama. Indeed, social media provide a wide variety of tools that enable users to build communities around them where they can create, share and exchange information in different formats [11]. Ever since the social media appeared, the idea of combining social media and healthcare has gained momentum. Studies have found that healthcare organizations, clinicians, patients and regulatory bodies could benefit from the use of social media [12,13]. However, in spite of the promising benefits of social-media-based healthcare, there are some challenges still to be solved. Most of those issues are linked to privacy and security concerns, but there are also open questions about usability, manipulation of identity, governance or confidentiality, along with the aforementioned demotivation [9,14,15].

As it has been marked in the literature, there is limited evidence related to the efficacy and effectiveness of social media in healthcare [9]. Most projects combining healthcare and social media so far use data mining to analyze shared health-related data and extract valuable information [16–18]. To date, however, there is little effort in the literature towards social-media-based mHealth systems where biomedical data are sent through social media while taking into account security and standardization. An example that partially accomplish such a goal was presented in [19]. This system leverages Twitter to send the main data to a back-end repository using a web based approach, while offering the possibility of sharing health-related messages constructed according to particular status descriptors using a medical nomenclature. Nonetheless, as regards to privacy and security, they only make use of the built-in security policies implemented by Twitter (open authentication and private lists). Related to this, the authors presented a technical proof-of-concept system for following up cardiovascular patients using Twitter and Health Level 7 (HL7) [20], which can be considered an improvement thereof. Nevertheless, no further security and privacy measures were implemented.

Thus, adequate protection policies shall be implemented in social-media-based, mHealth applications to achieve security and privacy levels in line with the demands of users and the regulations applicable. Examples of such regulations are the Health Insurance Portability and Accountability Act (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in Europe, effective since 2018. The use of HIPAA as a means for achieving standardized data security and privacy in mHealth scenarios has already been proposed in the literature [21]. A common objective of these regulations is to guarantee Information Assurance and Security (IAS). Traditionally, confidentiality, integrity and availability—also referred to as the CIA-triad—were the elements modelling IAS. Nowadays, the

CIA-triad has evolved to a more comprehensive IAS-octave (confidentiality, integrity, availability, accountability, auditability, authenticity, non-repudiation and privacy) [22]. Such regulations also enforce prevention and reaction to data breaches as well as responsibility and sanctions to those that do not thoroughly address the aforementioned measures. Social media services must comply with regulatory requirements of the countries they are working in. However, uploading unprotected data to social media could sow suspicion or mistrust among users. Therefore, the design and implementation of an additional robust security layer, being complementary to the measures already implemented by social media, would help to guarantee independence from their privacy policies and raise the trust of the potential users.

Additionally, it is true that any ad-hoc secure, private, social-media-based, mHealth solution is able to exchange biomedical information without the need of a common information model. However, this may not suffice if pervasive, distributed, integrated biomedical ecosystems are to be achieved [23]. In order to accomplish at least semantic interoperability—according to the model proposed by Turnitsa et al. [24]—the content of the information exchange requests must be unambiguously defined. Hence, biomedical interoperability standards are highly recommended. Within the healthcare domain, initiatives like HL7 or Digital Imaging and Communication in Medicine (DICOM) are robust, widespread examples of this standardization effort.

The elements presented in the paragraphs above—to wit, mHealth, social media, security/privacy and standardization—are usually treated as separate, or, at least, loosely integrated fields. The existing literature offers preliminary examples describing systems that partially cover some of these fields combined, being the most complete so far the work conducted by Triantafyllidis et al. [19] and a previous work by the authors [20]. Thus, the underlying hypothesis of this paper is that all these components could be seamlessly merged to build enhanced healthcare services (this is covered in Section 2). As a result, the main objective of this work is to propose a generic architecture for building standardized, secure, private, social-media-based mHealth services (Section 3.1). For the sake of simplicity and convenience, we introduce here the acronym mH3S (after mHealth, standardized, secure and social). Secondly, a proof of concept of the proposed mH3S architecture will be also presented (Section 3.2). It is composed of (a) Twitter as the social media, (b) version 2 of HL7 as a means for interoperability, (c) openPGP as security envelope and (d) a particular embedding algorithm. Third, both the generic architecture proposed and the technical proof-of-concept implementation are discussed in Section 4. Conclusions are drawn in Section 5. Table 1 provides a list of the acronyms used throughout the paper.

Table 1. Acronyms and their meaning.

Acronym	Meaning
AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certification Authority
CIA	Confidentiality, Integrity and Availability
CDSS	Clinical Decision Support Systems
CMS	Cryptographic Message Syntax
DICOM	Digital Imaging and Communication in Medicine
ECG	ElectroCardioGram
EHR	Electronic Health Record
GDPR	General Data Protection Regulation
GZIP	GNU ZIP
HIPAA	Health Insurance Portability and Accountability Act
HIS	Health Information System
HL7	Health Level 7
HS	Host System
HTTPS	Hypertext Transfer Protocol Secure
IAS	Information Assurance and Security
IEEE	Institute of Electrical and Electronics Engineers

Table 1. *Cont.*

Acronym	Meaning
IETF	Internet Engineering Task Force
IHE	Integrating the Healthcare Enterprise
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
LOINC	Logical Observation Identifiers Names and Codes
mH3S	Standardized, secure/private, social-media-based mobile health architecture
OAuth	Open Authorization
openPGP	open Pretty Good Privacy
ORU	Observation Result Unsolicited
PHD	Personal Health Device
PHR	Personal Health Record
PKI	Public Key Infrastructure
PNG	Portable Network Graphics
QR	Quick Response
RGBA/ARGB	Red Green Blue + Alpha
RSA	Rivest, Shamir and Adleman
S/MIME	Secure/Multipurpose Internet Mail Extensions
SCP-ECG	Standard Communications Protocol for computer assisted ElectroCardioGraphy
SNOMED-CT	Systematized Nomenclature of Medicine—Clinical Terms
TLS	Transport Layer Security
TPM	Trusted Platform Modules
UCUM	Unified Code for Units of Measure
UMLS	Unified Medical Language System

2. Materials and Methods

The methodology proposed in this paper consists of two steps. First, the current mHealth architectures—focused on telemonitoring architectures—are illustrated, covering the traditional usage and their evolution. Second, the limitations of the architectures composing such evolution are described through a comprehensive review of the existing literature, which ultimately will lead to the identification of new potentialities for a novel approach.

2.1. Analysis of Traditional Telemonitoring mHealth Architectures

A generic mHealth architecture facilitates the implementation of mHealth applications, usually grouped in three major—and interrelated—fields: health and fitness, independent living and disease management. Although other mHealth applications are possible—e.g., medical reference, nutrition or wellness applications–, the scope of this paper is focused in applications that are able to report medical status updates to formal or informal caregivers. Such applications demand a reliable and efficient acquisition of personal biomedical information, its adequate storage and a pervasive, ubiquitous and controlled access to the users that need to consult this information.

To cope with the requirements of the health scenarios described above, different mHealth architectures have been proposed and developed in the literature [4,5]. Reference [4], published in 2015, reviews mHealth services and shows a typical architecture thereof. Reference [5] performs a survey on the architectures of telemonitoring research projects in 2014 and, from that knowledge, derives a common architecture of telemonitoring systems, which can be summarized in three distinct tiers: sensors, gateway and remote server.

Based on such reviews, a generic mHealth architecture is illustrated in Figure 1, which is further detailed as follows. The most basic end-to-end mHealth architecture is comprised by two elements. First, a Personal Health Device (PHD) or, more generically, a sensing unit, which collects and sends the user's biomedical information. Second, a Host System (HS), which stores the collected information, for example a Health Information System (HIS) with an Electronic Health Record (EHR) or a Personal Health Record (PHR), the latter also referred to as personal data vaults. In addition, there are usually several PHDs around the patient/user, and they seldom have the connectivity to reach the HS—to date, few PHDs are Internet-ready, although the paradigm may be shifting due to the Internet of Medical Things [25]. Thus, most mHealth architectures today include a third element, namely the concentrator device, a mobile device, e.g., cell phone or tablet, which gathers the biomedical data from the different PHDs and forwards them to the HS. Furthermore, depending on the intended mHealth application, various other elements can be incorporated into the end-to-end architecture. For example, service providers and medical systems would be placed before and after the data arrive at the HS. Examples of medical systems are alarm systems or Clinical Decision Support Systems (CDSS). They would perform different operations included in the scope of the HS, such as the management, monitoring, processing or follow-up the user's biomedical information. Moreover, other elements can connect with the HS to either share medical information, such as third-party host systems, or access that information, such as a consultation systems, thereby interfacing the caregivers and the users with the HS (see Figure 1).

As regards to the persons involved, generally, up to four types can be distinguished in a traditional mHealth architecture:

- The patients or users: They record the biomedical measurements remotely.
- The formal caregivers: Nurses and physicians who review the information and follow up patients/users.
- Researchers: Eventually, a researcher would analyze the data gathered from a patient/user or a group of them to investigate into a specific scenario or pathology.
- The technicians: They are in charge of ensuring that the hospital devices and the back-end services work properly.

In the most basic architecture, the patient/user is the only actor involved, who monitors themselves using a mobile application (synchronized or not with a server acting as PHR), becoming an informal caregiver to themselves.

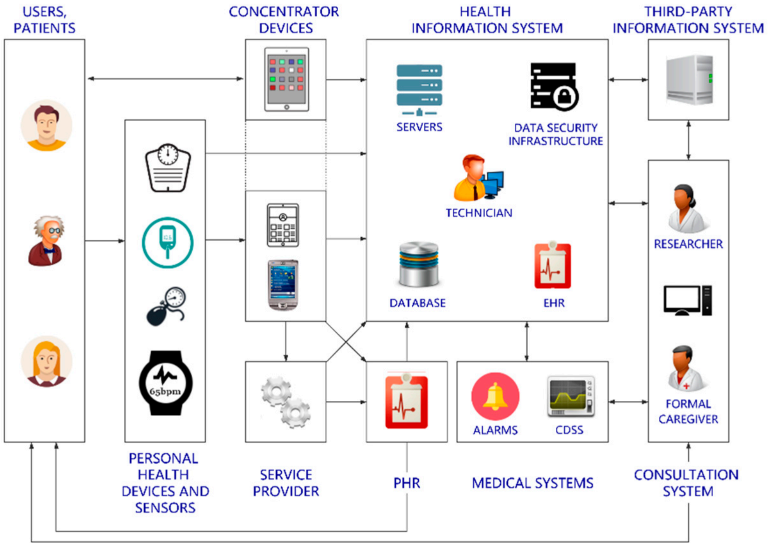


Figure 1. Actors and communication flows in a traditional mobile health architecture.

2.2. *Limitations of the Traditional Approach, Current Evolution and Potential Features*

As technology evolved and mHealth architectures were becoming more pervasive, additional problems were detected and higher architectural requirements were considered necessary.

One of the first issues identified was the lack of interoperability [26,27], which has been a common topic of debate to date. This issue has commonly been addressed by creating medical terminologies and medical standards. Within the former group, one of the most prominent is the Logical Observation

Identifiers Names and Codes (LOINC). The latter group comprises a wide variety of examples. A prominent effort would be the International Organization for Standardization (ISO)/Institute of Electrical and Electronics Engineers (IEEE) 11073, intended for the interoperability of medical devices. Another example is DICOM, intended for medical images. Additionally, the Standard Communications Protocol for computer assisted ElectroCardioGraphy (SCP-ECG) for the transmission of ECGs. To conclude with the examples, HL7 is a set of standards for facilitating the exchange of medical information. Architecturally, some organizations have proposed a variety of technical frameworks to promote the use of such standards. The most prominent is Integrating the Healthcare Enterprise (IHE). It provides a set of profiles that describe clinical information needs or workflow scenarios and rely on existing medical standards to accomplish them. There has been effort in the literature presenting some architectural proposals of end-to-end standard-based mHealth frameworks [28,29].

Despite the undeniable benefits of standard-based mHealth, the lack of privacy and security is still one of the major concerns [30,31]. As regards to consumers attitude, having control over mHealth privacy and security features as well as trust in providers were recently identified as key issues [32]. Moreover, there are national and international regulations—as the aforementioned HIPAA or GDPR—which compel mHealth architecture designers to take into account aspects such as confidentiality, integrity, availability, accountability, auditability, authenticity, non-repudiation and privacy. Effort can be found in the literature aimed at providing secure, standard-based mHealth architectures. For example, Rubio et al. proposed a flexible structure that provides features tailored to the needs of different mHealth applications, based on a multi-layered, IHE-based extension of ISO/IEEE 11,073 [33].

In parallel, while traditional mHealth architectures were deployed by healthcare authorities, they were thereby inevitably—albeit not intentionally—mainly focused on medical staff and supporting clinical work. In such traditional structures, the core of the service is located at the HIS, and it is developed by the public or private healthcare organization offering the mHealth service. This situation entails an effort to maintain the software updated and the data available. Current trends in data management, however, indicate a paradigm shift towards cloud computing, which enables designers to consume different resources on-demand. Such resources include infrastructure (computation, storage, networking), components that facilitate the creation of applications and services (e.g., middleware), and third-party software and/or data. Services such as Amazon Web Services, Google Cloud Platform or Microsoft Azure have gained noticeable ground, due to their wide range of options and flexibility. In this context, Rahimi et al. reviewed the state of the art of cloud computing in mobile environments and illustrated their application to various domains, including health [34]. In the same report, they alert that security and privacy are critical aspects with still open research issues. Nonetheless, effort towards mobile cloud computing in health environments taking into account—albeit to different degrees—standardization and security/privacy concerns can be found in the literature [35–37]. In 2012, Hsieh et al. proposed cloud and pervasive computing based 12-lead electrocardiography service to realize ubiquitous 12-lead ECG tele-diagnosis. In such paper, they selected the Microsoft Azure cloud to process and store heterogeneous ECG formats (e.g., SCP-ECG or DICOM-ECG). They included some security and privacy features. For example, authentication based on roles and internet protocol address range, data encryption (via hypertext transfer protocol secure (HTTPS)), secret key protected storage or ECG file encryption and verification while reports are retrieved [35]. In 2013, Ribeiro et al. described a solution for outsourcing medical images to Amazon elastic compute cloud based on DICOM and a number of IHE profiles, but foremost on cross-enterprise document sharing for images. As regards to security, they proposed an encryption method which hides access patterns to attackers, yet allows searches through the content [36]. In 2016, Hanen et al. published a healthcare system in mobile cloud computing environments. They used a cloud simulator to convey DICOM-compliant medical images considering some security and privacy issues, such as authentication, access control or data encryption [37]—although the real implementation has not been published so far.

While the cloud is a promising technology for mobile health care environments, it is mainly intended for back-end purposes—usually including computing load. In addition, cloud-computing technologies are not directly connected to the user's personal network. In contrast, healthcare frameworks that are user-friendly, social, empowering, decentralized, technically easy to deploy and self-manageable could lead to a paradigm shift. This can be achieved by using social media, e.g., Twitter, Facebook, etc., which can be seen as a particularization of clouds. At a technical level, this framework would be decentralized and highly flexible, designed to enable and promote contents with global reach and high frequency. This would be achieved thanks to inexpensive means (generally, no monetary cost is charged to the end user) and practical tools available for anybody to publish, share and view contents within short delay. Therefore, the development of social-media-based mHealth services has the potential to promote the recruitment and reinforce the engagement of users and their communities. It could also enable fast, flexible, user-oriented and user-controlled configuration of mHealth architectures and fast and inexpensive structural deployment. To do so, mHealth apps that use social media to manage and share personal biomedical data in an automatic way can be built by means of the public Application Programming Interfaces (APIs) exposed by social media companies. An associated issue that must be addressed when building social-media-based services is that such APIs do not allow a high ratio of data sent per post (e.g., Twitter limitation of characters). However, more importantly, interoperability, security and privacy concerns should not be overlooked.

Considering all the above, to date, the best approximation to an comprehensive system was conducted by Triantafyllidis et al. [19], who used a social media (Twitter) to monitor patient data. In order to univocally describe the symptoms or alerts tweeted by the patient, this proposal makes use of the Systematized Nomenclature of Medicine—Clinical Terms (SNOMED-CT) [38] and the Unified Medical Language System (UMLS) [39] metathesaurus API. Nevertheless, the biomedical message is not formatted according to any biomedical standard—leaving aside the thesaurus. Thus, it could not interoperate seamlessly with a HIS. With respect to its security policy, it is strictly based on Twitter-related features. In particular, they make use of lists of users to control the privacy of people subscribing to a service. They also use Open Authorization (OAuth) [40] to authorize the automated sending and receiving of tweets from a user account. Finally, their system relies on default HTTPS for secure communications. Nevertheless, this scheme does not implement end-to-end security, and thus the information can be accessed in clear in the Twitter servers. An enhancement of such framework was proposed by the authors in [20], where a proof-of-concept system for following up cardiovascular patients using Twitter and HL7 was implemented. However, it was still restricted to a specific medical standard and the security and privacy measures implemented were just those built-in by Twitter. Moreover, the system relied in a traditional client-server architecture, which reduced the possibilities of users creating their own systems and thus empower themselves.

Efforts in the literature have proposed some specific frameworks partially fulfilling the requirements for mH3S services. To date, however, there is no proposal integrating all these concerns in a single system. Therefore, in this paper we propose a generic architecture for easy-to-deploy mHealth services based on social media, which convey information in compliance with medical standards, while enhancing security and privacy of end users. Thus, the architecture will enjoy the advantages that a social media network provides, such as a user/patient social network, built-in reliability and scalability, or up-to-date GDPR-compliant servers, while conveying standardized biomedical data with enhanced security and privacy measures applied.

3. Results

As a result of the analysis performed in Section 2, a generic mH3S architecture is proposed within this section. The details of the proposed platform and a proof-of-concept thereof are thoroughly described in the following subsections.

3.1. Proposal of a Generic mH3S Architecture

The newly created architecture is illustrated in Figure 2. It depicts a system for storing, communicating, and consulting medical data as well as generating and distributing alarms, while bridging the traditional communication gap between users, formal and informal caregivers and researchers.

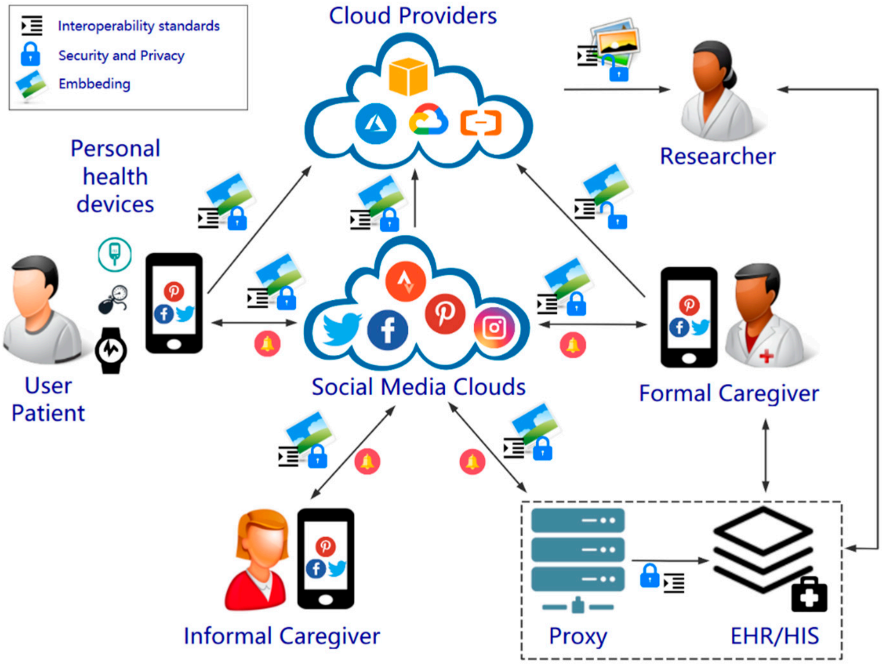


Figure 2. Proposal of the generic architecture.

3.1.1. Actors of the Proposed Architecture

The general architecture is comprised of eight actors (see Figure 2):

- User/Patient: They will gather the biomedical information and use a mobile phone or tablet to post them. They could also receive feedback from formal or informal caregivers.
- PHDs: The medical devices used to monitor the user's status.
- Informal caregiver: A friend, a neighbor or a relative that helps the user/patient to monitor and control their biomedical data within the healthy range of values. Informal caregivers are typically in charge of a reduced number of user/patients.
- Formal caregiver: A healthcare provider associated to a professional formal system, e.g., physicians, nurses or social workers. They are able to take care of a relatively large number of patients.
- EHR/HIS: This represents the traditional EHR/HIS. A proxy would be required to receive the biomedical data (by means of the social media API), decrypt, decode and send such information—with an appropriate format– to the actual EHR/HIS, which would store the information. This may need further policies in order to ensure the correct identification of the sender.
- Cloud providers: Such systems may be used for further computing (e.g., signal processing), redistribution or massive storage.
- Researcher: They would perform medical research over a potentially high volume of anonymized biomedical information thorough appropriate data mining. The open lock in Figure 2 means that the content has been decrypted—and anonymized—for research purposes.
- Social media: The core of the proposed system, which is used as a backbone for communication and, to a certain extent, storage purposes.

It is worth noting however, that in specific particularizations of the architecture, some of them may not be present. For example, in a simple system, a user/patient and an informal caregiver may use the social media to exchange biomedical information. In a formal scenario, a patient would communicate with a formal caregiver through the application.

The intended target audience of the system depends on the scenario. In a scenario with a formal caregiver, they should evaluate the situation, the patient, and their literacy (both general and technological). After the assessment, the formal caregiver would decide whether or not prescribe the use of the application. In an informal scenario, both the user and the informal caregiver should evaluate the advantages and drawbacks of the tool at their disposal. If it is a hospital developing the application, proper information about the implications could be published so that users could make an informed decision.

3.1.2. Configuration, Usage and Communication Flow

The set-up requires minimal configuration. Users will be asked to install a specific mobile application (which should be developed to offer the mH3S service). Such application shall rely internally on a social media API. Users will therefore need an account of this social media network. Since one of the main features of the framework is its security and privacy, first of all, the users—patients, formal and informal caregivers, a hospital—willing to exchange information need to generate a pair of public and private keys. The public key has to be exchanged beforehand. For security reasons, the public key must be sent through a communications channel different than the one used for biomedical data exchange. This could be done by various means, e.g., by scanning a Quick Response (QR) code. The details of the security and privacy scheme are detailed later in Section 3.1.4. Once the public keys have been exchanged, the communication can begin. However, depending on the social media chosen and on how their accounts are configured, the users may be required to befriend each other beforehand. This can also be done inside or outside the main application. Within the application however, the receiver has to introduce the name of the sender in the social media network. This will work as a subscription to the posts of the sender.

After the configuration process ends, the biomedical information would be gathered and subsequently standardized, encrypted and embedded in an object media, promoting the friendly use of social media network (see Section 3.1.4 as well). Immediately afterward, the user/patient would post the media object with embedded information to the social media (see Figure 2). All users subscribed to the sender posts—for example, an informal caregiver—will receive notification of the update and would be able to decrypt the biomedical information, given the key exchange carried out beforehand. Therefore, the social media would be used as a backbone.

Subsequently, the information received could be stored or analyzed. The receiver may answer to the initial sender just for acknowledgement or to recommend or prescribe something, and the communication could be naturally carried out in an analogous way. In any case, the information could be also transmitted to a HIS/PHR and it could be anonymized and sent to a cloud provider for research purposes.

3.1.3. Eligible Social Media

The architecture proposed relies in existing social media, such as Twitter, Facebook, etc. In order for a social media to be eligible for serving as backbone network for the proposed mHealth architecture, it should meet a number of compulsory requirements:

- The social media must offer a public API, so that the medical information can be exchanged programmatically. Most social media today offer this option.
- The social media must allow the sharing of information among users. Naturally, this is something that most social media enable, since it is part of the foundation of social media themselves.

- The social media must allow the sharing of multimedia contents, where the biomedical information would be embedded. Again, this is a common feature in current social media.
- The social media API rate limits and delays (if present) must be good enough to support mHealth services exchanging biomedical information in a sufficiently fluent fashion.
- The social media must fulfill the security and privacy regulatory requirements where applicable (GDPR, HIPAA, etc.)

Moreover, if the social media has other enhancing optional features, extra features can be implemented, for example:

- If the social media has a comprehensive, non-expiring search engine that allows searches against past posts using the API, the social media could be used as a reliable cloud storage system. Otherwise, the architecture shall rely on distributed local storage or re-posts for handling persistence.
  Given that the data uploaded will be encrypted, the search would rely on additional plaintext keywords. For example, the user could add a comment (e.g., "#vitalsigns" or "#myPHR") alongside the picture with biomedical data embedded, so that the social media search engine can find the post later. A wise selection of plaintext keywords is naturally required to preserve the actual content.
- If the social media has no API rate limits, the architecture can be scaled more easily. Some social media may offer paying plans for higher API rate limits. In any case, the limits established by most social media are high enough to support small-scale project, such as those involving informal caregiving.
- If the social media enables closed, easy-to-administrate groups, the architecture could be used for helping communities of users, such as users/patients with the same condition.
- If the social media enables private messages, the architecture can provide an extra layer of privacy.
- If the social media allows the verification of accounts, hospitals and formal caregivers could make use of them, increasing thereby users' trust.

3.1.4. Formatting of the Biomedical Objects

According to the principles of the proposed architecture, the biomedical information shall be in a standard format, appropriately protected and presented in a manner that fosters the exchange through social media. To create a suitable formatting, a number of issues have to be taken into account:

- Standards and Interoperability: First, the medical information shall be standardized with a medical standard, e.g., HL7, SCP-ECG, DICOM, etc. This enables that the information involved be transmitted and stored in an open, known format, which ensures a common structure, fostering thereby further exchange with hospitals or service providers. The selected standards may also use a medical terminology as a reference, such as SNOMED-CT.
- Security and privacy: To ensure the security and privacy of the users, the exchanged biomedical data, which were previously standardized into the medical standard chosen, e.g., HL7, shall be subsequently encrypted and signed. To do so, we propose that the architecture shall rely in any of the existing, standardized envelopes, such as the Cryptographic Message Syntax (CMS) [41] or openPGP (open Pretty Good Privacy) [42], which are able to encrypt and sign arbitrary data. These envelopes support a set of cryptographic algorithms. Thus, when implementing a specific envelope, a number of algorithms should be selected among the aforementioned set. In general, security envelopes are hybrid cryptographic systems, meaning that they combine symmetric and asymmetric cryptography. First, a session key is generated and used to encrypt the plain data. Then, the session key is encrypted with the public key of the receiver. The first process is performed fast (in bytes/second) while the latter is slower. However, given that the session key is small compared to the data to be encrypted, the whole process is carried out swiftly. All keys shall be created using high entropy key generators and have appropriate lengths in order to guarantee

resilience against cryptanalysis.
All cryptographic-related activities (cryptographic functions and cryptographic material management) are encouraged to be performed and stored inside a Trusted Platform Module (TPM).

- Presentation: The architecture proposed is based on social media, and as such, it is desirable that the information exchanged be presented in a user-friendly way. Therefore, we propose that the biomedical information—after being standardized into a medical standard and secured in an envelope—be embedded into a multimedia content. Such content could be a simple static image, an animated image, an audio file, a video file, etc., as long as the chosen social media supports the selected format(s).
- Embedding: Social media usually limit the amount of characters that can be sent in a single post. Thus, embedding the biomedical information—standardized and encrypted—into a multimedia content—such as image, audio or video—not only favors user friendliness, but it also provides a much higher data-per-post ratio. There exist a wide variety of embedding algorithms in the literature. As an example, an algorithm aimed at embedding biomedical information into images can be found here [43]. Any embedding algorithm could be potentially used within the architecture proposed. Nonetheless, in the architecture proposed, some features related to the embedding process have to be taken into account. First, the multimedia content has no clinical meaning per se. Second, the fact that there are data embedded has not necessarily to be a secret. Third, there is a need for some manipulation resilience—since social media may process images or videos for compression purposes. Finally, all embedded data must be flawlessly recovered. In general, when selecting an embedding algorithm, it is worth mentioning that there is usually a trade-off between speed, distortion and robustness, as discussed below in Section 4.

3.2. Technical Proof-of-Concept Implementation of the Architecture Proposed

This section could be seen as a particularization of Section 3.1. Thus, some choices were made for the proof of concept: Twitter as social media, version 2 of HL7 as medical interoperability standard, openPGP as security envelope and a specific algorithm for embedding the data into an image (see Figure 3).

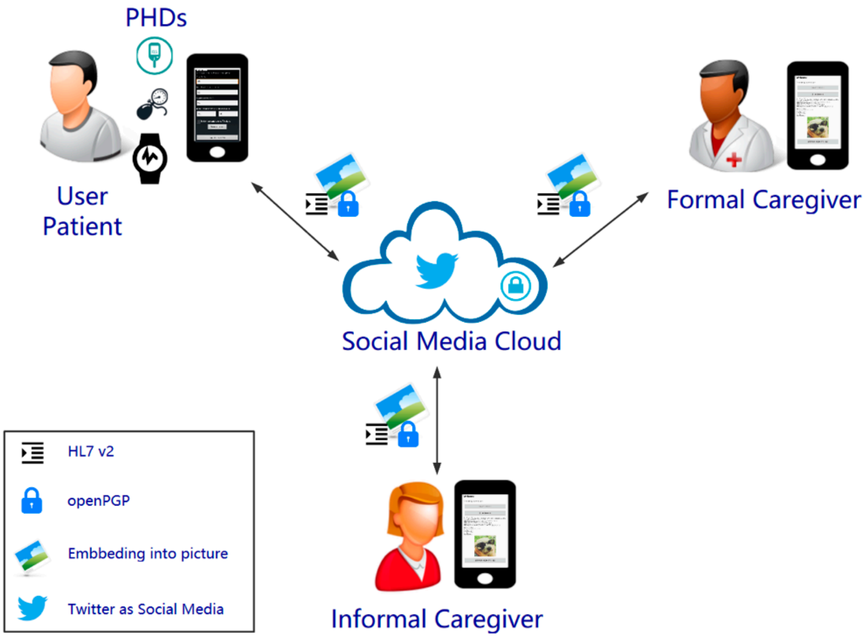


Figure 3. Architecture of the proof of concept using Twitter, Health Level 7 (HL7), open Pretty Good Privacy (openPGP) and a specific embedding algorithm.

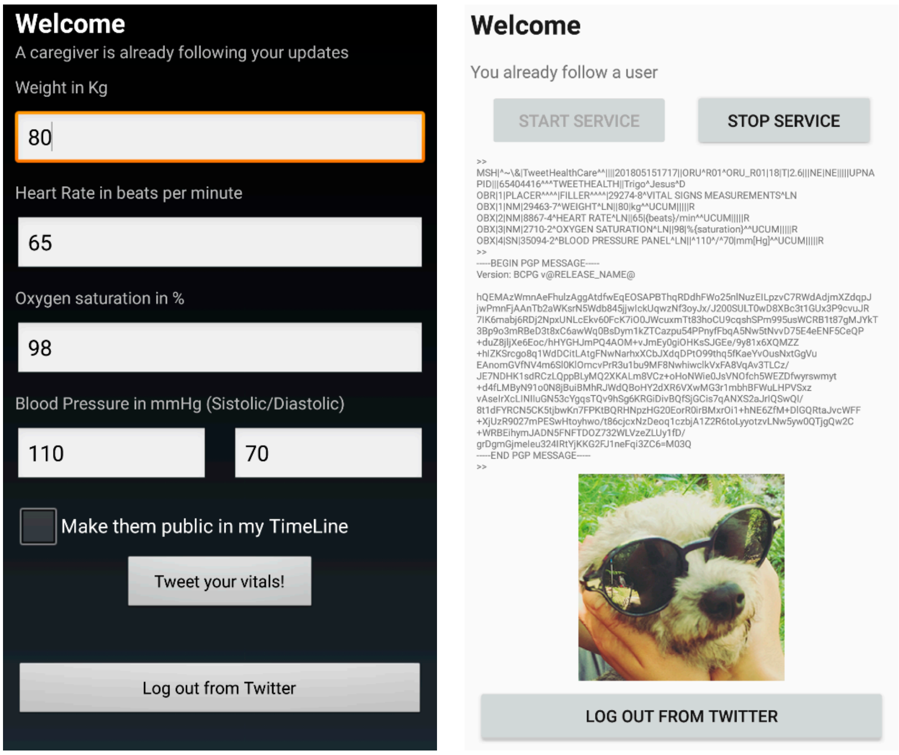
3.2.1. Actors of the Implemented Architecture

The architecture implemented for the proof-of-concept is comprised of up to five actors (see Figure 3), namely the user/patient, the PHDs, a formal and/or an informal caregiver and the social media, which is a subset of the actors enumerated in Section 3.1.1.

3.2.2. Configuration, Usage and Communication Flow

To accomplish such system, two Android applications were developed within the framework of this project (both written in Java programming language and developed using Eclipse). The first one is intended for users/patients and the second one for caregivers. Both shall install the respective application and log in with their Twitter accounts. They both have to set within the application who will be the person on the other end by filling out a textbox with the other person's Twitter account. Then, a QR code is used to exchange the public key. Note that this could be achieved by other means.

The application for users/patients, see Figure 4a, allows them to tweet a set of clinical findings (that is, a collection of medical-related observations). In this proof of concept, the clinical findings are weight, heart rate, oxygen saturation and blood pressure. After such findings are gathered, they will be posted as an HL7 v2 message encrypted and signed with openPGP and embedded into an image. For this proof of concept, users can include their weight, heart rate, oxygen saturation and blood pressure. The application, in addition to tweeting the standardized, encrypted, signed and embedded data into an image, also offers an additional option to tweet a textual summary of the clinical findings to the public time line of the user. This is intended for users who would like to disclose their data publicly.



Welcome

A caregiver is already following your updates

Weight in Kg

80

Heart Rate in beats per minute

65

Oxygen saturation in %

98

Blood Pressure in mmHg (Sistolic/Diastolic)

110 70

☐ Make them public in my TimeLine

Welcome

You already follow a user

```
>>
MSH|^~\&|TweetHealthCare^^||||201805151717||ORU^R01^ORU_R01|18|T|2.6|||NE|NE|||||UPNA
PID||||65404416^^^TWEETHEALTH||Trigo^Jesus^D
OBR|1|PLACER^^^^|FILLER^^^^|29274-8^VITAL SIGNS MEASUREMENTS^LN
OBX|1|NM|29463-7^WEIGHT^LN||80|kg^^UCUM|||||R
OBX|2|NM|8867-4^HEART RATE^LN||65|{beats}/min^^UCUM|||||R
OBX|3|NM|2710-2^OXYGEN SATURATION^LN||98|%{saturation}^^UCUM|||||R
OBX|4|SN|35094-2^BLOOD PRESSURE PANEL^LN||^110^/^70|mm[Hg]^^UCUM|||||R
>>
-----BEGIN PGP MESSAGE-----
Version: BCPG v@RELEASE_NAME@

hQEMAzWmnAeFhulzAggAtdfwEqEOSAPBThqRDdhFWo25nlNuzEILpzvC7RWdAdjmXZdqpJ
jwPmnFjAAnTb2aWKsrN5Wdb845jjwIckUqwzNf3oyJx/J200SULT0wD8XBc3t1GUx3P9cvuJR
7IK6mabj6RDj2NpxUNLcEkv60FcK7iO0JWcuxmTt83hoCU9cqshSPm995usWCRB1t87gMJYkT
3Bp9o3mRBeD3t8xC6awWq0BsDym1kZTCazpu54PPnyfFbqA5Nw5tNvvD75E4eENF5CeQP
+duZ8jljXe6Eoc/hHYGHJmPQ4AOM+vJmEy0giOHKsSJGEe/9y81x6XQMZZ
+hlZKSrcgo8q1WdDCitLAtgFNwNarhxXCbJXdqDPtO99thq5fKaeYvOusNxtGgVu
EAnomGVfNV4m6Sl0KlOmcvPrR3u1bu9MF8NwhiwclkVxFA8VqAv3TLCz/
JE7NDHK1sdRCzLQppBLyMQ2XKALm8VCz+oHoNWie0JsVNOfch5WEZDfwyrswmyt
+d4fLMByN91o0N8jBuiBMhRJWdQBoHY2dXR6VXwMG3r1mbhBFWuLHPVSxz
vAseIrXcLINIluGN53cYgqsTQv9hSg6KRGiDivBQfSjGCis7qANXS2aJrIQSwQI/
8t1dFYRCN5CK5tjbwKn7FPKtBQRHNpzHG20EorR0irBMxrOi1+hNE6ZfM+DlGQRtaJvcWFF
+XjUzR9027mPESwHtoyhwo/t86cjcxNzDeoq1czbjA1Z2R6toLyyotzvLNw5yw0QTjgQw2C
+WRBEihymJADN5FNFTDOZ732WLVzeZLUy1fD/
grDgmGjmeleu324IRtYjKKG2FJ1neFqi3ZC6=M03Q
-----END PGP MESSAGE-----
>>
```

(**a**) (**b**)

Figure 4. Android applications: (**a**) for users or patients (at this point, the user is already logged in and a caregiver is already following the updates); (**b**) for caregivers (the user is already logged in, the caregiver is following someone's updates, an image with biomedical data has already arrived, and such data have been de-embedded and decrypted). The first rows have data embedded.

The second Android application is intended for caregivers—either formal or informal. Therefore, it allows them to follow the updates of a user/patient. In order to implement such feature, an Android service is always running in background. Such service runs in a separated thread—so that the main thread is not blocked—and with background priority—so intensive workload does not disrupt the user interface. In addition, mechanisms to restart the service have been included, should it get killed. The service relies on Twitter's API to receive all the data from a user (the user logged in), so that when a new message is received, it can be adequately processed. In Figure 4b, a successfully received transmission of biomedical data embedded into a pre-existing image is shown. Details of the embedding algorithm are explained below at the end of this subsection. Note that the encrypted PGP message as well as the decrypted HL7 message have been printed for the sake of illustration, but in a real application, this could be transparent to the user.

The actions performed as well as the communication flow among actors are depicted in Figure 5. The upper section (section A) is only executed once (when the system is being configured). Thereafter, only section B of the flow is required.

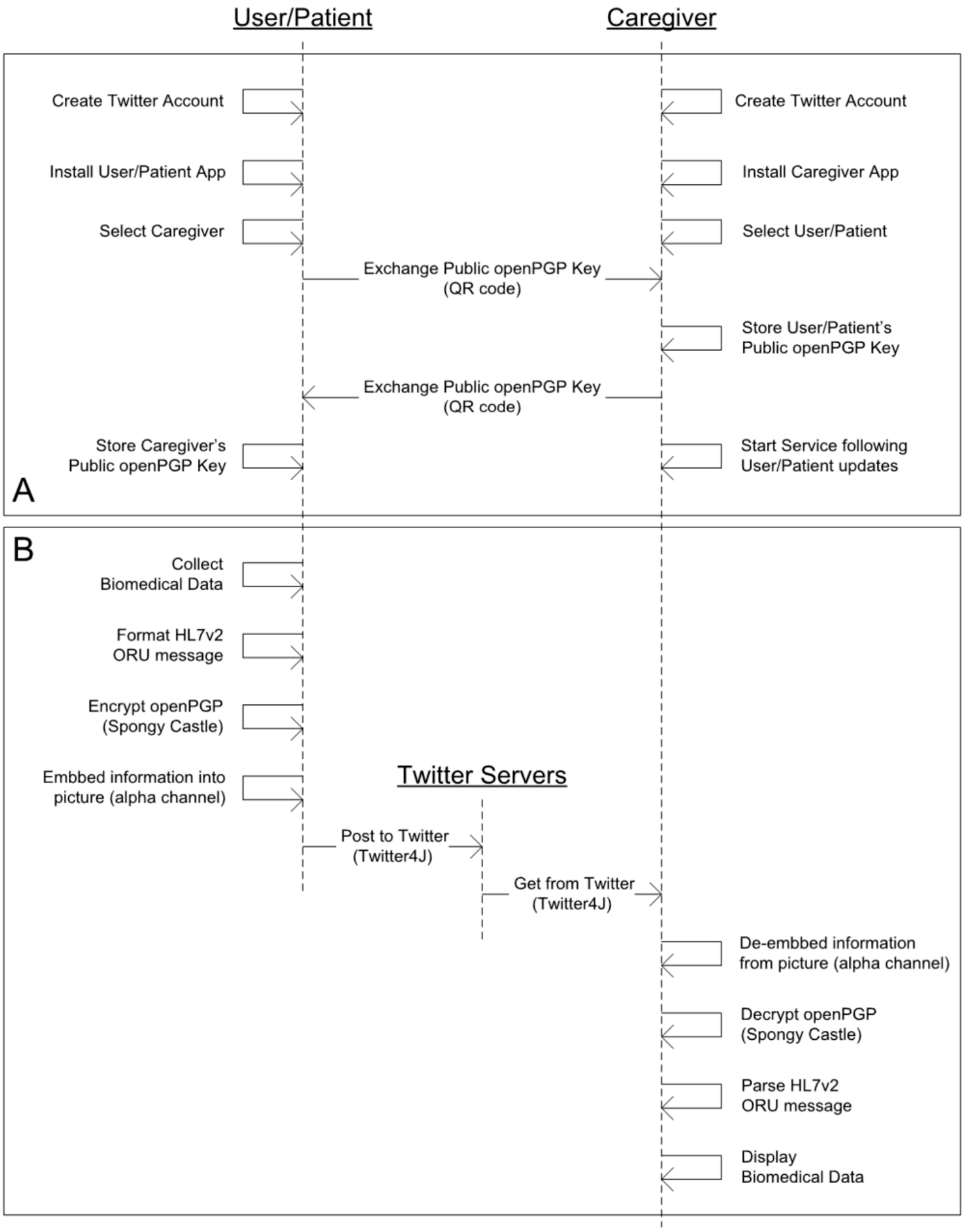

User/Patient
Caregiver
Create Twitter Account
Create Twitter Account
Install User/Patient App
Install Caregiver App
Select Caregiver
Select User/Patient
Exchange Public openPGP Key (QR code)
Store User/Patient's Public openPGP Key
Exchange Public openPGP Key (QR code)
Store Caregiver's Public openPGP Key
Start Service following User/Patient updates
A
B
Collect Biomedical Data
Format HL7v2 ORU message
Encrypt openPGP (Spongy Castle)
Embed information into picture (alpha channel)
Twitter Servers
Post to Twitter (Twitter4J)
Get from Twitter (Twitter4J)
De-embed information from picture (alpha channel)
Decrypt openPGP (Spongy Castle)
Parse HL7v2 ORU message
Display Biomedical Data

Figure 5. Actions performed and communication flow among the actors. Section (**A**) is the configuration phase. Section (**B**) is the operation phase and it is performed every time new measures are sent.

3.2.3. Social Media Selected

Most modern social media meet the compulsory requirements posed in Section 3.1.3. For this specific technical proof of concept, Twitter has been selected as the social media that will support the proposed architecture. Twitter complies with all the proposed requirements, since it offers a public API, which allows multimedia sharing. In addition, API rate limits are rather generous. More specifically, Twitter streaming API limits are 5000 follow users and roughly 1% of the tweets being tweeted globally. Considering that approximately half a billion tweets are tweeted every day, this provides a mean of 5787 tweets per second, enabling a wide margin telemonitoring applications. Finally, Twitter complies with the GDPR regulatory requirements [44,45].

As regards to the optional features, our proof-of-concept uses private messages, but not lists (a kind of groups). The search engine can be used to the extent allowed at no cost by Twitter. A verified account for a hospital or a formal caregiver could be easily created and used in our proof-of-concept. The rationale of these decisions is discussed below in Section 4.2 (subsection "Social Media").

In order to build the application programmatically, the Twitter4J library was used. A stream was created to follow all updates related to a particular user (the user logged in). We made use of Twitter direct messages to address a particular receiver, so that the application knows that new data may be embedded in the picture attached to the tweet.

3.2.4. Implemented Formatting of the Biomedical Objects

- Standards and Interoperability: HL7 version 2 has been selected to provide interoperability. Such standard aims at supporting hospital workflows by means of short, human-readable messages. More specifically, the message implemented is the "unsolicited transmission of an observation message" (referred to as "ORU" type). This message gathers the clinical findings of the user/patient. For enhanced interoperability, the findings are referred to with the appropriate LOINC [46] code, which is a common terminology for laboratory and clinical observations. In addition, the units of measure are expressed through Unified Code for Units of Measure (UCUM) [47], which is a code system intended to include all units of measures being contemporarily used in international science, engineering, and business.
- Security and Privacy: Regarding the social media selected, posting on behalf of an account in Twitter relies on version 1.0A of OAuth [40]. OAuth is a protocol aimed at allowing users to grant access to third-party applications to their account without sharing their password. As regards to the security envelope, we implemented openPGP, which allows the encryption and digital signature of messages. The encryption algorithms behind openPGP are Advanced Encryption Standard (AES) and Rivest, Shamir and Adleman (RSA). The length of the keys is 128 bit for the session keys and 2048 bit for the asymmetric keys. The digital signature algorithm is SHA-256. openPGP is based on the Web of Trust concept, a decentralized Public Key Infrastructure (PKI) where every user can potentially be a Certification Authority (CA), so a user's certificate can be signed by other users, which in turn creates an open way of distributing public keys. In practice, we exchange the public keys by scanning a QR code and store them in a PGP keyring. In order to accomplish the implementation, we used the Spongy Castle Android library, which is a collection of cryptography APIs repackaged from the Bouncy Castle Java library.

  In addition, the Twitter API post and get requests rely on HTTPS with in turn uses the Transport Layer Security (TLS) protocol, which enhances communications security and privacy.
- Presentation: Among all the possibilities of multimedia content available, we propose that the biomedical blob—i.e., the clinical findings formatted in HL7 and subsequently encrypted and signed with openPGP—be inserted into a pre-existing picture in compliance with a "Red Green Blue + Alpha" (RGBA/ARGB) color model (see below).
- Embedding: A simple embedding algorithm has been implemented, and it is explained as follows. As is well known, a raw image contains a number of pixels, each of them comprised by four bytes:

three for color information—red, green and blue—and one byte related to transparency—also referred to as alpha channel. In this implementation, two versions have been tested:

○ In the first one, the data blob is embedded into the picture by replacing the transparency bytes of the pre-existing image with the bytes of the data blob. This modification will add some "noise" to the image.

○ In the second one, the bytes to be embedded are preprocessed, adding 0x85 to each byte beforehand (see Figure 6). The rationale behind this approach is as follows. We are embedding openPGP data. openPGP uses Radix-64 encoding, which is similar to Base64 [48], but with an additional 3-byte cyclic redundancy check. Base64 was designed to convey data across channels that only reliably support (printable) text content. Thus, the most common Base64 implementation uses these 64 characters: A–Z, a–z, 0–9, plus symbols + and/. Those will be the most frequent characters. It also needs a character for padding (=), and openPGP will use line feed and carriage return to introduce BEGIN and END lines. Among those, the highest ASCII character is "z" (hexadecimal 0x7A). Adding 0x85 to 0x7A results in 0xFF, meaning that such pixel would be fully transparent after the embedding. This mechanism allows a less noisy image (the embedding uses bytes closer to transparent) at a cost of longer compression (the more transparent the bytes, the more complex the image in terms of color variation). See Section 4.2 for a numerical discussion.

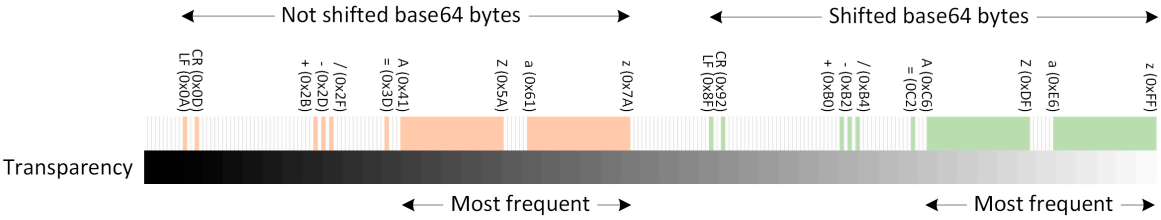


Figure 6. Longitudinally, this figure represents hexadecimal bytes from 0x00 (fully opaque in the alpha channel) to 0xFF (fully transparent). The left half shows where Radix-64 bytes are located (version 1 of the embedding algorithm). The right half shows where they are located in the shifted version of the algorithm.

4. Discussion

4.1. Discussion about the Overall Architecture

- Architecture: Architecturally speaking, this paper is innovative and different to every other approach published in the literature. We propose and end-to-end, encrypted and interoperable service running over online social networks. By creating and managing a pair of keys, the users become complete owners of their data. In addition, this is achieved without the users having to spend any money or deploying complicated frameworks. The social networks have usually powerful data centers with high uptime availability. Thus, the users are able to leverage such servers without actually sharing any personal/medical information travelling through them. Moreover, the process of information exchange is user-friendly.
- Social media: The importance and necessity of the introduction of social media for mHealth architectures is due to a number of reasons. They are user-friendly, social, empowering, self-manageable, decentralized, flexible, easy to deploy, global, (usually) free of charge, with high availability data centers, low latency and a vast mass of users.

  There are some other drawbacks or debatable characteristics as well. For example, the social media the architecture relies on may be a private company. This implies that the social media may unilaterally decide to modify the publicly exposed API. This could lead to temporary unavailability of the mHealth service until the applications are reprogrammed and redistributed. However, although the modification could be decided unilaterally, they are always announced in advance. Social media offer temporal windows to adapt the applications relaying on their API to the new situation. If the API is discontinued or the criteria for being an eligible social media are no longer met, the mHealth service as it is would be forced to definitive closure. Such reliance on corporations could be grappled with by using—or implementing—an open, decentralized social media, such as the Twitter-like microblogging systems GNU social [49] or Mastodon [50], or the Facebook-like, non-profit, user-owned, distributed, social media Diaspora [51]. However, users and caregivers would be forced to migrate to—or at least create a new account for—this service, while the number of active users in the main social media—Twitter, Facebook, etc.—is rather big, which clearly eases adoption. Users may join existing servers of open social media, which could be a simple option, or alternatively, servers could be installed and controlled by user themselves or by formal or informal caregivers. However, installing and maintaining a server is not a simple process for most users. With the proposal depicted in this paper, however, the only required set-up is the installation of the applications in mobile phones, which is easy and straightforward, even for users with reduced technological literacy.

Another point of debate is the proposal of using social media as a backbone. As opposed to traditional clouds, social media offer limited capabilities for storage and searching, and they do not provide computing power whatsoever. Nonetheless, social media are user oriented, and they certainly enable data communication, which may suffice for most telemonitoring use cases (see Section 4.2. for a specific analysis on speed and data capacity).

The decentralized nature of the architecture presented here is also debatable. The authors propose an architecture based on social media opposed to cloud services, while social media are themselves cloud based. However, social media are used in the system as a backbone. In traditional architecture proposals, the biomedical data were conveyed through a typical client/server architecture to a computing cloud or sent to a single HIS located at the hospital, where they would be persisted. By using the proposed architecture, the biomedical data are scattered throughout the mobile devices of the users/patients and formal/informal caregivers. Thus, the architecture is, by nature, decentralized. Biomedical data do go through and are stored in the servers of the social media selected, but encryption prevents data from being accessed by them.

The actual worth of introducing social media is something that requires further research and falls beyond the scope of the paper. The perceived worthiness should be validated in real scenarios. In any case, the success of the platform may depend on the final users and their technological literacy, as well as on the specific architecture selected, among other factors.

- Standards and Interoperability: In principle, any standard is eligible for the proposed architecture. It is noteworthy however, that some standards may be more suitable for a specific use case or may be more practical depending on whether such standard is used in some other place within the overall architecture. In any case, the inclusion of relevant biomedical standards in the architecture allows the straightforward integration of the gathered biomedical health information in healthcare systems, whenever required.
- Security and privacy: The claimed security and privacy is provided by the end-to-end envelope encryption. This provides confidentiality, via a combination of symmetric and public-key encryption. Additionally, the digital signatures provide authentication and integrity validation. The integrity validator could be crucial in an embedding application like this, since it can be used to verify that the content of the message has not be tampered with.

  The proposed architecture only adds one mandatory layer of encryption, that enforced by openPGP. The rest of the security measures are built-in features provided by the social network or the internet protocols. Generally, the users cannot opt-out of such measures, but they are applied in a transparent way.

  Once configured, the use of the application is straightforward. However, a trade-off between security/privacy and usability is always present when designing technological architectures. In general, including additional security measures reduces usability. In the proposed approach, the security and privacy procedure added implies the management of a pair of public/private keys (generation, exchange and storage). Users must create a pair of public/private keys. This has to be done only once and it is as easy as pushing a button. The exchange of the public key is only required when contacting to a new user. The procedure could also be simple, for example, transmitting the key through a messaging app or using a QR code. Finally, users would only have to manage a way to access the keys stored in the keyring (e.g., password, pin, drawn pattern or even biometric authentication, such as fingerprint or facial recognition). Thus, the additional complexity of the platform could be considered low. Although the added complexity is naturally a subjective matter, the authors consider that the improved security and privacy outweighs the decreased usability.

The security and privacy measures presented herein are structured under an end-to-end paradigm, from the user/patient device to the formal/informal caregiver device. This means that the security protocols used are stacked as different layers that partially overlap each other, providing security along the whole communication and storage process. For example, the requests to the social media API may be carried out using HTTPS/TLS, but this covers only the segment from the devices to the social media server. The additional security envelope provides further encryption so that the information is not in accessible even within the social media servers.

For increased security, the algorithms selected shall be—whenever possible—different from and complementary to the algorithms already defined by the physical network, the social media and the biomedical standard—if they implement any. This way, if one cryptographic algorithm is compromised, the other may still be valid. In any case, if any of the chosen algorithms for a specific implementation were compromised, it would have to be replaced with another existing algorithm considered secure and suitable for the application at such time.

Users/patients may configure multiple receivers of biomedical information within the application. The architecture proposed implies the need of creating one encryption per follower. This may cause scalability issues, although this is not a problem in small-scale projects with few formal or informal caregivers. This end-to-end encryption is required in order to prevent potential eavesdropping—including the social media service provider itself—and is widely implemented in digital services today—such as Telegram, Signal or WhatsApp.

Security and privacy could still be compromised due to, for example, the user/patient exposing his/her private key. In that case, the biomedical information should be removed from the social media network. In general, online social networks are exposed to a variety of privacy and security threats, such as phishing, fake profile creation, or identity clone attacks [52]. The proposal presented herein is vulnerable to these attacks inasmuch as it relies on both the internet and on online social networks. Internet users in general and online social network users in particular must be aware of them and follow general recommendations. For example, they should customize adequate privacy settings of the social network or build trust with those applications and persons receiving or managing the users' personal information [52]. In addition, online social networks today offer recommendation to minimize these potential threats. For example, Twitter has published a webpage to offer guidance and help with general safety and security concerns [53], such as potentially compromised/hacked accounts, report impersonation accounts, and fake accounts, among others.

In any case, for the architecture to function, the concept and scope of trust needs to be addressed. In a scenario with a formal caregiver, both the caregiver and the final user must trust the application. However, this could be achieved if the application is developed and distributed via official channels by the hospital itself. In addition, users must naturally trust the doctor suggesting the use of the application. Effective health care is based on substantial trust between patients and professionals, including the clinical tools they choose to use [54]. In any case, patients should be properly informed and should sign an informed consent beforehand. For the informal caregiver scenario, the user and the caregiver must trust each other. This could be easily achieved, since they know each other personally and would even be relatives. They must also trust the application. Such application could be the same one as in the formal caregiver scenario, that is, developed and published by a hospital they trust. Finally, regardless of the scenario, the application code could be open source in order to enhance trust through transparency.

Having this concept of trust among users developed, there is no reason to mistrust the other end. Therefore, the exchange of the public keys is always performed in a way that can always be considered trustworthy. The exchange of public keys could be easily performed in person. For the formal caregiver scenario, the exchange is performed at the moment of enrolling in the program, for example, by showing a printed QR code for the other user to scan. Nevertheless, if the public keys are sent through the internet, there are mechanisms to confirm that the keys have not been tampered with. For example, users could generate a token of the public keys (e.g., a hash) and exchange them through an alternative channel (e.g., they would be short enough to be read over the phone).

All said, despite of the measures proposed, no security scheme is 100% secure or private. There is always room for security or privacy breaches. However, the proposed architecture provides users with a framework that is secure and private enough to create a feasible mobile health scenario. In general, a project like this does require constant maintenance and supervision. Specifically, those parts related to cryptography (libraries, algorithms, keys, etc.) might be compromised and need special attention.

- Presentation: We proposed embedding the biomedical data blob into a multimedia content. The rationale behind this decision is twofold. First, because of the social benefit: it is clearly friendlier to share a beautiful image than an array of apparently nonsense bytes. Additionally, there is also an economy factor, measured in number of posts per biomedical data. Social media usually restrict the maximum amount of characters (e.g., 140—recently 280—Unicode codepoints in a tweet) or the maximum size of an image or video (a few megabytes usually) shared in a single post. Thus, the multimedia approach was selected since it is possible to embed far more data in a single social media post if a multimedia content is used, compared to plain text. In exchange, the approach selected increases the overhead and the processing load, which implies higher delays. Nevertheless, this is not an issue, since the architecture is not intended to work in real time, and the encryption algorithms are fast enough, given the amount of biomedical data that fit in a single multimedia content.
- Embedding: Designers have to take into account that social media may tamper with the uploaded multimedia content in a way that may not be public, and, if the embedding algorithm selected is not robust enough, the information could be corrupted partially or completely. In the proposed architecture, a robust enough embedding algorithm could prevent in practice any potential data corruption. However, if the data are corrupted, the recipient will detect that the information has not been received properly. The envelope provides a mechanism to check this. Thus, the receiver could ask for a retransmission. Then, the embedding could be carried out using a more robust algorithm to prevent the same problem from occurring again. The sender's application could be provided with a pool of embedding algorithms, ordered in terms of robustness, so that the simplest/fastest one would selected, as long as the data could be retrieved at the other end. In any case, there is a trade-off between speed and robustness.

4.2. Discussion about the Proof of Concept

- Social Media: Private messages have been used to address a particular receiver. However, users may exchange other images aside from the healthcare service. If that is the case, checking the alpha channel and looking for the openPGP data structure is an effective mechanism to detect if there is actual biomedical information embedded into the picture.

  Twitter offers a kind of groups (referred to as "lists"), which can be public or private. However, users are not allowed to post directly to a list, so that every member of the list receive the tweet. Alternatively, in the system proposed, users can send a private message to all receivers (see the paragraph below).

We propose sharing the pictures through private messages, although this could be carried out via the public timeline. While the former would enable the possibility of multiple receivers in one single post/image, the latter enhances privacy.

As regards to the Twitter API maximum rates, it can be noted that they are generous enough for store-and-forward, small-scale projects, but they may not be enough in scenarios requiring higher data rate transmission or when dealing with a vast amount of users/patients and/or caregivers.

Twitter offers a free search API, but it is restricted to the past seven days and it is focused on relevance—and not in completeness. This limits the use of Twitter as a reliable persistence handler. However, they offer some enterprise, high-level search APIs that could be implemented with a monetary cost.

Finally, since users only publish images that are aligned with their taste and the topic of their usual social posts, the system is minimally invasive with regard to information noise. In any case, users may also create a different account for this purpose.

- Standards and Interoperability: Version 2 of HL7 is sufficient for the use case portrayed in the proof of concept—a user connecting with an informal caregiver. The amount of data transmitted in an HL7 version 2 message does not give rise to any issues –regarding the embedding, for example. In the proof of concept, there is no integration with personal health records or higher HIS, so one can envision that the use of a standard may not be necessary. However, the implementation of HL7 leaves open the possibility of further integration.
- Security and Privacy: We chose to implement openPGP over CMS. Technical differences among them are not decisive. Indeed, both can be used as a base for Secure/Multipurpose Internet Mail Extensions (S/MIME), which is an Internet Engineering Task Force (IETF) standard for public key encryption and digital signing of MIME data. CMS is widely used by companies by building a PKI through X.509 certificates, which had to be approved and signed by a CA beforehand, which usually requires some kind of payment. openPGP, on the other hand, provides an open, decentralized, free system for certificate management, which are the main reasons why we chose to implement this envelope.

  The security library chosen, Spongy Castle, based on Bouncy Caste, currently supports all versions of Android. However, Android has announced the deprecation of Bouncy Castle. Therefore, the implemented applications will require changing to (most likely) the default Android implementation in the forthcoming future.

  Regarding OAuth, we have implemented version 1.0a. Twitter supports both versions 1.0a and 2.0. However, only version 1.0a can be used for posting on behalf of another account.
- Presentation: The image shared in the proof of concept is pre-stored in the application. However, in order to enhance the user satisfaction and personal binding, the application could be improved by letting users upload their own images.

  We have selected an image with a RGBA/ARGB color model in order to modify the alpha channel while embedding the information. One could envision that tagged formats could be selected to embed the information in tags, thereby eliminating distortion. However, this could not be implemented in this specific proof of concept, since, at the moment of writing, Twitter strips and discards the metadata from uploaded images.
- Embedding: The algorithm implemented works flawlessly under the conditions documented. As the transparency information is being modified, the process naturally distorts the image. Less distorting and more robust algorithms could be implemented. However, as discussed above, there is a trade-off between speed and robustness/distortion. In our implementation, version 2 of the embedding algorithm results in less noisy images, but it takes longer to compress them (see Table 2).

The fact that the image becomes distorted does not have any effect on the "clinical" functioning of the platform, since the picture itself has no clinical meaning. However, a distorted image may diminish the "social" component of the platform. Under the conditions tested, with only a few bytes of information being transmitted, little distortion can be perceived by the human eye (see and zoom the dog picture in Figure 4b).

Additionally, since the social media processing algorithm may not be public and, moreover, the owners of the social media can change the internal algorithms over time without notification, there is a need to review and update the proposed embedding algorithms periodically. It is worth noting as well that, since social media select their processing algorithms, an embedding algorithm that works for a social media may not work for another.

- Data capacity: In order to discuss the performance of the proposed technical proof of concept, we present below an analysis in terms of data capacity and speed (at the transmitter end). Two scenarios were considered: the one proposed for proof-of-concept (a small-scale telemonitoring system transmitting just a few bytes) and a boundary scenario for the proof of concept (when all the space available is used up). In order to assess the speed, two sizes of images were tested for the small-scale scenario. For the boundary scenario, both versions of the embedding algorithm were tested.

  For calculating the space available in the latter scenario, some calculations have to be performed, according to Twitter's current image support policies [55]. For this proof of concept, we have decided to use the alpha channel for embedding purposes. Among the image formats supported by Twitter, WebP and Portable Networks Graphics (PNG) provide such channel. However, Twitter states that it will transcode all WebP formats to 85% quality Joint Photographic Experts Group (JPEG) with 4:2:0 chroma subsampling. This may distort the embedded information. A PNG-32 (8 bits per channel ARGB), on the other hand, will be left as-is if the image has more than 256 colors and it is 900 pixels or smaller in the longest dimension (that is, it can fit into 900 × 900). If it has fewer colors, it will be transcoded to PNG-8. If it is greater than 900 × 900, it will be tested to consider if they will remain PNG or if they will be converted to JPEG, being the latter more likely. Thus, if these requirements are met, an image with 900 × 900 pixels and alpha channel could be uploaded and it will not be transcoded by Twitter. Thus, the receiver application will be able to retrieve the embedded information flawlessly. Given that a pixel is composed of 4 bytes (ARGB/RGBA), a maximum of 900 × 900 = 810,000 alpha-channel bytes are available. However, not all of those bytes would be actual biomedical information. First, the begin PGP and end PGP lines, the version line, the digital signature and the aforementioned redundancy check shall be deducted (95 bytes). The remaining data are encoded in Radix-64. In such encoding, 4*(n/3) characters are needed to represent n bytes, and this needs to be rounded up to a multiple of 4. In this case, 1 byte is needed for padding. After decoding, the resulting data include the session key encrypted with the RSA algorithm (in our proof-of-concept, once encrypted, its length is 2048 bits, that is, 256 bytes) and the biomedical data encrypted with the AES session key. AES works in blocks, in our case of 128 bits (16 bytes). Therefore, some padding may be needed. In this situation, 4 padding bytes are required. At the end, a maximum of 607,168 bytes of biomedical data would be available.

  The amount of biomedical data could fit in approximately 600 kB depends on the data themselves and the medical standard selected, if any. For example, the size of the HL7-compliant message generated in the proof of concept (including weight, heart rate, oxygen saturation and blood pressure) is just 439 bytes in HL7-formatted plaintext, 1035 bytes after encrypting and encoding. In a 900 × 900 image, it will distort only the first row and partly the second row (out of 900 rows). A non-compressed SCP-ECG file (12 ECG leads, 1000 samples/second, 2 bytes/sample, 10 s) is just 240 kB of ECG data, plus a few bytes of SCP-ECG metadata. A number of compression techniques could be applied to enable larger registration times. For the case of digital images, a DICOM file would also have some metadata, but a picture of 600 kB would probably have enough quality for a telemonitoring scenario (e.g., teledermatology). It is worth noting that DICOM itself also supports image compression using JPEG2000 [56], and it could be used internally. Although there is lack of consensus about the tolerable degree of compression, JPEG2000 have been reported to allow compression ratios ranging 30:1–50:1 without affecting the clinical quality. Even that could be enhanced by selecting an optimized, ad-hoc parametrization for the specific medical use case [57]. Thus, with a 40:1 compression ratio, a 600 kB compressed image would mean a 24 MB raw image (8 megapixels RGB), which should be enough for a teledermatology consultation.

Furthermore, this is considering just one tweet. Naturally, more than one tweet could be used, enabling the transmission of large amounts of data (even the whole EHR) in a few tweets. The size of an EHR depends on a variety of factors. As a reference, in a 2011 report of the Beth Israel Deaconess Medical Center, a teaching hospital of Harvard Medical School, it was published that they generate 1 terabyte of clinical text data (structured and unstructured) per year and 19 terabyte of image data per year [58]. With 250,000 active patients at that time, that means 80 megabytes per patient per year. That translates to approximately 138 tweets per patient per year, that is, one tweet per patient every 2–3 days.

- Speed: The speed of the whole procedure depends on a variety of factors. For example, the processor, the programming language, the programming coding efficiency, the amount of data to be embedded, the size of the image, the embedding algorithms and the size of the keys, among others.

  Regarding the cryptography algorithms, the test data published in [59] have been used as a reference. Such tests were run with an Intel Core 2 at 1.83 GHz (similar to current mobile devices). The algorithms selected are those comparable to our proof of concept: AES (128-bit key) and RSA (2048-bit key). The results are shown in Table 2 for the two aforementioned scenarios. Such table also shows the time required for encoding in Radix-64, considering the same 1.83 GHz processor. The speed used for the calculations is 2 cycles/byte, which is a typical speed for base64 encoding/decoding, although it could be up to ten times faster using vector instructions [60]. The data show that encrypting and encoding is a matter of milliseconds (subtotal 1, row number 05 in Table 2), even for ~600 kB of data. The most time-consuming task would be the RSA encryption (row 02 in Table 2) for both scenarios.

  As regards the time required for embedding, there are not publicly available data, because this is not a standard algorithm. The authors have performed a pool of tests under the following conditions. The process measured includes opening a bitmap image, setting up the necessary variables, modify the required alpha-channel bytes and generate a compressed file in PNG format in ARGB/RBGA.

  First, the system was tested for the small-scale telemonitoring scenario. More specifically, 1035 bytes (corresponding to the size of the encrypted HL7-compliant message) were embedded into the alpha channel of the dog picture in Figure 4b, testing two sizes: 900 × 900 pixels and a rescaled version of 225 × 225 pixels (exactly 16 times less pixels). Secondly, we performed two tests (one per each version of the embedding algorithm) using all space available (900 × 900 alpha bytes). The rationale of testing two different versions of the embedding algorithm is that the PNG compression uses DEFLATE [61], a lossless compressed data format involving a combination of Lempel–Ziv-77 compression algorithm and Huffman coding. This implies that faster compression could be achieved for images with small variations of color. In other words, the higher the transparency, the faster the compression. Conversely, the image would be noisier and therefore less friendly.

  The tests were run in a current (as of 2019), mid-range mobile phone with an octa-core MediaTek Helio G90T, including 2 ARM Cortex-A76 at 2.05 GHz and 6 ARM Cortex-A55 at 2.0 GHz, 6 GB of RAM and running Android 9.0 (PPR1.180610.011). We performed 10 executions per test and calculated the average time and the standard deviation. This provides the reader with a notion of the central value of the results (arithmetic mean) and the amount of dispersion of the set of results (standard deviation). Throughout the paper, the results are expressed as "mean ± standard_deviation".

The results are shown in Table 2. In general, the standard deviation is low compared to the average. This suggests that the data are clustered around the mean. The embedding is fast (row 12 in Table 2) for both small-scale and boundary scenarios, taking up less than 30 ms in any case. It is particularly fast for the smaller image, where it needs only 3.7 ± 0.5 ms. When comparing the two embedding algorithms for the boundary scenario, we can observe that the shifting process itself (that is, adding 0x85 to each byte) only adds 0.3 ms (5.6 ± 0.5 ms vs. 5.3 ± 0.5 ms, row 11 in Table 2). For the small-scale scenario, the two embedding algorithms were not tested, as they would behave similarly, since only a few bytes are being modified, and thus the image remains largely unaffected (see zoomed images in row 07 in Table 2).

As regards the PNG compression, it can be seen that it depends largely on the size of the image (38.1 ± 0.6 ms at 225x255 vs. the rest, several times slower, at 900 × 900, row 13 in Table 2). The other factor affecting the PNG compression time is the version of the embedding algorithm. The compression of the shifted version is slower compared to the non-shifted one (703.5 ± 4.2 ms vs. 548.5 ± 3.8 ms, row 13 in Table 2). On the other hand, the image is clearer, which could improve users' engagement (row 07 in Table 2).

The data show that the bottleneck of the whole procedure is the PNG compression (row 13 in Table 2). Approximately 80-95% of the total time (row 14 in Table 2) is spent on the PNG compression. In these tests, we use the built-in compressor provided by Android, which is single-threaded. It is arguably feasible to parallelize the compression process, achieving a reduction of time that would scale linearly with the number of cores of the machine. Besides the cores of the central processing unit, mobile phones nowadays also incorporate multi-core graphic processing units, which could also be used for parallelized compression. To the best of our knowledge, there is no Android library available able to perform a parallel compression of PNG files (at the moment of writing). There exist, however, Java-based parallel compressors for GZIP [62], a file format similar to PNG, since it is also based on DEFLATE. Nevertheless, programming such a compressor falls out of the scope of this paper.

As a conclusion, for both the small-scale telemonitoring scenario and the boundary scenario, the whole process could be performed rather fast (less than 1 s in any case). It can be carried out considerably faster with smaller images: 48.2 ± 0.9 ms (225 × 225) vs. 719.6 ± 9.6 ms (900 × 900). In addition, using the non-shifted algorithm, the process can be performed faster in the boundary scenario to the detriment of noisier images. In any case, the results show that the speed would not be a critical issue for most mHealth scenarios.

Table 2. Speed and size analysis.

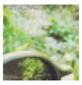
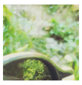
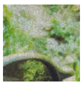
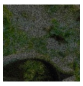
Row	Scenario	Small-Scale		Boundary	
01	Plaintext (biomedical) data (bytes)	439		607,168	
02	RSA encryption (ms)	6.370			
03	AES set-up and encryption (ms)	0.004		5.309	
04	Radix-64 encoding (ms)	0.001		0.664	
05	Subtotal 1: encryption and encoding (ms)	6.375		12.343	
06	Embedding Algorithm Version	Shifted	Shifted	Shifted	Not shifted
07	Zoomed (upper-right zone) image transparency overview				
08	Image dimension (width × height in pixels)	225 × 225		900 × 900	
09	Bytes embedded (bytes)	1035		900 × 900 = 810,000	
10	Embedding set up (ms)	3.6 ± 0.5	20.9 ± 1.2	24.3 ± 3.2	22.3 ± 1.8
11	Embedding (ms)	0.1 ± 0.3	3.9 ± 0.9	5.6 ± 0.5	5.3 ± 0.5
12	Embedding subtotal (ms)	3.7 ± 0.5	24.8 ± 1.9	29.9 ± 3.3	27.6 ± 1.1
13	PNG compression (ms)	38.1 ± 0.6	688.4 ± 9.1	703.5 ± 4.2	548.5 ± 3.8

Table 2. *Cont.*

Row	Scenario	Small-Scale		Boundary	
14	% PNG compression (100 * row 13/row 16)	79.1	95.7	94.3	93.2
15	Subtotal 2: embedding and compression (ms)	41.8 ± 0.9	713.2 ± 9.6	733.4 ± 5.2	576.1 ± 4.1
16	TOTAL TIME: subtotal 1 + subtotal 2 (ms)	48.2 ± 0.9	719.6 ± 9.6	745.7 ± 5.2	588.4 ± 4.1

As a final reflection on Section 4.2, it is worth noting that the elements selected for a specific proof of concept can be changed independently. If the designer decide to use Facebook instead of Twitter, the other particular selections (HL7, openPGP, or the embedding algorithms) would still be valid.

5. Conclusions

In this paper, a generic mH3S architecture has been proposed. Such architecture provides users with an enhanced healthcare service. The novelty of this approach can be summarized as follows:

- Novel architecture: Instead of a client-server architecture or a cloud-based architecture, we propose an end-to-end system that leverages online social networks as a backbone.
- Empowered users and patients: Users do not rely on anyone to create a secure, private communication channel. They decide what to share and with whom.
- User-friendly way: Attractive and relatable media objects (images, videos) with biomedical data embedded are shared through a social media network.
- Straightforward deployment: Users only need to install a mobile application and perform some minor configuration.
- Affordable: The users are not asked to spend any money to use the system. Only a mobile phone with internet connection is required.
- High uptime availability: The system leverages online social networks as a backbone. Thus, the servers are almost always up.
- Improved security and privacy: This is due to the security envelope, based on a hybrid cryptosystem, therefore combining the convenience of public-key approaches with the efficiency of symmetric-key schemes. Social networks convey the information, but not even they are able to read the biomedical data travelling through their servers.
- Reduced added complexity: Users are required to manage some key-related aspects, but this is common practice for current users of smartphones and applications and it could be carried out easily and swiftly.
- Augmented integrability: This is provided by the internal support of medical interoperability standards.

Additionally, a technical proof-of-concept implementation of such architecture has been developed by selecting a specific social media (Twitter), a security envelope (openPGP), an interoperability standard (HL7) as well as a specific embedding algorithm. To accomplish such a system, two Android applications were developed: one for users/patients and the other for formal/informal caregivers. This implementation demonstrates the feasibility of the platform. The tests show that the process is fast: less than 1 s, even for preparing (that is, encryption, encoding and embedding) ~600 kB of biomedical data. Thus, the additional complexity of the procedure does not entail impractical delays, and therefore, the platform can be considered fast enough for most mHealth telemonitoring services.

As a final reflection, it can be highlighted that, although the architecture presented and discussed in this manuscript has been motivated by a biomedical context, it could certainly be applied to other contexts. For example, by replacing the medical interoperability standard with another standard or data format suitable for the application. Therefore, the generic architecture proposed here can actually be seen as an enabler of payload transparency.

Author Contributions: Conceptualization, J.D.T., Ó.J.R., and Á.A.; methodology, J.D.T. and Ó.J.R.; software, J.D.T. and M.M.-E.; validation, J.D.T.; formal analysis, J.D.T. and Ó.J.R.; investigation, J.D.T.; resources, J.D.T., J.G. and L.S.-A.; data curation, J.D.T., Ó.J.R. and M.M.-E.; writing—original draft preparation, J.D.T. and Ó.J.R.; writing—review and editing, J.D.T., Ó.J.R., M.M.-E., Á.A., J.G. and L.S.-A.; visualization, J.D.T. and Ó.J.R.; supervision, J.G. and L.S.-A.; project administration, J.D.T., J.G. and L.S.-A.; funding acquisition, J.D.T., J.G. and L.S.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Public University of Navarra (project reference number PJUPNA29); Ministerio de Economía, Industria y Competitividad from Gobierno de España and European Regional Development Fund (reference number TIN2016-76770-R); Gobierno de Aragón (Reference Group T31_20R); and FEDER 2014-2020 "Construyendo Europa desde Aragón".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pew Research Center Mobile Fact Sheet. Available online: http://www.pewinternet.org/fact-sheet/mobile/ (accessed on 7 September 2020).
2. World Health Organization (WHO) mHealth. New Horizons for Health through Mobile Technologies. Available online: http://www.who.int/goe/publications/goe_mhealth_web.pdf (accessed on 7 September 2020).
3. Adibi, S. *Mobile Health: A Technology Road Map*; Springer Publishing Company, Inc.: New York, NY, USA, 2015; ISBN 978-3-319-12816-0.
4. Silva, B.M.C.; Rodrigues, J.J.P.C.; de la Torre Díez, I.; López-Coronado, M.; Saleem, K. Mobile-health: A review of current state in 2015. *J. Biomed. Inform.* **2015**, *56*, 265–272. [CrossRef] [PubMed]
5. Hamdi, O.; Chalouf, M.A.; Ouattara, D.; Krief, F. eHealth: Survey on research projects, comparative study of telemonitoring architectures and main issues. *J. Netw. Comput. Appl.* **2014**, *46*, 100–112. [CrossRef]
6. Awori, J.; Lee, J.M. A Maker Movement for Health: A New Paradigm for Health Innovation. *JAMA Pediatr.* **2017**, *171*, 107–108. [CrossRef] [PubMed]
7. Lee, J.M.; Hirschfeld, E.; Wedding, J. A Patient-Designed Do-It-Yourself Mobile Technology System for Diabetes: Promise and Challenges for a New Era in Medicine. *JAMA* **2016**, *315*, 1447–1448. [CrossRef]
8. Chen, C.; Haddad, D.; Selsky, J.; Hoffman, J.E.; Kravitz, R.L.; Estrin, D.E.; Sim, I. Making Sense of Mobile Health Data: An Open Architecture to Improve Individual- and Population-Level Health. *J. Med. Internet. Res.* **2012**, *14*. [CrossRef]
9. Househ, M.; Borycki, E.; Kushniruk, A. Empowering patients through social media: The benefits and challenges. *Health Inform. J.* **2014**, *20*, 50–58. [CrossRef]
10. Facebook Statistics. Available online: https://about.fb.com/company-info/ (accessed on 7 September 2020).
11. Kaplan, A.M.; Haenlein, M. Users of the world, unite! The challenges and opportunities of Social Media. *Bus. Horiz.* **2010**, *53*, 59–68. [CrossRef]
12. Househ, M. The use of social media in healthcare: Organizational, clinical, and patient perspectives. *Stud. Health Technol. Inf.* **2013**, *183*, 244–248.
13. Rozenblum, R.; Bates, D.W. Patient-centred healthcare, social media and the internet: The perfect storm? *BMJ Qual. Saf.* **2013**, *22*, 183–186. [CrossRef]
14. Grajales, F.J., III; Sheps, S.; Ho, K.; Novak-Lauscher, H.; Eysenbach, G. Social Media: A Review and Tutorial of Applications in Medicine and Health Care. *J. Med. Internet Res.* **2014**, *16*, e13. [CrossRef]
15. Hors-Fraile, S.; Atique, S.; Mayer, M.A.; Denecke, K.; Merolli, M.; Househ, M. The Unintended Consequences of Social Media in Healthcare: New Problems and New Solutions. *Yearb. Med. Inf.* **2016**, 47–52. [CrossRef]
16. Grover, S.; Aujla, G.S. Twitter data based prediction model for influenza epidemic. In Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 11–13 March 2015; pp. 873–879.
17. Al-Bahrani, R.; Danilovich, M.K.; Agrawal, A.; Choudhary, A. Towards Informal Caregiver Identification in Social Media. In Proceedings of the 2016 IEEE International Conference on Healthcare Informatics (ICHI), Chicago, IL, USA, 4–7 October 2016; p. 414.
18. Nair, L.R.; Shetty, S.D.; Shetty, S.D. Applying spark based machine learning model on streaming big data for health status prediction. *Comput. Electr. Eng.* **2017**. [CrossRef]

19. Triantafyllidis, A.K.; Koutkias, V.G.; Chouvarda, I.; Maglaveras, N. A Pervasive Health System Integrating Patient Monitoring, Status Logging, and Social Sharing. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 30–37. [CrossRef] [PubMed]
20. Trigo, J.D.; Eguzkiza, A.; Martínez-Espronceda, M.; Serrano, L. A cardiovascular patient follow-up system using Twitter and HL7. In Proceedings of the Computing in Cardiology (CinC), Zaragoza, Spain, 22–25 October 2013; pp. 33–36.
21. Luxton, D.D.; Kayl, R.A.; Mishkind, M.C. mHealth Data Security: The Need for HIPAA-Compliant Standardization. *Telemed. E Health* **2012**, *18*, 284–288. [CrossRef]
22. Cherdantseva, Y.; Hilton, J. A Reference Model of Information Assurance & Security. In Proceedings of the 2013 International Conference on Availability, Reliability and Security, Regensburg, Germany, 2–6 September 2013; pp. 546–555.
23. Velsen, L.V.; Hermens, H.; d'Hollosy, W.O.N. A maturity model for interoperability in eHealth. In Proceedings of the 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), Munich, Germany, 14–17 September 2016; pp. 1–6.
24. Turnitsa, C.D. Extending the Levels of Conceptual Interoperability Model. In Proceedings of the IEEE Summer Computer Simulation Conference (SCSC), New Jersey, NY, USA, 24–28 July 2005.
25. Islam, S.M.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K. The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access* **2015**, *3*, 678–708. [CrossRef]
26. Voskarides, S.; Pattichis, C.S.; Istepanian, R.S.H.; Kyriacou, E.; Pattichis, M.S.; Schizas, C.N. Mobile Health Systems: A Brief Overview. In Proceedings of the SPIE 4740, Digital Wireless Communications IV, Orlando, FL, USA, 1–5 April 2002; Volume 4740, pp. 124–132.
27. Price, S.; Summers, R. Clinical knowledge management and m-health. In Proceedings of the Second Joint EMBS-BMES Conference: 24th Annual Engineering in Medicine and Biology Society Conference and the Annual Fall Meeting of the Biomedical Engineering Society, Houston, TX, USA, 23–26 October 2002; Volume 3, pp. 1865–1866.
28. Martínez, I.; Escayola, J.; Martínez-Espronceda, M.; Muñoz, P.; Trigo, J.D.; Muñoz, A.; Led, S.; Serrano, L.; García, J. Seamless Integration of ISO/IEEE11073 Personal Health Devices and ISO/EN13606 Electronic Health Records into an End-to-End Interoperable Solution. *Telemed. E Health* **2010**, *16*, 993–1004. [CrossRef]
29. Clarke, M.; de Folter, J.; Verma, V.; Gokalp, H. Interoperable End-to-End Remote Patient Monitoring Platform based on IEEE 11073 PHD and ZigBee Health Care Profile. *IEEE Trans. Biomed. Eng.* **2017**. [CrossRef]
30. Harvey, M.J.; Harvey, M.G. Privacy and security issues for mobile health platforms. *J. Assn. Inf. Sci. Technol.* **2014**, *65*, 1305–1318. [CrossRef]
31. Arora, S.; Yttri, J.; Nilsen, W. Privacy and Security in Mobile Health (mHealth) Research. *Alcohol Res.* **2014**, *36*, 143–151.
32. Atienza, A.A.; Zarcadoolas, C.; Vaughon, W.; Hughes, P.; Patel, V.; Chou, W.-Y.S.; Pritts, J. Consumer Attitudes and Perceptions on mHealth Privacy and Security: Findings from a Mixed-Methods Study. *J. Health Commun.* **2015**, *20*, 673–679. [CrossRef]
33. Rubio, Ó.J.; Trigo, J.D.; Alesanco, Á.; Serrano, L.; García, J. Analysis of ISO/IEEE 11073 built-in security and its potential IHE-based extensibility. *J. Biomed. Inf.* **2016**, *60*, 270–285. [CrossRef] [PubMed]
34. Rahimi, M.R.; Ren, J.; Liu, C.H.; Vasilakos, A.V.; Venkatasubramanian, N. Mobile Cloud Computing: A Survey, State of Art and Future Directions. *Mob. Netw. Appl.* **2014**, *19*, 133–143. [CrossRef]
35. Hsieh, J.C.; Hsu, M.W. A cloud computing based 12-lead ECG telemedicine service. *BMC Med. Inf. Decis. Mak.* **2012**, *12*, 77. [CrossRef] [PubMed]
36. Ribeiro, L.S.; Viana-Ferreira, C.; Oliveira, J.L.; Costa, C. XDS-I outsourcing proxy: Ensuring confidentiality while preserving interoperability. *IEEE J. Biomed. Health Inf.* **2014**, *18*, 1404–1412. [CrossRef]
37. Hanen, J.; Kechaou, Z.; Ayed, M.B. An enhanced healthcare system in mobile cloud computing environment. *Vietnam. J. Comput. Sci.* **2016**, *3*, 267–277. [CrossRef]
38. Systematized Nomenclature of Medicine—Clinical Terms (SNOMED-CT). Available online: https://www.snomed.org/snomed-ct/ (accessed on 7 September 2020).
39. Unified Medical Language System (UMLS). Available online: https://www.nlm.nih.gov/research/umls/ (accessed on 7 September 2020).
40. Open Authorization (oAuth). Available online: https://oauth.net/ (accessed on 7 September 2020).
41. Housley, R. Vigil Security CMS. Available online: https://tools.ietf.org/html/rfc5652 (accessed on 7 September 2020).

42. Callas, J.; Donnerhacke, L.; Finney, H.; Shaw, D.; Thayer, R. openPGP. Available online: https://tools.ietf.org/html/rfc4880 (accessed on 7 September 2020).
43. Parah, S.A.; Ahad, F.; Sheikh, J.A.; Bhat, G.M. Hiding clinical information in medical images: A new high capacity and reversible data hiding technique. *J. Biomed. Inform.* **2017**, *66*, 214–230. [CrossRef]
44. Twitter's GDPR Hub. Available online: https://gdpr.twitter.com/en.html (accessed on 7 September 2020).
45. Twitter's GDPR: FAQ. Available online: https://gdpr.twitter.com/en/faq.html (accessed on 7 September 2020).
46. Logical Observation Identifiers Names and Codes (LOINC). Available online: https://loinc.org/ (accessed on 7 September 2020).
47. Unified Code for Units of Measure (UCUM). Available online: http://unitsofmeasure.org/trac (accessed on 7 September 2020).
48. Josefsson, S. Base64. Available online: https://tools.ietf.org/html/rfc4648 (accessed on 7 September 2020).
49. GNU Social. Available online: https://www.gnu.org/software/social/ (accessed on 7 September 2020).
50. Mastodon.social—Mastodon. Available online: https://mastodon.social/about (accessed on 7 September 2020).
51. Diaspora*. Available online: https://diasporafoundation.org/ (accessed on 7 September 2020).
52. Ali, S.; Islam, N.; Rauf, A.; Din, I.U.; Guizani, M.; Rodrigues, J.J.P.C. Privacy and Security Issues in Online Social Networks. *Future Internet* **2018**, *10*, 114. [CrossRef]
53. Twitter Help Center Safety and Security. Available online: https://help.twitter.com/en/safety-and-security (accessed on 17 November 2020).
54. Reddy, S.; Allan, S.; Coghlan, S.; Cooper, P. A governance model for the application of AI in health care. *J. Am. Med. Inform. Assoc.* **2020**, *27*, 491–497. [CrossRef]
55. O'Brien, N. Upcoming Changes to PNG Image Support. Available online: https://twittercommunity.com/t/upcoming-changes-to-png-image-support/118695 (accessed on 7 September 2020).
56. NEMA DICOM—JPEG 2000 Image Compression. Available online: http://dicom.nema.org/medical/dicom/2016c/output/chtml/part05/sect_8.2.4.html (accessed on 7 September 2020).
57. Helin, H.; Tolonen, T.; Ylinen, O.; Tolonen, P.; Näpänkangas, J.; Isola, J. Optimized JPEG 2000 Compression for Efficient Storage of Histopathological Whole-Slide Images. *J. Pathol. Inf.* **2018**, *9*. [CrossRef]
58. Halamka, J.D. Information Lifecycle Management at Beth Israel Deaconess Medical Center. Available online: http://mycourses.med.harvard.edu/ec_res/nt/DD5E7835-72FA-4CFD-9CF8-B0D31113E652/nlm.pdf (accessed on 7 September 2020).
59. Dai, W. Speed Comparison of Popular Crypto Algorithms. Available online: https://www.cryptopp.com/benchmarks.html (accessed on 7 September 2020).
60. Muła, W.; Lemire, D. Faster Base64 Encoding and Decoding Using AVX2 Instructions. *ACM Trans. Web* **2018**, *12*, 1–26. [CrossRef]
61. Deutsch, L.P. DEFLATE. Available online: https://tools.ietf.org/html/rfc1951 (accessed on 7 September 2020).
62. Shevek Parallel GZIP. Available online: https://github.com/shevek/parallelgzip (accessed on 7 September 2020).

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.