Gathering and Managing Complementary Diagnostic Tests

Santos, J. C.[1,2], Pedrosa, T.[1,3], Costa, C.[1], Oliveira, J. L.

jcandido@isec.pt; pedrosa.tiago@gmail.com; carlos.costa@ua.pt ;jlo@ua.pt

[1]DETI/IEETA, Universidade de Aveiro, Campus Universitário de Santiago, Aveiro, Portugal

[2] DEE, Instituto Superior de Engenharia de Coimbra (ISEC), Rua Pedro Nunes, Coimbra, Portugal.

[3] ***Instituto Politécnico de Bragança, Bragança, Portugal.***

Abstract—**Personal health information is constituted in its greatest part by complementary diagnostic tests which are an important medical aid. This information is generated dispersedly because the patient seeks medical care in many different places over his lifetime. Access to a comprehensive set of a patient's health information is a challenge. It revolves around the patient so any managing scheme must be patient-centric. We took a pragmatic approach to this problem and developed a software standalone platform for secure personal health information storage, namely complementary diagnostic tests, on a portable device for mobility. Simplicity and ease of use were main objectives. A special attention was given to the security aspects associated with storing this kind of information.**

Keywords-complementary diagnostic tests, personal health information, security, privacy.

I. INTRODUCTION

One of challenges patients and caregivers presently face is the access to a comprehensive set of patient's lifelong medical information, namely Complementary Diagnostic Tests (CDT).

A significant part of a person's medical information consists of CDTs. These tests are an important part of medical decision support information performed to aid the diagnosis of a specific condition. Examples of these tests are, just to mention a few, x-rays, clinical laboratory tests, Electrocardiograms (ECG), Electroencephalograms (EEG), blood pressure tests, blood glucose tests.

CDTs are a very powerful aid the caregiver prescribes very frequently, therefore generating great amounts of information that, subsequently to the immediate usage, needs to be stored for future analysis. This information is generated dispersedly since most patients receive health services by many professionals, in many different places, throughout their lives.

Often, the proliferation of disperse information leads to the accumulation of CDT results with loose links between them when the contrary should be the case. An example of this situation occurs when a patient, to whom has been diagnosed a certain condition decides to get a second opinion by another specialist in another facility. Both professionals generate clinical information relevant to the diagnosis but they share no formal communication link that allows them to consult the work of the other, therefore leading to data duplication.

At the lack of a better communication channel, it is the patient who usually mediates between healthcare professionals by storing whatever information is made available to him, physically transporting it from site to site, and making it available to caregivers. But this traditional method is susceptible to various mishaps like information loss or poor organizational skills by the patient.

It can be argued that CDT results are, or should, be stored by the institutions that generate them thus overcoming the problem of information loss, but formal communication between institutions is frequently very poor, if existing at all.

By definition, information in the healthcare system revolves around the patient and no single institution is capable of delivering a complete set of a patient's health data at the many possible points of care simply because patients seek care from various providers. The information infrastructure has to be patient-centric. When a patient accesses his health data, he does it in a fundamentally different manner than a professional caregiver. The contexts in which these accesses are carried out are different leading to policies and practices adapted to fit each one of them while ensuring proper authentication, security, interface, and access.

Our approach is a pragmatic one. Many high level and wishful functionalities have been described and deemed as essential parts of Personal Health Records (PHR) [1-2], but perfect is usually enemy of good. Simplicity and ease of use are frequently understated and overlooked factors even while searching for obstacles to the widespread adoption of these technologies.

For all the above mentioned reasons we developed the concept of a personal and portable health information platform oriented to CDTs under the patient's direct control and management. Ideally, the system should contain the most complete possible set of Personal Health Information (PHI) generated by many sources over time in an organized manner. The greater amount of information that should be included in a Portable Personal Health Record (pPHR) comes from CDTs by their own nature.

II. METHODS AND MATERIALS

CDTs exist in two basic formats: paper and digital. Until recently, the paper format was clearly predominant but at present the digital format is in expansion. Particularly the

image tests, ECGs, and medical reports are now beginning to be delivered in digital format.

Independently of the specific type of test and the format under which it is delivered, the patient is still, in the great majority of the cases, responsible for collecting, storing and making those tests available to caregivers thus becoming the main actor in the sharing of information between healthcare professionals.

It's hard to estimate de individual size of digital medical files since it depends on many variables like the amount of time they took to perform and many testing modalities are scalable, but it is common for some exams like computerized tomography to generate files of more than one Gigabyte [3] and similar evolutions are occurring with practically every other diagnosis modality that are stressing the archiving needs, in some cases, by 70% each year [4].

The amount of disk space required for storage of clinical information for each patient is also dependent of the type of patient, age, number and acuteness of medical conditions among others.

In the development of a mobile data acquisition and storage platform, two important aspects have to be taken into consideration: the kind of portable device the information is going to stored in, and the platform that allows the acquisition, archiving, and visualization of the information. These aspects have to be evaluated having in mind the basic requisites of a pPHR [5].

Portability is, by definition, one of them. With our model we obviate two kinds of portability: the personal mobility patients experience by having health services delivered to them in more than one institution, and the data mobility that has to occur in order for that information to be available in more than one healthcare site.

Security is another important aspect to bear in mind since PHI is sensitive and, in case of a breach, can seriously compromise the patient's privacy. Information has to be stored in such a way that only authorized users can gain access to it and with the assurance that it will not be lost or corrupted after some time, this is a person's lifelong information.

The possibility of using the same portable device in many different computers comes at a price. The whole platform, ideally, would be completely self-contained and fully autonomous. Some resources of the computer where the portable device is plugged in will always have to be used, but the least, the better. The interactions between the resident applications and the computer have to be kept to a minimum. The information contained within the pPHR is not meant to be disclosed without express patient's consent.

A. Portable Devices

Since the advent of the diskette, the evolution of portable storage media has led us through a number of devices like the Compact Disk (CD), Zip-drive, Digital Versatile Disk (DVD), USB (Universal Serial Bus), Smart Cards (SC), Flash Device (Pen), and the portable hard drives.

At present, the various portable media types available are very common, practical, secure, and allow for interesting storage capacities combined with personal mobility.

Smart Cards are secure authentication tokens, some of them with cryptographic capabilities, thus providing very interesting functionalities [6]. Some countries are shifting from traditional paper-based national identification cards and passports to smart cards allowing to envision many forms of integration.
The storage capacity of smart cards, although a limitation at present, is on the constant rise but due to all their security features it's a technology worth to keep under close observation in search of future developments [7].

CDs and DVDs are low cost solutions for mass data storage needs with the respective readers/writers being present in almost all the computers sold today. The younger DVD technology is compatible with the older CD technology.

Some technologies in recent expansion like the Blu-ray and High Definition DVD (HD-DVD) allow for 8.5 Gigabytes of storage capacity but are not compatible with each other and are still fighting for the supremacy with one another. For this reason, readers/writers are not standardized and their prices are still relatively high.

Portable hard drives, or external hard drives, available in 3.5 and the 2.5-inch disk drive formats, are also very common both are USB-based devices and the later, due to its reduced dimensions and power consumption, being the most expensive for the same storage capacity.

USB pens are practically ubiquitous since they don't need additional hardware, just the USB port, to function. Its variety of shapes, high capacity, and mechanical resistance are important advantages. They come in all shapes and sizes, some of them very practical for everyday use like necklaces and wristlets, making them very interesting devices for emergency information transport. The 64 GB capacity has become publicly available and 512 GB storage capabilities are under development. Unlike the CD/DVD technologies they are not vulnerable to scratching and since they have no moving parts, they're not susceptible to dust and are much more resilient to physical damage than the later. They can also be bought with added features like hardware cryptography and biometric authentication.

For all the above mentioned reasons, the USB-pen is the most likely candidate for a portable personal health record device [8].

B. Aplication Platform

- Virtualization

Virtualization consists essentially in the separation of the computational resources in different layers and operating environments, allowing for greater flexibility of resource management [9]. There are various types of virtualization: hardware virtualization is achieved through the use of applications that simulate the physical components of a computer, allowing the creation of virtual machines capable of running various Operating Systems (OS) simultaneously;

presentation virtualization where the applications, in reality run on a remote computer and the user has access to the remote application menus thus allowing various users to use the same resources without interference; and application virtualization that can be used to solve incompatibilities that may arise between Dynamic Link Libraries (DLL). This is done creating an intermediate layer between the applications and the OS.

Although our objectives could possibly be achieved through a solution of this general type, there is very few software available in general and none that we could find open source, therefore platform development had to follow a different direction.

- Standalone Applications

There are many Standalone applications available at the moment [10] that don't need installation and can be executed directly from an external drive which is an interesting security feature since no information needs to be written on the computer's hard drive itself [11]. From this perspective, all the external devices that don't allow writing are limited in this respect.

A standalone application just needs the computer's OS and there are portable OS under development.

There are some commercial solutions to transform normal applications into standalone ones [12], but they are somewhat expensive. Another way to obtain portable applications consists of developing a standalone and self-contained software that doesn't depend on other applications or any middleware and does not need the installation procedure. This type of applications requires only on the OS resources.

In a study about usage statistics for Internet technologies , the company Net Market Share, estimates that the Windows OS shares 93.3% of the OS global market, while Macintosh is responsible for 4.77%, Linux takes 1.01%, and others appeal to only 0.52% [13].

Ideally, a standalone application would be developed for all OSs thus allowing full interoperability but given the increase in human effort it would take, and based on the market study above we investigated the availability of software development tools for the windows OS.

Java is one of the most popular programming languages [14] and was developed with portability between OSs in mind so it relies on a Java Virtual Machine (JVM) installed in the computer it's going to be executed on. These characteristics enable the programmer the development of multi-platform applications but there also is a mandatory dependence: the need of a Java Runtime Environment (JRE) in order for the application to run. There are also efforts in the direction of making a portable JRE [15].

Another interesting software development tool in the .NET (dot net) platform that includes a number of languages and interfaces the allow the creation of very versatile applications. Like the Java environment, the *.NET* also implies the dependence of the .NET Framework installed on the computer. This framework has the disadvantage of their various versions being incompatible with each other, thus obligating the installation of the specific platform associated with the application [16].

Delphi is a software development platform, by the manufacturer Borland, that allows the development of multi-platform applications (Windows, Linux) without the need of any other runtime environment or framework installed on the computer in order to run.

- Security Aspects

Personal Health Information (PHI) is very sensitive by definition. A Personal Health Record (PHR) contains information about a person's health and should be confidential, therefore, data security is critical since in case of compromise, the person's privacy is breached. The increasing mobility of people and the portability of clinical information implies that efficient security strategies must be deployed.

Cryptography is a reliable data security technology. Encryption of the whole disk is considered an advantage regarding a device that holds sensitive information. This can be achieved through the use of an open source and free software like *TrueCrypt* [17]. There is, however, a limitation to this software in the fact that, although the portable mode is contemplated by this software, it requires machine administrator privileges in order for the encryption drivers to function. In a usage scenario in which the information contained in a portable device might be accessed by various computers, this is a serious limitation.

A very well-known data compression format is the Zip file that incorporates various files in a single archive. Data can be compressed by built-in compression algorithms or it can be stored in an uncompressed manner. The latest Zip format standard supports up-to-date cryptographic algorithms like the AES-256. Practically all applications are compatible with the Zip format and it is embedded is most recent versions of OS.

The latest format specifications, Zip 64, allow files larger than 4 GB e new compression methods [18]. This file format has the advantage of being capable of both compressing and encrypting data. The disadvantage is the possibility of analyzing its contents (file names, directories, sizes, etc.), therefore the information is not completely hidden but, on the other hand, it is impossible to extract data from within the archive without the coding password.

III. PROPOSED MODEL

We aimed for a device that allows the collection and safe storage of health information and conforms to the current mobility of people. It is a data repository but also an aid that professional caregivers can rely on for better management.

The platform we developed allows easy, secure, and organized access to PHI, specially CDTs, independently of any other software, needing just a computer with an OS. It is a completely autonomous and functional platform.

For the portability purpose the device in which the platform is developed is a USB pen, the Graphical User Interface (GUI) allows the insertion and analysis of data in a simple and

intuitive manner. Organization is crucial since the information should be readily available. Data can only be inserted and accessed through the GUI and only after user authentication thus assuring information security. In terms of a medical emergency situation aid, the is a special data storage area that doesn't need authentication and where data is freely accessible.

From the various professional caregivers' perspective, through the pPHR, they can access all the information they generated and the information generated by others in an organized manner. The different pPHR actors are subject to differentiated authentication privileges but the main user, the patient, has full access to all the information with insertion, visualization, editing and deleting capabilities. Professionals have access to predetermined, by the patient, storage areas [19].

A. *Application Architecture*

The platform is divided in two parts: the Data Repository (DR), and the pPHR applications. This is shown in figure 1.

The DR is implemented by a secure zip archive with password protected access and holds two main storage areas, one public and another private.

The public storage area holds emergency data and the respective DB. Its access is provided via a soft-coded password embedded in the application.

The private storage area holds the CDT repository and BD with private information and CDT indexing data.

The DR was developed with the *ZipArchive* API that implements a set of operations on *"zip"* objects that is compatible with our pPHR needs. The files contained in the archive can be stored in the archive's root or organized in a structure of directories. We chose to store the files in the root.

The *ZipArchive* software [20] was the next choice. It is a library written in C++ with compression and cryptographic capabilities. With this library we were able to create a zip file and to use it for safe storage of the information we needed while maintaining total control over its contents.

In order to have a truly *standalone* application, total development framework independence is needed. Due to the fact that *ZipArchive* is developed in *C++*, we chose *Microsoft Visual C version 6.0* which the last stop before *.NET* technology.

For object identification, the application uses a key, a unique integer that works as an index associated with the file upon insertion. This index is the sole file identifier, other file attributes like name, size or file type are irrelevant for this objective.

Another interesting functionality of this software is that each protected file can be accessed via a unique password allowing for files protected by different passwords, including no password, inside the same archive.

This architecture allows the implementation of different storage areas with different passwords as well as free-access areas and, an important security feature, files inside an archive can be stored with random names thus providing a blind organization structure. Browsing through the archive doesn't display any other information besides the random number identifier.

For the structured CDT repository, we considered a Data Base (DB) to be the best option for simplicity. The choice of software fall upon *SQLite* [21] which a library capable of implementing a transactional, self-sufficient, without server, without configuration needs, SQL (Structured Query Language) DB. Most DB engines are implemented with a separate server process so the applications that access the DB do it by communicating with the server by some kind of process. *SQLite* doesn't work in that manner. Direct DB access is provided enabling reading and writing directly to the disk without any server process in between. This library is written in *C,* free and open source.

The DB is implemented with the *SQLite* library due to the fact that it is portable, light, and has a good overall performance. The DB is distributed by two files, the first of which holds two tables: the personal information and the CDT information table. The second of these files holds just the emergency data table.

The pPHR applications are a set of *.exe* and *.dll* files that implement the GUI and interact with the zip archives and SQLite database with standalone characteristics.

The GUI is the only interface between the user and the stored information. Information manipulation is achieved through the use of DLLs where the access functions to the DB and the zip archive are implemented. The GUI interacts with these libraries creating an abstraction layer between the user and them. There are two conceptual types of files in the application: the ones that are an integral part of the application when it is initially instantiated (the *.exe* file that supports the GUI, and the *.dll* files for information manipulation), and the files that are created by the application after its first execution, these being responsible for storing the zip files and associated information, and DB indexing data.

For the GUI, we chose Visual Basic and its respective development platform, Microsoft Visual Basic 6 mainly because of the standalone features of the files created.

Three access levels are defined for the different data types an user needs, emergency data is freely accessed (no password needed), CDT data can be accessed via a *user-password*, and very sensitive data is only accessible through a master-password. This latter password is also used for managing the information contained in the pPHR.

Data can be imported from various sources, the most common formats being Digital Imaging Communications in Medicine (DICOM) for medical imaging; Standard Communications Protocol for Computer Assisted Electrocardiography (SCP-ECG) for vital signs; and many picture formats plus the Portable Document Format (*.pdf)* for text files. Figure 1 shows these application aspects.

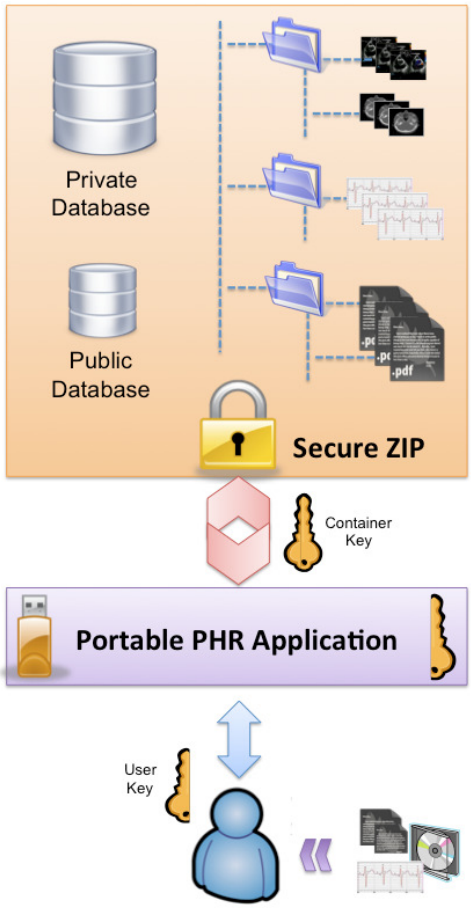

Figure 1. pPHR application and security architecture.

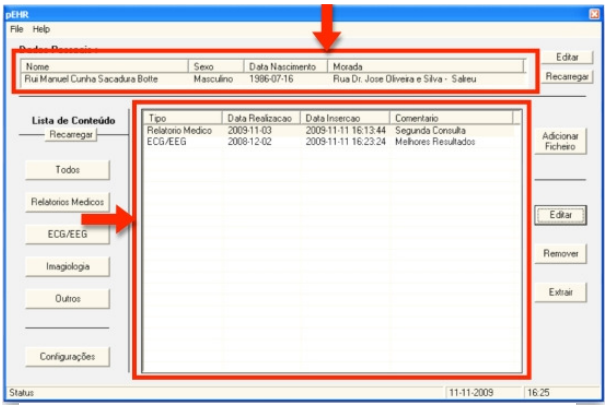

Figure 2. Graphical user interface.

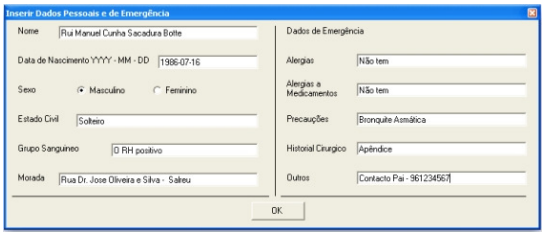

Figure 3. Personal data insertion menu

B. *Security Architecture*

The developed platform assures privacy by storing data inside a safe container, data being encrypted with the AES-256 cryptographic capabilities of the *ZipArchive* software.

In addition, both the files and the structure are blinded by random naming.

Access control is provided by the authentication functionality. The zip archive can only be accessed by a combination of a two-factor password, one of which is factory-supplied and avoids direct access to the container by third-party applications. The other one is supplied by the user ensuring the access control is subject to the patient's express consent.

Secure data management outside the container is possible by extracting the *SQLite* DB to the computer's memory and reading the file directly from memory thus avoiding writing procedures to the computer's hard disk.

IV. Results and Conclusions

We implemented a secure and portable standalone platform for CDTs storage and management with simplicity and ease of use in mind but still complying with all the security, privacy and authentication aspects required by the inherent sensitivity of PHI. This implementation relies on an intuitive GUI, presented in figure 2.

Personal information is inserted into the pPHR through the use of easy graphical menus, shown in figure 3, that carry it out through the use of a set of two keys, one soft-coded in the application and another provided by the user.

The secure zip archive, containing the different data types and DBs is accessed only via the pPHR application GUI that allows free access to the public DB and requires passwords for accessing other information.

A standalone and self-contained implementation of a secure CDT repository is viable. Security and privacy are very important issues when dealing with PHI so we provided the developed platform with control access via password authentication, random naming of files inside the container, and different data storage areas with different access requirements for data scalability under the patient's control.

Other important features of the developed platform include secure encryption provided for the container of information with an option of different access levels for different files; two-factor password protection, one of them factory included and the other user-supplied.

Different access levels (through the use of different passwords) are provided for different types of data. Emergency data is freely available, CDTs and other not very sensitive data are accessed through the use of a normal user-password and management capabilities as well as specially sensitive data access are only granted through the use of a super-user-password (belonging to the patient) who is completely empowered is respect to his health information.

REFERENCES

[1] Kaelber, D., Jha, A., Johnston, D., Middleton, B., and Bates, D., *A Research Agenda for Personal Health Records (PHRs).* Journal of the American Medical Informatics Association, 2008.

[2] Tang, P.C., et al., *Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption.* J Am Med Inform Assoc, 2006. **13**(2): p. 121-6.

[3] Costa, C., A. Silva, and J. Oliveira, *Current Perspectives on PACS and a Cardiology Case Study.* Studies in Computational Intelligence. Berlin. Springer-Verlag, 2007: p. 79-108.

[4] Raths, D., *Information explosion. Advanced imaging technologies are forcing CIOs to rethink their storage needs.* Healthc Inform, 2007. **24**(2): p. 52, 54.

[5] Román, I. and e. al., *Demographic Management in a Federated Healthcare Environment.* International Journal of Medical Informatics, 2006. **75**(9): p. 671-682.

[6] Costa, C., et al., *A New Concept for an Integrated Healthcare Access Model. The New Navigators: From Professionals to Patients.* Proceedings of MIE2003, page 101, 2003.

[7] Costa, C., *Doctoral Thesis: A Security Dynamic Model for Healthcare Information Systems.* Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, 2004.

[8] Srinivasan, U. and G. Datta, *Personal Health Record (PHR) in a Talisman: An Approach to Providing Continuity of Care in Developin Countries Using Existing Social Habits.* 9th International Conference on e-Health Networking, Applications amd Services, 2007: p. 277-279.

[9] Kind, T., et al., *Software platform virtualization in chemistry research and university teaching.* J Cheminform, 2009. **1**: p. 18.

[10] PortableApps.com, *Portable Aplications - Your Digital Life Everywhere™ [Online]. (Accessed 2010-02-10)* 2010.

[11] Pendriveapps.com, *Portable applications that can be run from a usb device [Online].* (Accessed 2010-02-10), 2010.

[12] VMware, *VMware ThinApp - Agentless Application Virtualization Overview.* VMware White Papers [Online] (Accessed 2010-02-16), 2008.

[13] Share, N.M., *Global Market Share Statistics. Operating Systems [Online: http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8] (Accessed 2010.Fev.18).* 2009.

[14] Dedasys, *Programming Language Popularity. [Available Online: http://www.langpop.com]. (Accessed 2010.02.18).* 2009.

[15] Haller, J., *Java Portable - Get Your Java to Go.* Portable Apps. [Online: http://portableapps.com/apps/utilities/java_portable]. (Accessed 2010.02.18), 2009.

[16] Development, M., *Overview of the .NET Framework. Visual Studio Developer Center. [Online: http://www.msdn.microsoft.com/en-us/library/a4t23ktk.aspx]. (Accessed 2010.02.18).* 2009.

[17] Foundation, T., *Free Open-Source Disk Encryption Software.* TrueCrypt [Online: http://www.truecrypt.org/docs/]. (Accessed 2010.02.18), 2009.

[18] Computing, W., *AES Encryption Information - Encryption Specification.* Winzip [Online: http://www.winzip.com/aes_info.htm]. (Accessed 2010.02.18), 2009.

[19] Santos, J., et al., *Modelling a Portable Personal Health record.* Proceedings of the International Conference on Health Informatics (HealthInf 2010), January 20-23, 2010, Valencia, Spain., 2010: p. 494-498.

[20] Software, A., *The ZipArchive Library. [Online: http://www.artpol-software.com/ZipArchive/Default.aspx]. (Accessed 2010.02.18).* 2009.

[21] SQLite, *HWACI - Applied Software Research. [Online: http://www.sqlite.org/index.html]. (Accessed 2010.02.18).* 2009.