A FRAMEWORK FOR SECURE COMMUNICATION OF MOBILE E-HEALTH APPLICATIONS

Burgsteiner H[1], Prietl J[1]

Abstract

Many different applications in health care have some common characteristics. We used these characteristics to develop a framework with which it is possible to rapidly develop and deploy secure mobile applications in the Austrian eHealth context. We implemented mechanisms to be able to securely connect and process medical data according to current legal regulations via a secured communication server acting as a relay between mobile devices and the protected data storage. Additionally, the development of new applications is reduced to the writing of a single configuration file.

1. Introduction

Most applications in the health care sector are based on medical and administrative patient data. The quality of care is heavily depending on having the right and complete data at hand when needed. This is easy when the patient or generally the healthcare customer is attending a medical facility where all of his prior medical data are stored. One of the many possible obstructions of this scenario is that the healthcare customer is immobile or/and that the medical or nursing personnel is on a visit. [10] present some basic mechanisms and technologies of data transfer in a local wireless environment and discuss how to utilize a mobile device in a wireless local area network. When a doctor has to leave the hospital or a practitioner has to leave his office for a visit, one has to use other means of wireless communication, like e.g. GSM or UMTS networks. In [6] several applications of automatic data acquisition and their transfer over wireless wide area communication channels are discussed. The transfer and processing of medical or administrative patient data are strictly bound to legal requirements to protect privacy ([1] and [2]). Additionally, rigid logging of all activities is required. The design of really secure software in health care can be challenging. An overview about the requirements and technologies available can be found in [9], while [3] give a detailed description, how secure health information systems can be designed and analyzed.

When considering developing applications for mobile devices one has to deal with various restrictions like a small screen size that does not allow complex data being display simultaneously and the quality of graphics, drawings or images is very low. In many cases applications are text based only. Also, the input of data is much more difficult than it is at workstations, because most mobile devices do not offer a real keyboard. This has to be considered when designing a graphical user interface (GUI). Another point is, that mobile devices are running on batteries most of the time and hence, the manufacturers are limiting the computational power and the storage capacities of mobile

[1] Graz University of Applied Sciences, Department of eHealth and Health Care Engineering, Graz

devices to a relatively small amount. Therefore, and together with security considerations concerning stolen or lost devices, applications should not store patient data locally. All data will have to be transferred back to a central storage. [6] describe similar considerations in their work.

But the basic requirements are usually not very different from one such application to the other. In our work we tried to unify this process, to be able to rapidly develop and deploy new applications for different but similar applications. We will show in the methods section that these similarities in fact can be viewed as restrictions through which it becomes possible to use one single framework for multiple applications and hence reduce the development process to the writing of a single configuration file. A more detailed discussion of this framework can be found in [8].

2. Methods

eHealth applications should be able to run on a mobile device that can communicate over a public cellular phone network. Using such a network still imposes some performance issues. Response times and data transfer rates are very limited, especially when being used in rural areas. This is an issue to be dealt with as we do not save any data locally and deny any off-line activities: a user has to be on-line and authenticated to be able to access any data. After a short evaluation of available programming frameworks for mobile devices, we decided to use a windows based PDA with integrated GSM and GPRS functionality. This is a similar setup like it was used in [6]. This configuration was chosen because of the availability of a development environment with appropriate libraries for mobile devices. We used the .NET framework for some server applications that will be explained below and the .NET compact framework for the application running on the handheld device.

2. 1. Data model and configuration

We summarized basic technical functionalities of eHealth applications in 3 groups:

- Data queries: based on some primary data like patient name or social security number, additional data (medical history, administrative data, etc.) must be looked up in a database
- Data manipulations: new data must be added to a database, existing data has to be updated or data must be deleted.
- Special functionality: these are necessary procedures that are not associated with a database, itself, but require tasks that have to be executed at the site of the database server. These are e.g. consultations of the eCard system to check the insurance status of a patient, etc.

Furthermore it is interesting to observe, that many applications use very simple data models. Two simple categories are (i) orphaned tables, i.e. they have no references to other tables in a database and (ii) simple master/slave relationships. In this case, a single master table serves as the master ID holder for some other tables. Here, we restrict ourselves from more nested data models where e.g. a slave table could act as a master table to further slaves, to be able to handle the complexity of data display. Note that this is due to the limited display capabilities of handheld devices. This does not imply that the framework can not handle such data dependencies.

We abstracted these requirements and developed a .NET client application for a mobile device which can be controlled by a configuration file. I.e. a single configuration file which is based on the XML standard is responsible for each specific application in our framework. This configuration file can be validated against a DTD and contains three basic categories of information:

- Setup of the elements of the GUI which can be seen in *Figure 1*. This includes the application title, which tables should be displayed in which tabs, how these tabs are called, etc.
- Layout and functionality of the control elements of each tab. Each attribute of a table is mapped to a control element that is suitable to display, input or edit the attribute of a certain table. Available control elements include e.g. text areas for long text attributes, drop-down elements for a predefined set of values, check boxes for Boolean values, etc. Additionally a label and the read-only attribute can be set.
- Configuration of the database interface. In this part of the configuration file database queries can be mapped to certain control elements. A typical example is a list of ZIP-codes that is the result of a SQL-query which is then mapped to a drop-down list.

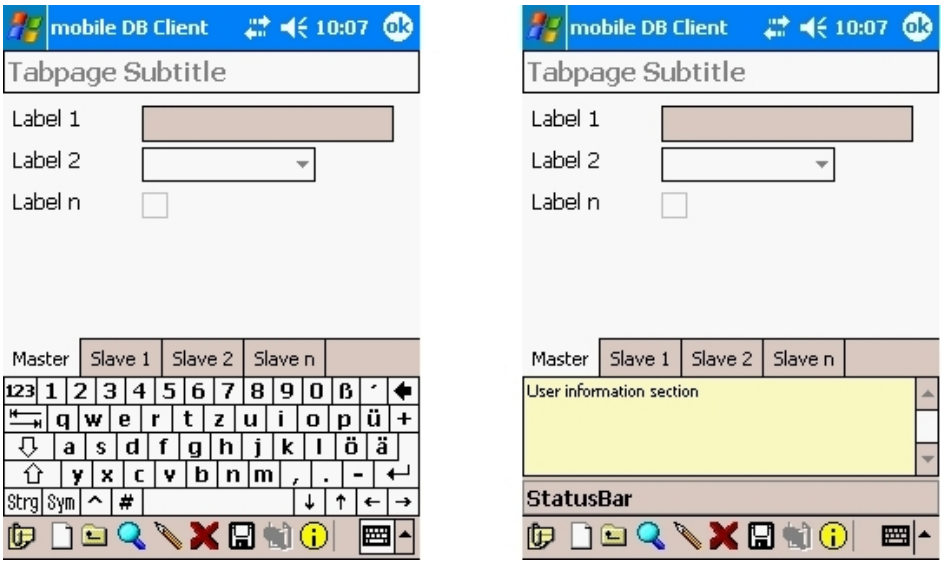


Figure 1: Generic design of the user interface with and without a virtual keyboard. The master/slave relationships are displayed with different tabs. Input and output elements are chosen according to the data type of each attribute in the table.

Hence, to "develop" a new application one can write a configuration file, download it to the handheld device and the client application uses it to generate the user interface and connect to a database. This gives us an easy way to rapidly setup new applications that conform to the restrictions described above. In our laboratory we eventually stored these definitions of applications in a database and used a tool to generate the XML file out of the values from the database. A summary of this process can be seen in *Figure 2*.

2. 2. Network and system architecture

Due to security considerations and legal restrictions (see e.g. [1] DSG2000 §14(1) and [2] GTelG §§4(4),5(1),6(1) and 7(1)) one must not place a database server storing unencrypted critical patient data in a public network like the internet. Furthermore, medical applications usually have to access data that is physically located within a doctor's practice or in a hospital network, which is (hopefully) additionally protected by a firewall that prohibits access originating from outside the network to the inside. Other applications like the ones described in [5] and [7] use a communication server placed in a demilitarized zone with an open firewall port for communication. Another option would be e.g. to use a virtual private network (VPN) to allow clients to connect to an internal network. At the time of the development of our framework no VPN-clients that run on the Windows mobile

edition were offered. Due to our security considerations and the ease of installation of the software, we do not want to have to reconfigure or open any port on a firewall. One possible solution is to use a communication server located in a public network that can be contacted by both, the database server within the protected environment and the mobile device. *Figure 3* outlines such a scenario.

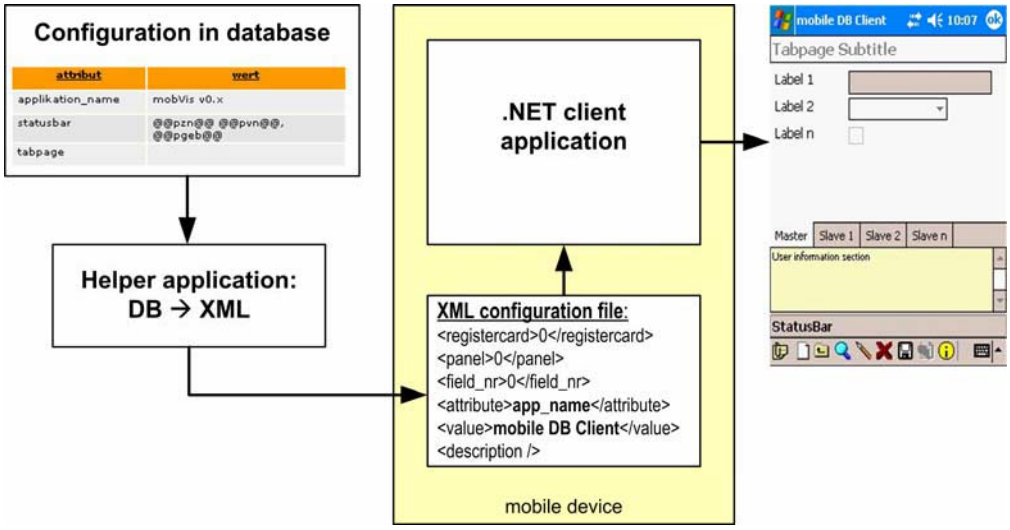


Figure 2: Rapid application generation in the framework. Application descriptions stored in a database are used to generate XML configuration files. These are used by the .NET client application to setup the application.

Two web services provide an open connection for communication. Typically an application on a mobile device sends a request to its web service on the communication server. The web service stores this request in a local database. Since this server has no means to contact the protected database server, queries have to be pulled off frequently. This is done via the server web service. Both web services can only be contacted over a secured hyper-text transfer protocol, i.e. TLS or SSL. Additionally, the data can be encrypted by the mobile device or the database server to provide end-to-end security in case the communication server is not trustworthy. Authentication at the web services and end-to-end encryption of data can be handled with X.509 certificates and generated session keys with symmetric encryption respectively. According to [1] and [2] this is an adequate protection for medical data, when we restrict the usage to applications where medical personnel accesses data which they need to fulfill their professional duty. In many cases, the usage of certificates with certain minimal demands is obligatory. In other cases when one is "directly accessing a data application from the distance" certificates can be omitted and a standard user/password mechanism can be used (see GTelG [2] §4(4). In this case the authentication must be repeated periodically, which can easily be implemented with a reasonable session timeout.

3. Results

We tested our framework through the implementation of a simple mobile pharmaceuticals information system (mPhIS) within our framework. mPhIS allows a remote access to the list of available drugs in Austria. This list is provided on a regular basis by the association of social insurance carriers. It was installed on a database server behind a firewall and an application description file was created. *Figure 4* shows a small part of this application's description file and the corresponding screenshot.

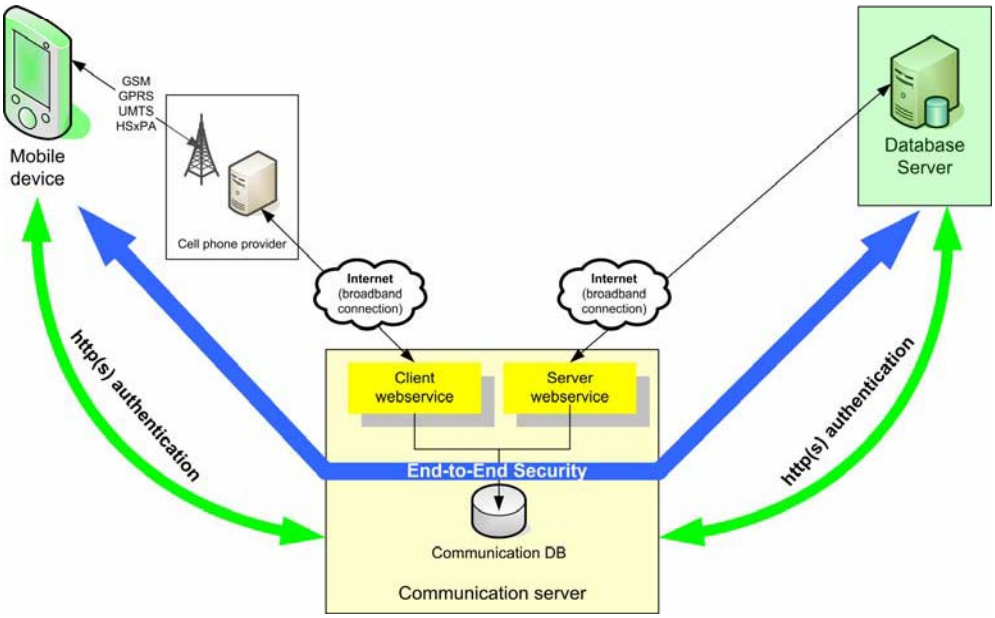
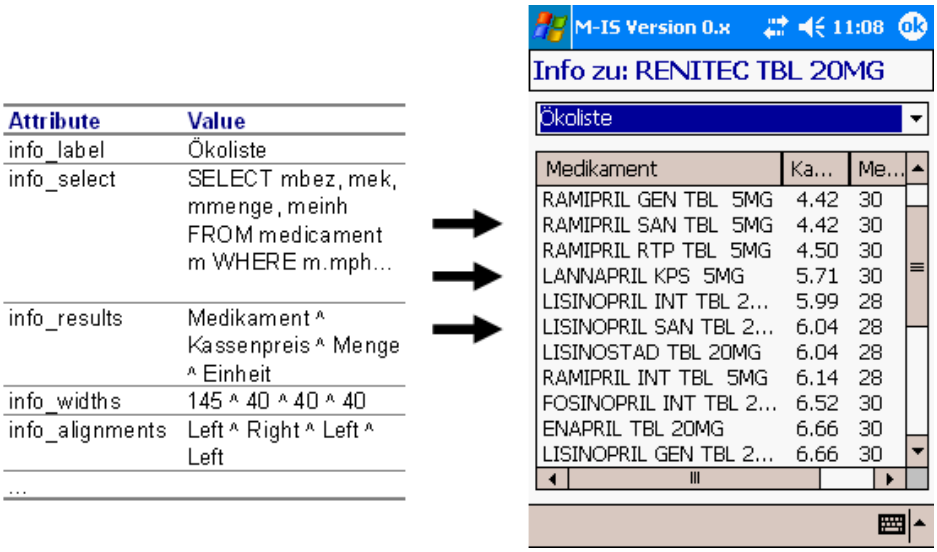


Figure 3: Complete system architecture. Client and database server interact via web services on a central communication server. Data security is provided with a TLS-secured HTTP. Additionally data can be encrypted on client and database side to provide high grade end-to-end security.

According to our experiences the biggest problem with mobile applications seems to be the latency between queries sent to the database and the answers one receives, even when the database server polls for outstanding queries every second. This latency is especially high for the first request of a new session, when the network subsystem gets initialized and the mobile user is authenticated at the client web service. Even for low amounts of data we measured delays of about 3-5 seconds, while the first request took up to 20 seconds. This is a general problem of the cellular phone technology in comparison to wireless local networks. While we used GPRS for our tests, one can expect to shorten the latency by up to some hundred milliseconds when using UMTS or HS-xPA.



Attribute	Value
info_label	Ökoliste
info_select	SELECT mbez, mek, mmenge, meinh FROM medicament m WHERE m.mph...
info_results	Medikament ^ Kassenpreis ^ Menge ^ Einheit
info_widths	145 ^ 40 ^ 40 ^ 40
info_alignments	Left ^ Right ^ Left ^ Left
...	

Figure 4: Section of the application description (lhs) and the corresponding screenshot of the mPhIS application displaying data about alternative pharmaceuticals from a central database (rhs)

4. Discussion

Mobile applications are intended to support medical decisions on location. Various problems have to be solved to provide adequate security mechanism according to Austrian laws. The framework presented in this article is promising rapid development of certain classes of applications conforming to the standards described in the previous sections. Simple database entries, a generated XML configuration file and a small .NET client application suffice to provide mobile eHealth applications on handheld devices. This enables organizations to quickly supply users with new mobile applications for eHealth purposes without having a full software development cycle. Some issues could not be satisfactorily solved in this version of the framework. First, the available .NET compact framework which has to be used on Windows mobile edition, lacks support of handling X.509 certificates stored in the system's certificate storage [4]. This feature would enable one to separate the mobile applications from the certificate management. A further drawback is that the .NET compact framework also does not provide http authentication via X.509 certificates. Certificates from an official certificate authority could be used for authentication purposes. This would enable mobile users to legally access data located in the GIN, not only in secured other networks. In the current version of our framework, user authentication was done with a TLS/SSL secured user/password mechanism only. In our concept, a single communication server can provide the corresponding web services for the client and the server for multiple users and multiple applications simultaneously. All queries and results include IDs indicating the originating or intended user and application. This concept still has to be tested in future applications for performance and security. But even if one could trick the communication server into sending data of other users or applications, this would not compromise the privacy, since all communication is secured from one end to the other with strong standard cryptographic algorithms.

5. Acknowledgments

We thank Alwin Günzberg and Hannes Sattler from ALAG GmbH for their support of this work.

6. References

[1] BGBl: Bundesgesetz über den Schutz personenbezogener Daten – Datenschutzgesetz 2000 (DSG2000) vom 17.8.1999, BGBl 165/1999.

[2] BGBl: Bundesgesetz betreffend Datensicherheitsmaßnahmen beim elektronischen Verkehr mit Gesundheitsdaten und Einrichtung eines Informationsmanagement (Gesundheitstelematikgesetz) vom 30.12.2004, BGBl 179/2004.

[3] BLOBEL, B. and ROGER-FRANCE, F.: A systematic approach for analysis and design of secure health information systems. International Journal of Medical Informatics, Volume 62, pp. 51-78, 2001.

[4] DRÖGE, R., NOWAK, P. and WEBER, T.: Programmieren mit dem .NET Compact Framework. Unterschleißheim: Microsoft Press, 2006.

[5] DUNCAN, R.G. and SHABOT, M.M.: Secure Remote Access to a Clinical Data Repository Using a Wireless Personal Digital Assistant (PDA). Proceedings of the AMIA Symposium 2000, pp. 210-214, 2000.

[6] ISTEPANIAN, R.S.H., PATTICHIS, C.S.E. and LAXMINARAYAN, S.: M-Health: Emerging Mobile Health Systems. USA, New York: Springer Science+Business Media, 2006.

[7] MENDONCA, E.A, CHEN, E.S., STETSON, P.D., MCKNIGHT, L.K., LEI, J. and CIMINO, J.J.: Approach to mobile information and communication for health care. Int. J. of Medical Informatics, Volume 73, pp. 631-638, 2004.

[8] PRIETL, J.: Entwicklung eines generischen PDA-Systems als Grundlage für mobile Anwendungen im Gesundheitswesen (Thesis). Graz: Graz University of Applied Sciences, 2007.

[9] SMITH, E. and ELOFF, J.H.P.: Security in health-care information systems – current trends. International Journal of Medical Informatics, Volume 54, pp. 39-54, 1999.

[10] WALUYO, A.B., HSIEH, R., TANIAR, D., RAHAYU, W. and SRINIVASAN, B.: Utilizing push and pull mechanism in wireless e-health environment. Proceedings of the IEEE Conference on e-Technology, e-Commerce and e-Service, pp. 271-274, 2004.

[11] YU, P. and YU, H.: Lessons learned from the practice of mobile health application development. COMPSAC 2004. Proc. of the 28th Annual Int. Computer Software and Applications Conference, 92(2), pp. 58-59, 2004.