Enforcing Security in Pervasive Healthcare Monitoring Gestational Diabetes Mellitus

Stefano Bromuri, Johannes Krampf, René Schumann, Michael Ignaz Schumacher
Institute of Business Information Systems,
University of Applied Sciences Western Switzerland,
Emails: stefano.bromuri@hevs.ch {johannes.krampf, rene.schumann, michael.schumacher}@hevs.ch

***Abstract*—Life expectancy is rising world wide thanks to the current advancement of medicine. Due to the fact that the population is growing old, also the incidence of chronic illnesses in the population is rising. For this reason, new paradigms of healthcare are being developed to achieve a better medical follow-up and also handle the rising costs. One approach that is proving successful is telemedicine, which focuses on decentralising the delivery of healthcare by means of new technologies based on network connectivity. One problem that rises in the definition of telemedicine systems is the one of security of medical data. In this paper we present our telemedicine system for monitoring Gestational Diabetes Mellitus (GDM). We addressed the problem of securing the communication between the patients and the doctors. The result is a fully implemented telemedicine system for GDM that mitigates the risks associated with the most common malicious attacks directed to a distributed system.**

Keywords-Telemedicine; Gestational Diabetes; Security; Personal Health System.

I. INTRODUCTION

The life expectancy is rising world wide thanks to the availability of new and higher standards for healthcare, but to this improvement a decrease in the incidence of chronic or permanent health conditions [1] did not follow. The world expenditure in healthcare is surging due to the wide spread availability of high standard care. This creates new challenges for healthcare professionals. Also new trends in technical development enable new services that allow to improve care even more.

In particular, we are addressing the issue of collecting and evaluating medical data by means of telemedicine. This allow healthcare professionals to have more accurate data. By pro-actively notifying medical experts they can react faster to a changes in the condition of a patient. Further more patients can benefit as well, because they can live their file with more freedom, following their daily activities.

Healthcare activities can be grouped into three categories: measuring physical values, diagnosing and administering therapies. These activities can be described more technically as monitoring, recognizing, and decision making. In our research we are going to set up a common pervasive healthcare infrastructure that aims to support all these activities. Here we report on the architecture for the pervasive healthcare monitoring framework, that addresses the first category of activities. Therefore we are going to set up a personal health system (PHS) that integrates the patients as actors into the monitoring process. We are doing so to obtain more accurate data, which in consequence allows medical services to provide better services to the patients.

Patients collect their physiological data either on their own or by using smart devices, e.g. in form of wearable computing devices forming a body area network that collects physiological data autonomously. The physiological data needs to be collected and eventually augmented with metadata, like the data origin, although it is not enough to simply store this data. The monitoring process covers a first data processing step, which is a filtering to identify abnormal conditions. If such a condition has been identified a medical expert has to be notified. By this notification, the medical expert gets supported, because a) more data is available and b) his attention is drawn to the cases where the data indicate an abnormal condition, and an action from him might be required. This supervision of incoming data is the core of the monitoring activity. Monitoring can use a reasoning component that evaluates the incoming data and checks it, respecting the context and the history of the patient.

It goes without saying that the design of a pervasive healthcare monitoring framework, as well as the entire pervasive healthcare infrastructure, has high requirements towards the security of those system, as they deal with highly confidential personal data.

In our current study, we are addressing patients suffering from Gestational Diabetes Mellitus (GDM). GDM occurs during pregnancy due to increased resistance to insulin. GDM is a type of diabetes which temporarily affects 4% of otherwise healthy pregnant women, and typically disappears after delivery. As relatively milder hyperglycemia can cause adverse effects in the baby and in the mother, then cases of glucose intolerance in pregnancy are also considered to represent GDM. Current GDM care consists in a routine check once per week, meaning that in between these checks, the woman can develop poor glycemic control and further adverse effects. GDM is not a typical chronic disease, where patients diverge from the healthcare plan over time. In contrary women suffering from GDM are typically very engaged

 ISBN: 978-1-61208-179-3

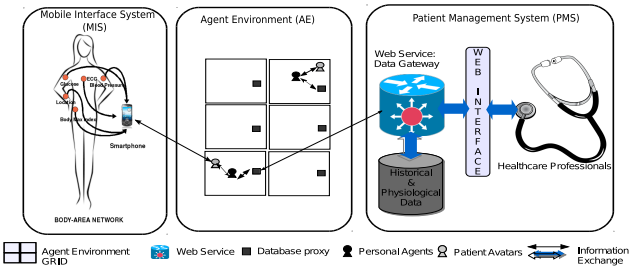

Figure 1. Components of the GDMM system

in their course, as their health, as well as the health of their child is effected. Also it is desired to let the women live their normal day-to-day activities, therefore a telemedicine system is preferred. Thus women suffering from GDM have a high motivation to participate in a monitoring of their condition. In fact, the women collect their physiological data, like weight, blood sugar, blood pressure, with conventional measuring devices and the data is transfered to the pervasive healthcare monitoring system using a special application on a smart phone. On the server side the data is stored and a reasoning component scans for anomalies or threats based on the data collected so far, and additional information given by the doctors in charge. If this component detects a possible threatening condition it notifies a medical expert. Caretakers can interact with the system with a web front-end. They can analyse the entire data collected from a particular patient, can see and react on notifications, and can update the current treatments of the patient. The main tasks of this monitoring system is to collect and provide data and to hint medical experts to possible interesting data. The medical activities of recognition and acting upon it, i.e. naming a therapy, are not addressed by this system, as these activities are reserved to medical experts. The overall process and main components are shown in Figure 1. We refer to the overall system developed in our project as the Gestational Diabetes Mellitus Monitoring (GDMM) system.

In this paper we will present the architecture of the GDMM system (Section II) and discuss how security has been addressed (Section II-B. Then we highlight related work and compare our approach to existing ones concerning the implementation of security issues in Section III. Finally we conclude our paper and give an outline to future work.

II. Architecture of a Pervasive Healthcare Monitoring System

Here we describe the components of a pervasive healthcare monitoring system and its security mechanisms.

A. Components of a pervasive healthcare monitoring system

In the previous section we have outlined the intended usage of a the GDMM system and the requirements towards the system. Here we propose the architecture for a system that will satisfy these needs. Despite the fact that it has to cope with different devices used by different groups of users, from a broad perspective such a system is a web-based application and it has to be able to manage different devices and roles users can have. Currently the medical staff, like doctors and nurses, will use a web-based interface to interact with the system. The patients will send their physiological data via smart phones. Nevertheless, the communication devices are not strictly bound to user groups. So a mobile application for doctors can be considered, as well.

Figure 1 shows that our system is composed of three main components, which are the *Mobile Interface System* (MIS), the *Agent Environment* (AE) and the *Patient Management System* (PMS). Furthermore, these components are interfaced between each others by means of a mediator component, realised as a Web service Data Gateway connector that accepts HTTPS requests. The MIS component collects the physiological data of the patient and delivers such data to the AE component and to the PMS component. The AE component utilises logic programming to model intelligent agents that filter the data submitted to the PMS and rise alerts in case of significant events, such as a possibility of preeclampsia in the patient or a high level of blood sugar that requires a treatment adjustment. Depending on the dynamic load, the AE system is subdivided into multiple instances where the patients connect with their smart phones to transmit their physiological data, that are then evaluated by intelligent agents. The patients are represented in the AE as *avatars* that can communicate to a personal intelligent agent, embodied in the AE. A personal intelligent agent exists for each patient. This agent is responsible to monitor her condition. This allows us in future extensions to specify individual strategies about when to notify a medical expert. Furthermore, it enables us to respect general medical guidelines but also to handle individual deviations from the standard procedures for each patients. Finally, the PMS allows the doctors to visualise the patient's data, and to visualise the alerts produced by the AE. The three tier logic architecture shown in Figure 1 translates then to a four tier architecture as shown in Figure 2.

We use a design pattern described by Meier et al. [2] for the design of the monitoring system. This patterns extends the well-established 3-tier architecture for web applications in two ways. First, it advocates for a fourth layer, taking the client device into account. It proposes to distribute the overall system into specific clients, web servers, application servers and databases. Secondly, it suggests for an additional layering within the application server.

The web interface for the medical staff can be accessed by a current standard web-browser. In contrast we need to create a specific application for the Android smart phones to allow patients to send their physiological data to the system. These components form the first layer in the *client*. More details about these user interfaces can be found in [3], [4].

 ISBN: 978-1-61208-179-3

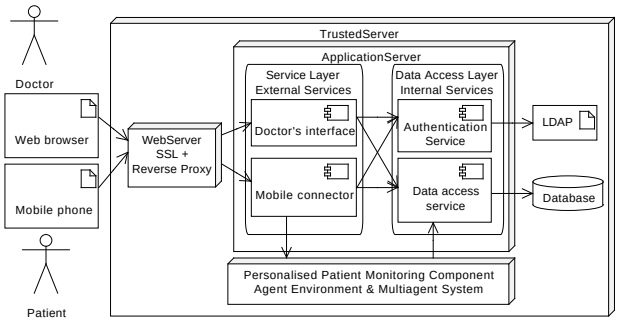

Figure 2. Architecture of the GDMM system

For the interaction with the backend system we use a REST architecture style [5], which forms the second layer. In the layering of the *application server* we have deviated form the 4-tier pattern by externalizing the business logic into the AE. The agents subscribe to the information produced by the patients and then the agent stores the incoming data (glucose, blood pressure, pulse, weight and symptoms) in the database. The rules for monitoring the patient are encoded as logic rules that produce alerts according to the physiological values of the patient. To define such rules we use two approaches, one based on deductive reasoning, specifying that if a set of glucose events are out of the boundaries, then an alert for treatment adjustment has to be risen, and one based on abductive reasoning, where, given a set of observations related to the symptoms of the patients the agent sends an alert with an explanation on the current status of the patient. The details of the rules used and the application of abductive logic is explained in more detail in [4]. Furthermore, to handle resources efficiently the agent itself can be serialized and stored in the database.

Finally, in the fourth layer the data is stored in a database to allow for efficient and persistent storage. Here we use a Postgres database. For authentication we can either use a database or an existing LDAP service.

The entire GDMM system has been fully implemented and first scalability analysis have been performed. According to our tests the system performs well. With a single instance of the system on a single machine we would be able to monitor up to several hundreds women suffering GDM [3].

B. Enforcing Security in the pervasive healthcare monitoring framework

Security is a central aspect when dealing with personal and medical data. We want to ensure confidentiality, which means that no one but the caretaker of a patient and the patient itself should be able to access the medical data. To create a secure system, extra effort has to go into modelling the interaction amongst the different components, securing the stream and storing of the data. To build our system, we followed the following security principles as defined by the Open Web Application Security Project (OWASP) [6]:

- We keep security simple, preferring simple security solutions over complex ones to reduce the potential for errors.
- We minimise the attack surface area, giving to an attacker as few attack opportunities as possible.
- We follow a *positive security model*, or white listing, which restricts values or actions to pre-defined elements. This is contrary to black listing, which allows all values or actions except those which are forbidden.

Generally speaking, a doctor is only allowed to access patient data if there exists a treating relationship between them. User's permissions are determined by their group membership and treatment relationships. Direct outside access to the server is restricted to secure HTTP. There is no mail server or other service running on the machine which could be vulnerable and lead to outside intrusion.

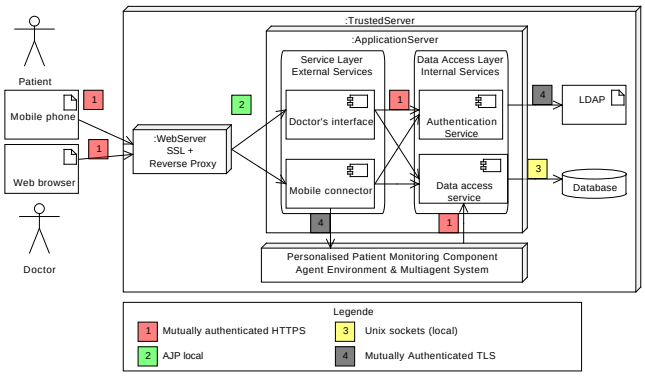

Figure 3. Security of the GDMM system

Figure 3 illustrates the details about how security is defined in the GDMM system, that we will explain in details later. To define this architecture we took into consideration the top ten list of security risks [7] for Web applications. Then we will show the proposed security interfaces to tackle these risks and we will describe their functionalities. This approach of validating a security of the architecture has also been used by Maji et al. [8].

- **Injections** can lead to unwanted code execution by insufficient input validation and escaping when creating commands between layers of a system.
- **Cross site scripting (XSS)** is a special case of code injection where an attacker can input HTML code which will be directly included when generating a page.
- **Broken Authentication and Session Management** refers to vulnerabilities related to authentication and session management.
- **Insecure Direct Object References** are requests using a user changeable object identifier which is not verified.

- **Cross-Site Request Forgery (CSRF)** denote malicious requests, which make use of an existing authentication token to perform requests on the user's behalf.
- **Security Misconfiguration** results in security problems due to outdated or wrongly configured software.
- **Insecure Cryptographic Storage** refers to breakable or circumventable cryptographic data protection.
- **Failure to Restrict URL Access** is a missing access control for restricted pages, which allows users without permission to access the pages.
- **Insufficient Transport Layer Protection** can cause an attacker to read passwords or sensitive data by monitoring traffic.
- **Unvalidated Redirects and Forwards** can be manipulated by an attacker to redirect to a malicious page while originating from a trusted page.

To avoid the security issue listed above, we defined a security layer at every interface of the system.

1) Security in the Mobile Phone and in the Web Interface: We have developed a mobile client as an App for the Android OS, we require at least version 2.3.3. In this App we have to make sure that the patient and the smart phone are authenticated with the system. For this purpose we use a double mean of authentication. To authenticate the phone, we store an encrypted certificate within the smart phone in binary format. Then, we provide our patients with a QR code, containing the keys for the certificate store and for encryption on the phone. This is described in Figure 4.

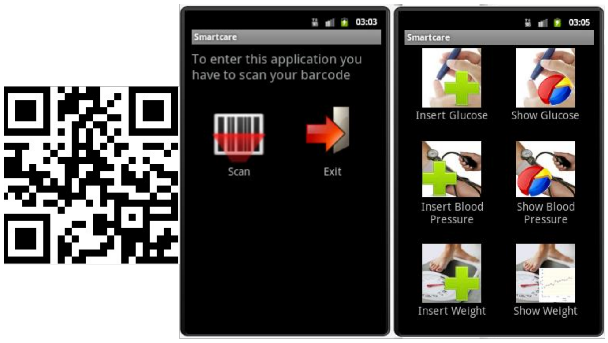

Figure 4. Bar Code Scanning in the Mobile Phone Client.

Consequently, in order to steal the identity of a patient, it is not enough to have the handheld device, it is also necessary to have the barcode. Through this double mean of authentication we aim at minimizing the risks of *Insufficient Transport Layer Protection*, *Unvalidated Redirect and Forwards* and *Insecure Cryptographic Storage* with respect to the smart phone client. In particular, the protection on the transportation layer is achieved using a mutually authenticated HTTPS connection between the smart phone and the system that makes use of signed certificates, this also minimize the risk of unvalidated redirect and forwards weaknesses. The storage within the smart phone again depends on the key stored in the QR code, consequently if the smart phone is stolen, without the QR code it is not possible to access the data of the user.

To authenticate the caregivers when accessing our GDMM system, we utilize user name, password and a certificate as the two authentication means. The certificate is stored on a smart card. Such a certificate is used to open a mutually authenticated HTTPS connection with the PHS. A limitation of our approach occurs when *CSRF* attacks have to be handled. Currently we cannot completely handle those attacks. One way to mitigate this could be the usage of a one time password, including the current time into the computation of the password. This will limit the validity of authentication tokens that can be used to produce data in the system from both of the patient and caregiver sides. Finally, *XSS* attacks are handled by sanitising the input that caregivers and patients can introduce in the system.

2) Security in the Server side: Figure 5 shows how we secured the distributed agent platform by means of a TLS transportation layer and HTTPS connections.

To secure the agent environment, we need to secure all the interfaces with the external world. As far as it concerns the *Data Layer*, this resides on an encrypted partition, consequently if access on the Data Layer is gained maliciously, to access the data it would be necessary to know the key for the encryption. This ensures that we can minimise the risks of attacks performed by *injection* or *insecure cryptographic storage*. Furthermore, every node of the agent environment, where the personal agents of the patient are deployed, contains a keystore and a truststore. The keystore contains a certification for the node of the agent environment which is used to open TSL connections to the other agent environment nodes. Similarly, when a mobile client connects to our GDMM system, it first opens a secure HTTPS connection with the mobile connector of the GDMM system, exchanging certificates with it. When the mobile client is authenticated, the mobile connector opens a TSL connection to the agent environment, by exchanging certificates, which minimises the risk that attacks based on *broken authentication and session management* are successful with respect to the communication performed by the smart phone client and the agent environment. Also, the use of certificates, minimises the risk that an attacker can exploit *insecure direct object references* as the connections between the different entities in the system are all authenticated, mitigating the risk of a middle-man attack. Additionally the right management system ensures that clients can only see data they are entitled to see, mitigating the risk of URL manipulation. Using truststores and keystores improves the security of the system also with respect to *security misconfiguration* issues as the certificates have to be properly set up in order to have a meaningful communication amongst the components of the system. Finally, *injection* attacks by injecting agents into the agent environment are particularly

 ISBN: 978-1-61208-179-3

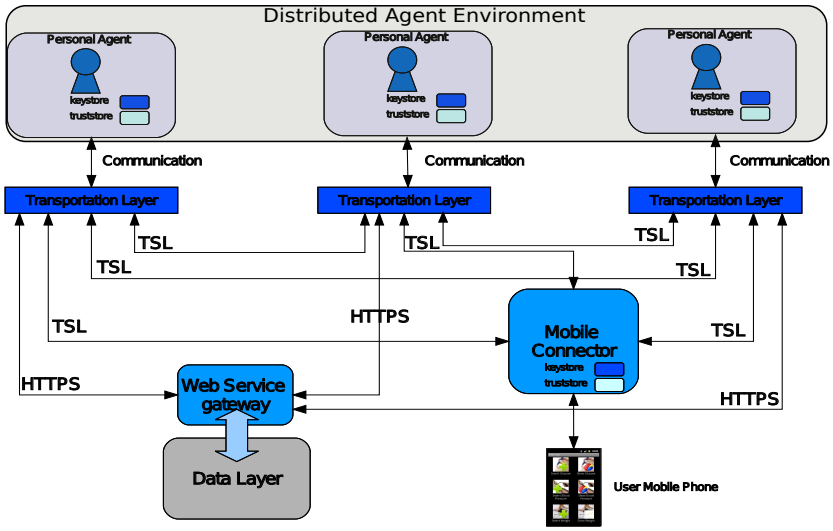

Figure 5. Security in the Agent Environment APIs

difficult as the only mean of communication with a node is the transportation layer, meaning that to inject an agent, the attacker should have a legitimate certificate and know the private key of one of the internal components of our framework. Other injection attacks, like SQL injections, are not possible, because we apply the white listing security approach, allowing only pre-defined types of data as inputs.

III. Related Work

Orwat et al. [9] have presented an extensive survey about research performed in pervasive healthcare. They have in total reviewed publications from 67 different projects worldwide. According to their survey the monitoring of patients is a heavily addressed field (in 63% of the projects surveyed). Also the usage of common mobile devices is recognised as a common way (51% of all projects). Automated monitoring and alerting experts is addressed by 46% of the projects. So the functionality described here are in the core of pervasive healthcare systems research. Even though the potential values of such applications is widely recognised it is quite surprising that the issue how to design those system in a secure way, has not been adequately addressed. This impression is supported by the review from Isern et al. [10]. They describe several agent-based monitoring projects aiming to monitor the state of patients with different degrees of centralisation. One problem of those projects is that they did not focus on security aspects. The pervasive healthcare monitoring framework, presented in this paper, tries to narrow this gap, by embedding concepts needed for an individual monitoring of patients into a framework for secured web-based applications.

As pointed out before the body of work using pervasive healthcare to monitor patients is considerable large. Even though two aspects are interesting. First, to the best of our knowledge the GDMM system is the first one that addesses GDM. So far only diabetes type I and II have been addressed, in particular type I, e.g. by Farmer et al. [11] or in the DiaBetNet project [12]. Second, the aspect of securing such PHS is not broadly addressed, so far. For instance Orwat et al. [9] found only in 11 publication references to the problems induced by privacy and security concerns to PHS. And only few of them discuss how to address those issues. We will focus here on these and more recent work, addressing the security issue in PHS.

A number of researchers used mobile phones to implement pervasive healthcare systems. Therefore securing GSM and WAP connections has been focused e.g. by [13], [14].

Systems, like ours, which taking advantage of the IP based communication use well-established approaches, based on encrypted communications via HTTPS, e.g. [15].

Toledo et al. [15] have presented a platform for chronic care for patients suffering from chronic obstructive pulmonary disease (COPD). The authors performed a field test with 157 patients, connecting the patients to medical experts in a call centre. Therefore they use an electronic chronic patient record that can be accessed by the persons in the call centres but also by the caretakers while they visit the patients. So different devices needed to connect to a centralized server. The authors presents an extended security concept. Different devices had to be integrated. For all devices connection are SSL encrypted, e.g. using the HTTPS protocol. Specific services can only be accessed from the Intranet or via a VPN access. Also token and certificates are used to secure the connection between mobile services. Also user have to identify themselves, using login and password information.

Salvador et al. [13] aim to monitor cardiac patients. Patients interact with a mobile phone with a centralised

 ISBN: 978-1-61208-179-3

service. Consequently considerable efforts have been but into securing the mobile connection based on GSM build in security and WAP sessions. Patients are served by personal healthcare agents (note here these are real medical experts). These experts need access to the patients data. They are connected via a secured internet connection. Also medical data is transmitted in an anonymous way, where the id of the patient is a shared secret between sender and receiver, to foster privacy aspects. Of course also an authorisation and right management has been implemented to control the access of the healthcare agents. Both user groups has to identify them to the system using a login/ password pair.

Maji et al. [8] have presented a four tier architecture for web-based telemedicine applications, too. Instead of adding specific client components, they have added a web proxy layer in front of a firewall. This proxy separates the server side application from the internet. From a functional point of view the web proxy layer allows to separate the session handling from the generation of the device depended presentation of the content. While this is an interesting idea for separating the different tasks located in the presentation layer of a conventional three-tier web architecture, this is quite similar from the security perspective to what we can achieve by using a reverse proxy in the web server.

IV. Conclusion and Future Works

In this paper we have presented the Gestational Diabetes Mellitus Monitoring system. In particular, we have highlighted the components of this PHS and how these are interconnected. We have also explain in details how these interconnections are secured and how these security means can protect the GDMM system from the most widely spread and serious security threats.

The GDMM system is currently prepared to test it in the field. Another aspect we are currently work on, is to generalise the architectural concepts of the GDMM system into a more general pervasive healthcare monitoring framework. Other aspects that we plan to implement is to allow for personalised monitoring rules.

Acknowledgment

We would like to thank Dr. Juan Ruiz and his team for sharing their insights in GDM, and their support of our research. This work has been partially funded by the Hasler Stiftung and by the Nano-Tera grant 10020.

References

[1] J. Epping-Jordan, "Innovative care for chronic condition," World Health Organization, Tech. Rep., 2001.

[2] J. Meier, A. Homer, D. Hill, J. Taylor, P. Bansode, L. Wall, R. B. Jr, and A. Bogawat, "App Pattern: Four-Tier Web Application Scenario," 2009, https://apparch.codeplex.com/wikipage?title=AppPattern-Four-TierWebApplicationScenario(TableModule, accessed 02.09.2011.

[3] J. Krampf, S. Bromuri, M. Schumacher, and J. Ruiz, "An agent based pervasive healthcare system: a first scalability study," in *Proceedings of the 4th ICST International Conference on eHealth (eHealth 2011)*. Springer, 2011, (to appear).

[4] S. Bromuri, M. Schumacher, K. Stathis, and J. Ruiz, "Monitoring gestational diabetes mellitus with cognitive agents and agent environments," in *Proceedings of the 2011th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2011)*, Aug. 2011.

[5] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, 2000, chair-Richard N. Taylor.

[6] OWASP, "The Open Web Application Security Project: Category:Principle," 2011, https://www.owasp.org/index.php/Category:Principle, accessed 15.06.2011.

[7] ——, "The Open Web Application Security Project: Top 10 2010-Main," 2010, https://www.owasp.org/index.php/Top_10_2010-Main, accessed 14.09.2011.

[8] A. K. Maji, A. Mukhoty, A. K. Majumdar, J. Mukhopadhyay, S. Sural, S. Paul, and B. Majumdar, "Security analysis and implementation of web-based telemedicine services with a four-tier architecture," in *Proc. 1. Int. ICST Workshop on Connectivity, Mobility and Patients' Comfort*. IEEE, 2008.

[9] C. Orwat, A. Graefe, and T. Faulwasser, "Towards pervasive computing in health care a literature review," *BMC Medical Informatics and Decision Making*, vol. 8, no. 26, 2008.

[10] D. Isern, D. Sanchez, and A. Moreno, "Agents applied in health care: A review," *international journal of medical informatics*, vol. 79, pp. 145 – 166, 2010.

[11] A. Farmer, O. Gibson, P. Hayton, K. Bryden, C. Dudley, A. Neil, and L. Tarassenko, "A real-time, mobile phone-based telemedicine system to support young adults with type 1 diabetes," *Informatics in Primary Care*, vol. 13, pp. 171 – 177, 2005.

[12] V. Kumar and S. Lie, "DiaBetNet project page," 2010, http://slie.dyndns.org/projects/DiaBetNet/webpage/, accessed 07.09.2011.

[13] C. H. Salvador, M. P. Carrasco, M. A. G. d. Mingo, A. M. Carrero, J. M. Montes, L. S. Martin, M. A. Cavero, I. F. Lozano, and J. L. Monteagudo, "2005," *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, vol. 9, no. 1, pp. 73 – 85, 2005.

[14] G. Ghinea, S. Asgari, A. Moradi, and T. Serif, "A jini-based solution for electronic prescriptions," *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, vol. 10, no. 4, pp. 794 – 802, 2006.

[15] P. d. Toledo, S. Jimnez, F. d. Pozo, J. Roca, A. Alonso, and C. Hernandez, "Telemedicine experience for chronic care in copd," *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, vol. 10, no. 3, pp. 567 – 573, 2006.

 ISBN: 978-1-61208-179-3