Rehabilitation at Home of Patients with Neglect Using a Telemedical Intervention: a Security Perspective

Emil Palm
Abstract

The rehabilitation of stroke patients suffering from neglect is today a time and resource heavy process for therapists and patients. New technology based methods are currently being developed to make the process easier and make the rehabilitation more in control by the patient. One method is the RehAtt™ currently developed by Brain Stimulation AB, the aim of this thesis is to investigate the possibility to use this method at home with a telemedicine solution.

The patient-specific data produced by the software makes the foundation for the assessment done by doctors or therapists in the rehabilitation process. This needs to be accessible for the assessor and therefore must be transferred securely between the two parties. The security of the system will be the main focus of this thesis.

The major contributions presented in the thesis includes a list of requirements for a telemedicine solution for the existing software together with a system design and a demo implementation. Another important contribution is an introduction to the digital platforms currently building up the Swedish national healthcare system, which is based on a literature study and interviews. A deeper evaluation study was conducted of currently used systems and tools within healthcare systems to examine if existing software could be used in a solution to allow integration against national healthcare platforms, focused on Swedish healthcare platform. An introduction is also provided to the security topics of digital healthcare, based on recent research in the security field. The results of this literature study also informed the design of the telemedical system in focus for this thesis.
Acknowledgements

First I would like to thank Brain Stimulation AB for letting me do this interesting master thesis and the always so appreciated kind words and valuable feedback on the work. Secondly I would like to thank my supervisor Helena Lindgren for many valuable meetings with constructive feedback and good suggestions for the thesis work.

Finally I would like to thank family, friends and my girlfriend for the support throughout the thesis work.

Icons used in figures throughout the report are designed by Freepik and distributed by Flat-icon.
List of Figures

1. Illustration of the Diffie-Hellman key exchange 10
2. Illustration of symmetric encryption 11
3. Illustration of asymmetric encryption 12
4. Computational costs of the PAAS scheme 18
5. The general structure of an archetype 28
6. Example archetype of a guitar 31
7. Example code of connecting to the NTjp with the HIP SDK in the Java language for retrieval of information about the fictive person “Tolvan” 36
8. System description of the Brain Stimulation telemedicine system 45
9. Brief description of the modules needed for the patient’s side in the telemedicine system 47
10. Brief description of the modules used in the server system 48
11. Database scheme for the server system 52
12. Local database’s schema 53
13. Internal calls of server upon incoming requests 56
14. The analyse window in RehAtt analysing data for a test user 59
**List of Tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Search phrases and databases used in the study</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Requirements for the doctor client of the system</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>Requirements for the patient client of the system</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td>Requirements of the server system</td>
<td>42</td>
</tr>
<tr>
<td>5</td>
<td>General security requirements for the system</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Requirements of the administrator part of the system</td>
<td>43</td>
</tr>
</tbody>
</table>
1 Introduction

Usage of Internet based healthcare tools have increased in the last couple of years. Specialist healthcare is now offered to people living far away from larger hospitals through telemedicine systems. Through these systems specialists can assist the doctors at a local care center or therapists can perform rehabilitation on distance through video communication. Often the video communication is to recreate the physical room through a virtual environment. There types of systems saves both time and money for patients as well as for the hospitals. [12]

One group of patients who could be using a telemedicine solution in their home for their rehabilitation is patients who suffers from neglect[1] which is caused by a stroke. Neglect is a condition where the attention and awareness of one side of space is decreased compared to healthy people. Patients with this condition doesn't process and perceive stimuli from one of the space, and most of the patients are also partially paralysed.

Rehabilitation of neglect patients has traditionally needed specialists and therapist to assist and perform the rehabilitation in the physical room. Products are currently developed to take one step into the future and into an virtual environment. This cuts down the need of specialists in the rehabilitation process, and frees up more of their time to be able to treat more patients. Brain Stimulation AB[2] currently develops a platform for both fast diagnostics of neglect after a stroke, and rehabilitation software tools for neglect patients.

The aim of this thesis will be to investigate the possibility to make the rehabilitation platform from Brain Stimulation into a telemedicine application allowing patients to rehabilitate in their home. The thesis will focus on the security of personal data of this solution. To create a system design a literature study where conducted, which included published peer-reviewed articles an scientific conference material in the computer science area published within recent years. From the literature study a set of tools where chosen to be evaluated if they could be used in the telemedicine solution. The thesis project ended up with a system design and a demo implementation together with a introduction for the reader of the digital infrastructure of the Swedish national healthcare platforms, how it is structured and how it could be used by external applications.

1.1 Background

Brain Stimulation AB is currently developing a platform consisting of two sub platforms for stroke patients with neglect. The first platform is designed for fast diagnostics used in emergency health care. This program cuts down the time for diagnosing neglect compared to traditional tests. This is set to be used as a substitute to the traditional tests in the future.

The second platform is designed for rehabilitation of neglect patients, called RehAtt™. This program is used in the rehabilitation process and is shown to have positive effect for patients using the program in their daily training.

A natural step in the development of the product would be to implement functionality for telemedicine. By this making it possible for sending a complete set of the training equipment to a patient’s home allowing them to rehabilitate in their own home.

At the same time a doctor or therapist can, from the hospital or rehabilitation center, connect to the patient’s home and check how the training process is progressing. This saves time for doctors/therapists as well as patients and more patients can be handled by a single doctor/therapist.

1.2 Telemedicine

Telemedicine is defined to be the usage of electronic communication to exchange medical information from one site to another to improve the patient security, education and specialist treatment. As of today telemedicine solutions use a variety of different technologies such as smartphone applications, wireless tools, video communication and other communication solutions.

Telemedicine is used to spread the availability of specialist treatment to patients in regions without larger hospitals or specialised rehabilitation centres. Specialists from different regions or countries can participate in the same rounds as specialists at the local hospital to effectively increase the knowledge base of the team. This increases patient security and decreases the likelihood of incorrect treatment of patients.

Solutions have been tested and researched in Sweden, Norway, Australia and Canada to decrease the travelling for both doctors and patients. At the hospital in Longyearbyen at Svalbard, Norway there have been studies with virtual emergency healthcare teams with colleagues at the university hospital of Tromsø.

All telemedicine solutions must have strict security policies and be implemented with security based architecture when handling personal data about patients. A system must also be designed to strictly follow the laws in countries where the system is used. In Sweden a system must follow the “Patientdatalagen (2008:355)”. The law states how personal data and how patient records should be handled, created and used in the healthcare.

1.2.1 Example of a Current Telemedicine Project in Västerbotten, Sweden

Patients who have suffered from a stroke with dysphasia need support and treatment from a speech therapist. In Umeå, Sweden, the rehabilitation is a series of 16 meetings where the patient performs speech training together with the therapist. Some patients however chooses to abstain the treatment due to large distances between their home and the rehabilitation center. In this case a telemedicine solution which can be borrowed from the rehabilitation

http://www.americantelemed.org/about-telemedicine/what-is-telemedicine

http://www.riksdagen.se/sv/Dokument-Lagar/Lagar/Svenskforfattningsamling/

Patientdatalag-2008355_sfs-2008-355/
center and the rehabilitation series can be done with a specialist from the patient’s home on distance.

The dysphasia example above saves a lot of time and money for both patients and the hospitals. In Sweden if all patients in the counties without hospitals with rehabilitation centres would offer this solution with treatment in their home it would decrease the need of travel from $100000km$ to $20000km$. The time for patients would also decrease with 3000 hours which directly saves 1.3M SEK/year. [12]

1.3 How RehAtt is Used Today

The Brain Stimulation system is today used in a proof of concept study in hospitals and rehabilitation centres in Europe and the USA. When used, the computers running the software are not connected to the Internet. All running is performed offline, due to regulations in the hospitals network policies not allowing computers from external sources to be connected to their network.

Since the offline “requirement” the results are saved to disk. The system uses a profile system where a patient is handed a profile in the system. The profile is used when diagnostics or training is performed. The results are saved to the profile. Due to the fact that the system is running offline there is no possibility to synchronize the patient profiles between the diagnostic- and the training platform.

The diagnostic program is designed and developed to be used bedside as well as mounted in a “station” or at a stationary desk or similar. The diagnostic program uses a touch screen laptop with five tests, which two of them require a mounted position of the laptop and a fixed position for the head in relation to the screen. The two tests measure reaction times of button presses from a stimuli shown on the screen. The three other tests only use the touch screen to drag, press and picking objects in the different tests. The tests are designed to be a digitalized version of standardized tests in neglect diagnostics [9].

RehAtt uses a desktop computer together with stereoscopic 3D glasses and a haptic device in the training. It uses applications similar to computer games but designed for neglect rehabilitation, one is similar to the classical Tetris game. The training forces the patient to be more alert, and includes spatial hand training with the haptic device.

Similar to the speech therapy example, the Brain Stimulation training system would be as applicable for neglect patients doing their rehabilitation in their own home. As similar to the example a lot of time and money could probably be saved by this solution. A neglect patient would need a total of 15 hours rehabilitation time under a period of three weeks with RehAtt. The time for the patient to only go to the hospital 15 times during the three week period is a great time saver for the patient and would benefit the time used for a therapist as well. [10]

1.4 Thesis Outline

This report starts with an introduction and the background of the thesis project. Information about the product to use in the project is presented and general information about telemedicine system.

Chapter 2 states the problem description with what problem to solve within the thesis work which in this case is to create a foundation for a telemedicine system for the existing RehAtt software. In Chapter 3 the methods used to solve the specified problem are presented.

An in-depth study in the security field is presented in Chapter 4 containing general information to the reader about methods and technologies used in the material studied and in the system design. Chapter 5 continues the in-depth study with an evaluation of selected software tools used in healthcare systems today.

Requirements for a telemedicine system for RehAtt are categorized and listed in Chapter 6, both legal and functional requirements are included.

Chapter 7 specifies the constructed system design for the telemedicine solution for RehAtt. Following, Chapter 8 documents the demo implementation of the system design.

The thesis report finishes with Chapter 9 with the discussion chapter of the report. Chapter 10 specifies the conclusions and what contributions has been made within the thesis work. At last Chapter 11 discuss, and specifies future work from the thesis work.
2 Problem Description

The company Brain Stimulation AB – located in Umeå, Sweden, wants to expand their product for rehabilitation of neglect patients via a telemedicine solution where their product is placed in the home of neglect patients. Telemedicine solutions have proven to be time savers for both patients and doctors, they also reduce travel distance for necessary rehabilitation sessions [12].

2.1 Problem Statement

Today the company has no plan or design of how a telemedicine system for the product could be built and used in the healthcare. One part of the project will then be to create a system design for a telemedicine system for the existing product. The design must also be suitable for a start-up company, such as Brain Stimulation, to implement. The design should not need two years and 100 developers to be implemented properly, 2 developers on 3-6 months of work is a more reasonably amount of work for the start-up company. This estimate could be proven wrong during the work of the thesis project. It is possible that the market is not ready for a telemedicine system for this product. This will be investigated in the project.

The security of patient data in the system will be key and have high requirements from the healthcare to be accepted and used in the daily work. An investigation will be conducted of tools and methods of security in eHealth systems designed to be strong and resilient against attacks in the future years.

2.2 Deliverables

When finishing this thesis project the result of it will be a complete system design of a telemedicine solution for the existing RehAtt™ program.

The design should include database schemas, transfer protocols, encryption methods, key exchange schemas, system design diagram, proposed deployment methods (e.g. cloud resources, self hosted infrastructure, etc.), how to authenticate and authorize, a recommendation of frameworks and languages to use in the implementation. All together will be the result of the thesis work, and the necessary foundation for starting the implementation of the telemedicine solution.

Parts of the design will be implemented in a demo implementation and possibly tested at Björkgården, Sävar Sweden or NUS, Umeå Sweden. This depends on the result of the demo implementation and if approval for the test is given.
3 Methods

The main part of the thesis project consists of a literature study of published material from both peer-reviewed journals and scientific conferences. The main topic is security in eHealth systems. The literature study will mostly include articles and conference material from 2014 to current date. This literature makes the foundation for the in-depth security study and selected software and tools which are further evaluated to be used in the system design.

To complement the literature study, interviews were conducted with professionals from the IT-section at Västerbottens County Council (Västerbottens Läns Landsting, VLL) to get an insight in what systems, frameworks and tools that would be preferred to use in a telemedicine system that would be used within the daily work of VLL and other regional and national healthcare institutions.

With the knowledge gained from the published material, interviews combined with all knowledge gained during the education from relevant courses a design for a telemedicine system is created. The design should be carefully created to later be the ground for a future fully implemented telemedicine solution for the RehAtt rehabilitation platform.

3.1 Literature Study

To find relevant material for this study of current security techniques for eHealth systems searches in scientific databases of articles and conference material have been used. Mostly PubMed and IEEE databases have been used. In Table 1 some of the used search words are presented.

Worth to mention is that all published material referenced in the report will be listed in the

<table>
<thead>
<tr>
<th>Search phrase</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>health security software medical system encryption</td>
<td>PubMed</td>
</tr>
<tr>
<td>health security software</td>
<td>PubMed</td>
</tr>
<tr>
<td>ehealth security encryption</td>
<td>PubMed</td>
</tr>
<tr>
<td>”a review”[Title] AND security AND ehealth</td>
<td>PubMed</td>
</tr>
<tr>
<td>”a review” security ehealth</td>
<td>IEEE</td>
</tr>
<tr>
<td>ehealth and security</td>
<td>IEEE</td>
</tr>
<tr>
<td>Attribute-based authentication eHealth</td>
<td>IEEE</td>
</tr>
<tr>
<td>password storage scheme</td>
<td>IEEE</td>
</tr>
<tr>
<td>system authentication</td>
<td>IEEE</td>
</tr>
<tr>
<td>security AND ehealth</td>
<td>Scopus</td>
</tr>
</tbody>
</table>
3.2 Selection of Systems and Tools to Evaluate

An evaluation of selected existing systems and tools was done on the basis if the system or tool is suitable to use in a telemedicine solution for the existing platform. The main criteria in the evaluation is if the system or tools is developed to make telemedicine applications for the Swedish healthcare system.

Other criteria for the evaluation of the systems or tools is if they are currently used in telemedicine solutions within Sweden or EU today and whether or not the system or tool makes the design, implementation or usage of the finished product easier and/or more convenient.

3.3 Design Process

When creating the design for a telemedicine system for the RehAtt platform, the knowledge from literature study and the interviews combined with requirements for the system was carefully used to design a secure and robust system that could be used in today’s healthcare in Sweden. The design is produced in collaboration with a medical expert at Brain Stimulation to create a suitable solution for doctors and other medical professionals included in the treatment and rehabilitation of neglect patients.

3.4 Identification of Requirements

The requirements where identified in collaboration with professionals at VLL through interviews and together with professionals at Brain Stimulation. The requirements are classified into sub-classifications of the different clients, server system, security and legal requirements.
4 Security in Telemedicine Systems

In eHealth systems the security is one of the most important aspects. If the security fails and a malicious user gains access to the system the health of patients can be in danger if the intruder can change health records, lab results, patient history, and so on. The security system must support basic services such as preserving privacy for patients [11]. Further the system must also handle confidentiality, authentication and authorization of users accessing the system. The system also needs high availability which can be achieved by deploying the server resources to the cloud.

In the following sections the following themes that were identified in the literature study will be presented:

- Encryption, Section 4.1
- Authentication and Authorization, Section 4.2
- Access Control, Section 4.3
- Examples of authentication schemes, Sections 4.4 & 4.5
- Cloud deployment 4.6

Two examples of authentication schemes were examined in-depth. The first implementation is an Attribute-based authentication protocol for anonymous authentication of patients which is an example of authentication and authorization, see Section 4.4. The second scheme is a mutual agreement of session keys and authentication using biometric information which is an example of an authentication scheme using key exchange, see 4.5.

4.1 Encryption

Security in eHealth systems can be obtained in some different ways. The most common way is the encryption approach where the content during transmission is encrypted with cryptographically strong methods.

To initiate the encrypted communication a exchange of keys is often performed to make a shared secret that is used in encrypting and decrypting messages sent between the parties. When using a shared secret the method is called symmetric encryption and is often used together with another cryptographic technique called asymmetric encryption where the parties does not have a shared secret. Instead they uses a key pair, one public and one private key used in encryption and decryption of messages.

Within eHealth systems only authorized persons should be able to retrieve medical information. Therefore the system must handle user authentication and authorization to only allow
the user with the sufficient access level to the material within the system. The common way is that the user gives some sort of identification to an authentication service in the system and then gets granted access, and then uses the access to retrieve, edit or creates Electronic Health Records (EHR) or documents in the system.

**Key Exchange Scheme** To initiate an encrypted conversation of message exchange some sort of key exchange is most often performed in symmetric cryptography. There are several ways of performing the key exchange securely over an insecure channel such as the Internet. It is one of the most known and used ones and is illustrated in Figure 1.

Another version of the key exchange can be done with public and private keys. One example is the Hypertext Transfer Protocol Secure (HTTPS) traffic which is initiated by a key exchange between a server and the web browser. The browser connects to a server and requests the server to identify itself. In the identification response the server sends a copy of its Secure Socket Layer (SSL) certificate. The browser checks the certificate and then encrypts a session key with the public key from the certificate and sends it to the server. The server uses the private key do decrypt the session key and then encrypts an acknowledgement with the session key and sends it back to the browser. Thereafter the connection is established and information can be sent securely between the two parties.

---


Figure 1: Illustration of the Diffie-Hellman key exchange[^1]
Encryption Technologies Encryption in computer systems often comes in two forms, symmetric encryption and asymmetric encryption. The common between the two of them is that one user can encrypt a message and send it to another to make sure no one other than the receiver should be able to read the message.

The idea behind symmetric encryption is to have a shared secret between the two parties who want to communicate. This secret can be stored from before or agreed upon at the start of the communication, when the later approach is the preferred one. Some examples of symmetric encryption standards is Advanced Encryption Standard (AES), Blowfish, Triple Data Encryption Standard (3DES) and Twofish. An illustration of symmetric encryption can be seen in Figure 2.

Asymmetric encryption utilizes two keys for each user, instead of one shared. The first key is called “Public key”, this is used for encryption of messages to the corresponding user. The second key, called “Private key” is used for decrypt received messages from others. The only key that can decrypt a message encrypted with a public key is the corresponding private key, see Figure 3. This means that the public key can freely be distributed to other users for them to use in communication to yourself.

AES is one of the most used techniques of symmetric encryption, it’s used in the encryption of HTTPS traffic. So every in the every day life the technique is used when people accesses their online bank, social media and their email accounts. However, before the HTTPS communication is set up a session key needs to be agreed upon between the two parties, this is done by asymmetric encryption.

Usually the Rivest-Shamir-Adleman (RSA) cryptosystem is used in the HTTPS key exchange, also Elliptic Curve Cryptography (ECC) can be used. In the key exchange the requester collects the server’s public key and creates a session key to use and encrypts it with the public key received and send it to the server. The server decrypts the key and then the two parties both shares the same secret to use in the encrypted communication between them.

Storage of Passwords A method for storing passwords which are also scalable to meet future requirements of storage of passwords is called bcrypt. It’s designed to store passwords securely and meet the computational power of future machines by scalable computational
Figure 3: Illustration of asymmetric encryption

Other techniques used to store passwords is to use one-way hash functions together with randomly selected salt values. These are mostly not scalable like bcrypt and the hashing methods used to store passwords needs to be changed by time to not jeopardize the user credentials to end up in the possession of attackers.

However there is some special hardware solutions designed to attack bcrypt stored passwords to increase the guessing speed compared to general computers [23]. A password scheme will probably never be 100% secure but the grade of secure can almost every time be increased and it will be the implementers/administrators task to keep their users’ credentials safe.

Transmission of Clinical Data All transmissions of patient data must be encrypted. The laws and directives states that patient data transferred must not be in plain text form. So any encryption standard is viable to use.

However the healthcare provider is the responsible for the system they use. Therefore the encryption methods used must be strong and practically hard for attackers to break. Else the system would not be used, therefore the security and encryption standards to be used should be the strongest available.

One idea to protect the patient data is to “wash away” the personal data in the communication, hence the communication will not include any personal data such as names, addresses, telephone numbers, etc. The data sent would then be anonymous.

4.2 Authentication and Authorization

The users can use username/passwords, tokens, biometric information, certificates etc to authenticate to a system. Within the authentication process the system is checking the users authorization level as well and then gives back the corresponding access level to use in the system. Two examples of authentication schemes are given in Section 4.4 & 4.5
Biometric authentication has a lot of interesting and appealing properties, such as the user has not to remember passwords to get access to system resources. This is specially appealing for neglect patients who have reduced memory and could be partially paralysed in one side of the body making it hard for the patient to login to a system by inputting a user identity and password. [7]

Other nice properties for biometric authentication are the increased security and it is harder for attackers to steal a user’s identity. The biometric authentication could be fingerprint- and/or iris scanning, speech recognition, facial features or even DNA [15]. All of these methods requires additional hardware to be added to the system which makes the system more complex and the system is forced to be dependent on another companies products with supplied software.

Another solution to the biometrics authentication is to utilize smartphones which are constantly growing in numbers in the society. The camera of a smartphone or tablet can be used to scan facial features to authenticate users. [20] [21]

Another technique utilizing signed certificates could be used within the system where each user gets a personal certificate used to identify the user. This technique is currently used for the international student WiFi-network Eduroam [16]. But the same technique could be used in a medical system.

This signed certificate needs to be stored for each user. This creates a problem that the user will be bound to only use the computer where the certificate is installed. A solution to this could be that each user gets a special USB-flash drive to store the certificate so it can be used on multiple machines. This solution however then requires the user to never lose the USB-flash drive and the distribution of the drives then is forced upon officials at the hospital where the system is used.

The assurance of privacy and security in eHealth systems for patients’ data is based on three main components: a) Authentication of users, b) Authorization of users and c) Handling patient’s consent to access their health data. [3]

In the authentication process the identity of a subject is verified to check whether the subject is what or who it claims to be. The authentication links the identity to a subject. The process have a set of parameters [22].

**Assertion** A user is guaranteed access at a particular time using an accepted authentication method. Or simpler, User “U” was granted assertion at time “T” by “A” provided condition “C” is valid.

**User Credentials** The required input to the authentication. In most cases this is user password, one time password, digital certificate, biometric information, etc. In eHealth applications entity authentication has to be implemented, where one party proves its identity to another party.

Next process is the authorization process where the identified identity is checked against an “Access Control Decision” (ACD). This decision is based on access policies in the system, to grant correct access to the corresponding identity, this is always performed when a user requests access to restricted resources such as documents, personal data, Personal Health Records (PHR), etc.
The authorization process can easily be described as a function $A$ takes three parameters $S$ – the subject, $R$ – the resource and $O$ – the operation to perform on $R$. The function returns a boolean value $True$ access is granted to $S$ on resource $R$ to perform operation $O$, or False. $A(S,R,O) \rightarrow (False, True)$.

In eHealth systems the two most common models for access control policies are “Role-based Access Control” (RBAC) and “Attribute-based Access control” (ABAC) [22]. Below follows a short introduction to RBAC and ABAC, more information about access control is presented in Section 4.3

**Role-based Access Control** [19] In this model the users are assigned into roles. Then the roles is mapped against permissions to access different resources. This model has proven to be very stable and be reducing the possibility of incorrect assignments of user permissions.

**Attribute-based Access Control** [18] Allows more flexibility and fine-grained user permissions than RBAC, at the same time as not specifying relationships between subjects and objects. The ABAC grants access by evaluating the rules against the attributes of entities (subject and objects).

The last process is control structure of patient’s consent for eHealth Data. More and more patients would like to have better control over their own health information, such as who is allowed to access the information, when is the information accessible, and for what purposes are the information accessible. To make an eHealth system handling these functionalities two services should be supported, "Delegated Consent" and "Shared Decision-Making" [22].

**Delegated Consent** In this service when the consent of the patient is handed over to a trusted entity, for example a trusted doctor or similar.

**Shared Decision-Making** This service is built around the principle “no decision about me without me”. The patient is actively contribute to their healthcare. The patient is also able to share the information with third parties which are involved in their care.

Systems using consent modules need to support a number of functionalities to allow proper consent management. A system with a consent module should first support the functionality to handle consents from users, i.e. how they give their consent or dissent for accessing their personal data. The second functionality needed is to determine when and how consent is obtained to different parties. The third functionality is the patient’s right to choose and to change their mind regarding the consent or dissent given. The patients should also be able to see who that have access to their information. Lastly the module needs to handle exceptions, if the patient is for example unconscious or in coma doctors should be able to override the consent and access necessary data [22].

**4.2.1 Owner of Data**

In the interests of the company, access to all data generated by users would be ideal. This data could be used in future research in the field. But this access can be tricky to implement since the nature of patient data must be secure and not handled by unauthorized users.
Usually the healthcare providers own the data created by software used in the daily treatment of patients. Another model lets the patients to be the owner over their own data. Patient’s then administer who, how and when other gets access to their data. This can of course be overridden in case of emergency of the patient’s life.

In neither of these cases data is necessary accessible for the developing company of the product/software. To solve this it is important to make the meta data of the rehabilitation from RehAtt available for analyse for future development and research project. The training data could be stored anonymously in the server system and it would still contain information about the neglect type, sex and age, but attributes such as names, social security number, addresses, etc. would be removed from the training data.

### 4.3 Access Control

Managing access control to resources in an eHealth system is one of the most important aspects of the system. Only users with the correct authorization level are the ones that should have access to a resource. This prevents leakage of personal data to users that are not supposed to get access to the data.

The most used strategy when building a access control system is to use the RBAC scheme. RBAC is as stated above, very stable and a well tested scheme but it lacks the capability of defining fine-grained policies. Such as access to a patient’s records will be accessible within a time frame, where ABAC has it’s strengths[1].

#### 4.3.1 Role-Based Access Control

The idea of RBAC is to assign the users into roles, and then give the roles different authorization levels. The method is easy to use and administer. Because of this it has a number of ready-to-use frameworks (e.g. Spring Security[7] and Apache Shiro[8] both for Java). This makes the implementation into a system smooth, together with the framework additional functionalities is bundled, such as authentication and authorization of users, protection against common attacks, etc.

Most of the previously designed EHR systems utilize a RBAC schema for the access control defining permissions and restrictions. Most of the schemas utilizes predefined roles for the users. Some schemas however allows the patients to do refinements of the roles to allow customization of access and restriction to make access to their data customizable. [8]

#### 4.3.2 Attribute-Based Access Control

System using the ABAC use, most often, the “eXtensible Access Control Markup Language” (XACML) which is the standard language for ABAC policy systems. The language is a declarative policy language implemented in eXtensible Markup Language (XML), the language has a processing model evaluating access requests to the rules in the corresponding policies attached to the resource requested access to.

The XACML language has a reference architecture which handles all access requests and policies of resources. The modules used in the architecture are described as follows:

**Policy Administration Point (PAP)** Holds all policies in the system. When a request is processed by the system the policies needed for the request is retrieved from PAP and evaluated in PDP.

**Policy Decision Point (PDP)** Matches requests against the rules stated in the policies retrieved from the PAP. PDP returns the response of the request with an access decision that could either be **permit**, **deny**, **not applicable** or **indeterminate**.

**Policy Enforcement Point (PEP)** When a decision is determined in the PDP it is PEP’s task to enforce the decision as well as enforcing the policy rules for the resource.

**Context Handler (CH)** Acts as the interface between the PEP and PDP. When incoming access requests is received in PEP they are converted into the XACML language and sent to PDP for a decision.

**Obligations Service (OS)** This module handles the obligations in policies. A policy can have the obligation if a user permits access to it, the owner will be informed about the access. Or if an unauthorized user requests the resource the owner also will be informed.

### 4.4 Example of an Attribute-Based Authentication Scheme

An interesting idea for preserving user information and privacy while authenticating to eHealth systems is to use an attribute-based authentication. Gou and coworkers [13] describes a technique, PAAS: Privacy preserving attribute-based authentication, where an eHealth system authenticates the patients and physicians with attributes instead of real identities. The main feature of the technique is that patients are identified by showing a set of attributes instead of traditionally using an identity combined with a secret. In this way the patients real identities will always be anonymous.

The PAAS framework is designed to preserve the privacy of personal information of the users (patients and physicians) in an eHealth network. The framework uses attribute-based authentication where the users use attributes to identify and verify each others attributes regarding diseases and professions, without compromising the personal identity. This is the main design goal of PAAS.

The framework uses a Trusted Authority (TA) for distribution of public and private keys to all users. The framework also has a semi-trusted registration center (RC) which generates and issues the credentials to users based on their individual attributes, the physicians is checked of their professionals and with the information issues the corresponding credentials. The TA can be refereed to as the health administration and should be trusted to 100%. The RC can be refereed to as a hospital or clinic with qualifications certified by the TA.

Patients in the system can use pre-assigned pseudonyms or random ones in the communication in order to add additional security to preserve the integrity of the real identity by being anonymous, but still can prove and authenticate the attributes to obtain medical ser-
services based on the diseases. The users can also communicate with each other within the framework.

The PAAS framework consists of four privacy levels. The levels are numbered from 0 to 3, and the highest level has the highest level of privacy and discloses least information. The lower levels “leaks” more information details but may be more efficient, however all levels provides anonymity and untraceability for all users within the system.

Privacy Level 0, PAAS0

Within the privacy level the physicians are required to show the validity of their professional qualifications on third party platforms (such as social networks). The anonymous patients needs the confirmation from the doctors that what they say is true, this without revealing the doctors credentials or identities. If the credentials or identities is compromised they can be used by attackers to launch a impersonating attack.

In short, the PAAS0 privacy level requires that every user in the system can verify the validity of the attribute credentials without compromising the identities of the users.

Privacy Level 1, PAAS1

In this privacy level not just verification of attributes is made, also a check of the attributes values is performed. This is done in the authentication phase in the higher levels as well. Users does not care if information about meaningless things are revealed as long as the information is non-traceable to real identities.

Privacy Level 2, PAAS2

The privacy level 2 considers patient interaction with each other. In communication with each other patients may want to share some information about their own diseases with other patients that have the same symptoms. When they have learnt each others identical attributes they can communicate to share information with each other. When the sharing of identical attributes is done the two communicating patients knows the intersection set of their attributes. The intersection set is respectively a subset of both individual sets of attributes.

Privacy Level 3, PAAS3

In the highest privacy level offers the highest security and privacy requirements. The level ensures that the patients’ attribute values used in the authentication are not revealed to anyone else. The improvement of security from PAAS2 consists of only knowing the size of the intersecting set, i.e how many attributes that is identical without saying which ones. Else this level would have the same security features as PAAS2.
**Performance**

The performance of the scheme was measured by the authors on a MacBook Pro with an Intel Core 2 Duo with a clock speed of 2.8GHz and 4GB of RAM memory. The total computational cost of PAAS0 is around 0.1s and for PAAS3 close to 1.2 seconds with close to linear relationship between the levels. The computational costs can be seen in Figure 4 [13, Fig. 2].

A quick comparison to a laptop processor used today, Intel Core i7-5600U the computational speed is 2.2 times faster\(^9\) and would lower the computational costs significantly. New processors would probably halving the computational times for all the privacy levels.

The authors state the scheme has acceptable performance on the old MacBook Pro referenced above, so with new hardware the scheme would more so have good performance on laptops.

![Figure 4: Computational costs of the PAAS scheme](https://www.cpubenchmark.net/compare.php?cmp%5B%5D=2456&cmp%5B%5D=1012)

**Security Analysis**

The authors of the article have discussed and the privacy and integrity of the privacy levels and general attacks against all levels [13]. The two most common attacks against this scheme is the tracing attack and the collusion attack.

**Tracing attack**  The attacker traces a user in the system and collect information about a user to finally be able to link a user to a real identity. This scheme does protect against this attack using random numbers in commitments and use of proofs.

To be able to perform this attack the attacker needs a valid user within the system to be able to verify its proofs against the server. If the attacker gets a valid user the attack

\(^9\)https://www.cpubenchmark.net/compare.php?cmp%5B%5D=2456&cmp%5B%5D=1012
cannot be performed due to the switching of pseudonyms and random numbers in commitments. The attacker cannot create a link between an attribute set and identity.

Collusion attack In collusion attacks the attacker tries to create new copies of data from a set of obtained data. The idea is to create new valid data sets to use in the system to retrieve all stored data.

In this scheme the collusion attack can happen between users or between users and RC. The simplest form of this attack is easily prevented. If a malicious group of users wants to get vital information about another user they have to send a large amount of queries to the user. But since the user will use different pseudonyms and different random numbers in the commitments the group of attacking users cannot get the relationship between the pseudo-identities and the commitments.

The attacker can also together with RC launch a collusion attack where the attacker is able to obtain all attributes of a user, the attacker cannot retrieve the plain-text of a commitment since it is only the user who knows the random number corresponding to the particular commitment.

The scheme is resilient to statistical attacks performed together with RC. The attack cannot obtain credentials or certificates due to it is impossible for RC to get the relationship between multiple commitments.

Attacks on PAAS0 The authors classifies the impersonation attack as the most possible attack against PAAS0. The TA can be used in this case to trace the credentials and real identity of the user with leaked user credentials.

Another attack is the “unique identification” attack. This attack is only possible against users that have a unique attribute. For example if there is only one physician at the hospital with a particular attribute and if someone knows this fact. The physician is easy to identify from the rest of the physicians. This is however protected against since all users (physicians and patients) should have same number or attributes in their attribute set.

Attacks on PAAS1 This privacy level have a asymmetric structure and this causes the possibility of two attacks, “self-proving” attack and certificate leakage.

The self-proving attack is protected against by telling the RC to issue an additional certificate to the server containing the secret user attribute together with a secret random number not known to the attacker. By doing this the attacker cannot prove one of the essential equations without the additional certificate.

A group of users can cooperate to obtain the secret random number so they can generate their own certificate to make the verification process. However the scheme is constructed around the Discrete Logarithm Problem (DLP) and is hard to figure out the secret values from the certificate. If miss behaviour is detected the user can be traced by the TA and the credentials can be withdrawn.

The second attack, certificate leakage is impossible due to the assumption of complexity in the DLP problem.

Attacks on PAAS2 Two major types of attacks are considered by the authors, first when sending the random numbers, the second attack is on the outcome of comparison result.

http://www.igi-global.com/dictionary/collusion-attack/4512
The first attack is handled by the scheme by the difficulty in deriving the critical parts of the message sent containing the random numbers between two users. This because the assumption of DLIN problem is hard to force. This guarantees the attribute privacy when two users sends random numbers back and forth.

The other attack uses the outcome of the comparison. PAAS2 specifies that if no attribute values matches nothing is to be learned of the other user. The verification returns 1 if and only if the attribute values does match. If not the result returned is an arbitrary string instead of the matching attributes. The users can also return incorrect random numbers and proofs which makes attacks harder to perform since the inconsistency in the comparisons.

**Attacks on PAAS3**  The extra protection in PAAS3 is that the scheme allows users to respond different queries within the same time slot. This makes it even harder to for statistical attacks on the user’s attribute set.

Another attack is to use the inconsistency of the encryption and multiplication of the received packet. There are two protections against this in the scheme. The first one is applying another set of proofs for checking verifiability and validity. Together with this the homomorphic encryption will have guaranteed consistency. The second protection is to encrypt the random numbers used in the generation of the cipher text and sends these together to check and validate consistency.

### 4.5 Example of a Key Exchange Scheme

Reliable and secure communication is key when using a system that collect, transfers and store data from patients in a healthcare system. The transmission of data needs to be encrypted when sent between a remote user and a system server. There are a few ways to initiate and create an encrypted connection between two connectors over an insecure channel (public Internet).

Das and co-workers [4] proposed a scheme based of another scheme with security improvements for reliable and secure exchange of session keys. The scheme utilizes smart cards and biometric identification. However the scheme have weaknesses against a number of common attacks presented by Amin and co-workers [2], together with the co-workers he proposes a new schema with better security.

However, recently Das and co-workers [5] has shown in a new article that in fact the scheme developed by Amin has design flaws and is not considered as secure as previously stated. The nature of the scheme is however very interesting regarding the possibility to mutually authenticate both parts and make a mutual agreement of a session key used to encrypt transmissions between the two parties during a session. Therefore the Amin scheme [2] is studied in-depth and follows:

The point of the scheme is that the scheme is anonymity preserving and uses mutual authentication in session key agreement. This means that the user validates the server as real and authenticated and the server validates the user as real and authenticated and then agrees upon a session key to use, to encrypt the traffic sent between the two parties.

Amin’s scheme utilizes smart cards and biometric information together for authentication to the system. The smart cards stores information making it possible for the user to authen-
ticate the server and the server to authenticate the user. Together they agree upon a session key used to encrypt messages sent between each other during the session.

The scheme has four phases in which the security and anonymity is tested and proven. The phases are:

**Registration phase** in the phase the smart cards is created either in person or over a secure channel. The card stores information about the user and the server to be able to authenticate the remote server.

**Login phase** When a registered user wants to connect to the system this is initiated through an insecure channel. The user inserts its smart card in the smart card reader. The smart card reader then validates the user using biometric authentication together with identity and password. The smart card reader then sends a first message to the server of initiating a secure communication channel.

**Authentication and key agreement phase** When the server retrieved the message from the card reader it verifies the sender against the registered users in its database. The server then answers the card reader with a reply message continuing the secure channel creation. The smart card reader then validates the information and the agreement is finished and both parties now have the same secret key for communication.

**Password change phase** This phase is for updating users passwords. The user identifies itself to the smart card reader with its smart card and biometric information, identity and old password. The user then is prompted to input two identical new passwords. If the passwords is identical the smart card reader updates the values of the password on the physical card.

When communicating over the insecure channel the scheme uses cryptographic one-way hash function, bio-hashing function, concatenation operation and XOR operation on the information. The concatenation operation is only used together with the one-way hash function to add information after each other to the same hash.

The session key consist of random nonce chosen by user in login phase and by server in authentication phase. The session key is then decided to be the hash value of the two random nonce concatenated together.

This makes up a secure way of mutual agreement on a session key while preserving the user anonymity. The scheme is proven to be secure and performing mutual authentication using BAN logic. Together with this both a formal and informal security analysis is presented. The formal analysis is made with the Automated Validation of Internet Security Protocols and Applications (AVISPA) simulator and the program declares the scheme as SAFE which means that the scheme is secure against both active and passive attacks, including man-in-the-middle and replay attacks.

The informal security analysis discusses and proves that the scheme is secure against a number of attacks and preserves users anonymity. The attacks listed and discussed in the article is the following ones:

1. [https://en.wikipedia.org/wiki/Burrows%E2%80%93Abadi%E2%80%93Needham_logic](https://en.wikipedia.org/wiki/Burrows%E2%80%93Abadi%E2%80%93Needham_logic)
2. [http://www.avispa-project.org/](http://www.avispa-project.org/)
Offline password guessing attack  The attacker guesses the password and tries if it is the correct one. Due to the fact that the probability of guessing the correct password is extremely low and cannot be done in polynomial time the attack is negligible.

Insider attack  An insider user, system manager or administrator, that gets the correct password of a user can later use the credentials to login to other services, where the user use the same credentials. This is however protected against since during the registration phase the insider only gets to know a hash of the identity and the password. And due to the cryptographic one-way function the password cannot be retrieved. And guessing is again not possible in polynomial time.

Smart card theft attack  To successfully theft a smart card the attacker must calculate a valid value of its biometric information together with the servers secret, the attacker cannot get possession of the server secret. This together with that in order to successfully launch this attack the attacker needs to know the real identity of the user but that is not possible since the anonymity is preserved.

Impersonation attack  In this attack the attacker tries to impersonates the user by trapping the login message sent to the server. To generate a new login message with it’s own secret the attacker once again needs access to the server secret and valid unique identity of the user. The attacker can also try to impersonate the server, but as before it needs access to the server secret to generate valid response messages. Therefore the scheme is secure against impersonating attacks.

Session key discloser attack  The session key is decided to be the cryptographic one-way hash of two random nonce and the attacker have no way of disclosure these two individually. Regarding guessing of the two random nonce the guess of the two high entropy parameters cannot be done in polynomial time.

A similar scheme to Amin’s scheme is the one created by Li and co-workers [15] and utilizes a similar key exchange with built in authentication of the user as well as the server. By utilizing biometric information together with cryptographically strong hash functions and fuzzy extractors. The server authenticates the user as a valid user at the same time the user authenticates the server as valid. Then both parties will be sure that they communicate with a reliable party. The scheme is designed for smart-phones and the biometric information is collected with the camera of the smart-phone.

4.6 Cloud Deployment

Today more and more systems are deployed in a cloud environment. The cloud offers many appealing properties such as resource pooling, rapid elasticity and on-demand self-service [6]. eHealth services deployed in cloud environments have special requirements in Sweden to secure personal data. These are presented on The Swedish Data Protection Authority’s (Datainspektionen) web page [13].

Within the cloud research must focus has been put to improving the security in the services offered. Including secure encrypted storage of data, enabling searching on the encrypted data and to be able to update encrypted data in storage [17].

For high availability, redundancy and energy efficiency a cloud deployed platform would be ideal for eHealth applications. This would probably be done in a private cloud, where the infrastructure is maintained by the organisation running the platform and offers the healthcare services. The private cloud could be combined with public cloud resources to handle less critical data which could be the anonymous training data.

The public cloud hosted by Amazon, DigitalOcean, Microsoft, Rackspace, etc. could be used in a solution where the patient records are stored encrypted and could only be decrypted at the hospital treating the patient. The idea is that no one, not even Brain Stimulation, could decrypt the records but the doctor users at the hospital treating the patient.

4.7 Conclusion and Implications for Design Requirements

When building medical systems today the system security is one of the highest priorities. Access control to resources is required and all accesses of resources should be logged so every access can be traced to who accessed, when was it accessed, and why it was accessed.

The proposed access methods included in the security study is the RBAC and ABAC. RBAC is a bit simpler to handle and implement than the ABAC, but ABAC has greater customization where the users can specify who can access their data and how/why. According to Fernández-Alemán and co-workers [8] most of the EHR systems utilizes RBAC.

Signed certificates for authentication could be a good idea. It makes the authentication process more secure when the user can validate the server’s authenticity and the server can do the same for the user. However a drawback of this system is that the client certificate must be stored somewhere, and if this is done locally on the machine this means that the user can only login from that machine only. This could be sufficient for a telemedicine system used only in their home, but causes problem if the same user is to use the telemedicine system from another location, i.e the hospital.

The Attribute-based authentication scheme presented by Gou [13] offers appealing properties such as anonymity of personal data. However it seems as a complex and abstract solution where users only authenticates with attributes.

Regarding the Amin’s [2] scheme which is based around the Das [4] scheme is an interesting approach to make a secure key exchange with a mutual authentication, which are two important properties for eHealth systems. This scheme requires smart cards, or similar, to use in the authentication. When using the smart cards additional hardware will be needed, the smart cards also needs to be initialized and distributed to the patients by the hospital which needs more work for the system to be used by patients at home.

Li and co-workers [15] have presented a similar scheme utilizing biometric scans from a smartphone. This could be viable for a telemedicine solution since the usage of smartphones have increased steadily the last couple of years. This scheme could be interesting for a secure mutual authentication solution for a telemedicine system.

Both Amin’s and Li’s schemes would be appealing to use in a eHealth system for a secure and reliable authentication. Amin’s scheme, or another based off it, could be used for stationary rehabilitation solution where the patients uses their personal smart card for authentication, and Li’s scheme could be utilized in a stationary approach but also in new methods which are more mobile without a stationary smart card reader since the biometric
information is collected by a smart-phone.
5 Systems and Tools Evaluation

In the making of a telemedicine system the system must handle patient records. In this case to save diagnostics results and rehabilitation results. To make this easier an integration and/or connection to a existing patient records system could be useful. Therefore, an evaluation is performed on systems and tools to see if it is a good idea and possible to make an integration against a national centralized healthcare system.

The selected systems/tools to evaluate has been chosen based on the interviews and knowledge from the literature study. A tool has been selected if it is currently used in the digital national healthcare systems or if it makes integration to them more convenient. The following three tools were selected:

**openEHR** openEHR is an open platform that handles Electronic Health Records (EHR). It consists of predefined profiles which is built to be configurable for the end customer with the principle of archetypes. Archetypes is a standardized way of storing clinical data and specified in the international standard CEN/ISO 13606. openEHR is used in 8 regions is Sweden as of today within the CAMBIO Healthcare system. It is also currently used within the National Health Service (NHS) Leeds North Clinical Commissioning Group in UK. Also Australia uses the system in a number of regions. Because the system of data representation is used in Sweden as well as other countries the system is interesting to evaluate if an integration to a Brain Stimulation telemedicine solution would be possible, feasible and useful.

A in depth study of openEHR follows in Section 5.1.

**FHIR** Fast Healthcare Interoperability Resources (FHIR) pronounced “fire” is the predicted successor of openEHR. By using newer data formats, light weight solution and is REST based makes implementations against mobile devices easier.

The FHIR standard is developed by Health Level Seven (HL7) which is an international organization that develops standards in the healthcare area. The greater study of FHIR follows in Section 5.2.

**HIP SDK** Health Innovation Platform (HIP) Software Development Kit (SDK) is developed to make healthcare applications easier to register and use patient records’ data. This

---

platform is developed in Sweden by INERA AB and is currently used to build and run the system “Mina vårdkontakter”. The system is used by residents to access their patient record data, book appointments with doctors at their locally chosen clinic or renew prescriptions.

This platform is only used within the Swedish healthcare system. But the integration possibilities seems to be simple and the work to connect an application to the “Nationella Tjänsteplatformen” (NTjP), which holds all patient records, is simplified in the SDK.

This makes the system interesting to investigate further on how a integration and connection to the national platform could be used in the Brain Stimulation telemedicine solution to access the Swedish market. See Section 5.3 for more information about HIP.

Systems and tools not included in the evaluation follows here with a short description and explanation why:

**Open eHealth Integration Platform** The platform is used to build open source healthcare applications for the Open eHealth provided services. The platform is built upon Apache Camel and extending the routing and mediation engine. The platform supports most of the used eHealth standards today, but is too big and complex to be a manageable alternative for the RehAtt system today. In a future re-evaluation of systems and tools this needs to be evaluated again if the needs is fulfilled with this system at that time.

In the following sections openEHR, FHIR and HIP SDK will be further described and evaluated.

### 5.1 openEHR

openEHR is a open framework which focuses on the EHR and offers computability and interoperability in healthcare systems and avoids product and vendor lock-in. Worth to note openEHR is not a full scale system that is simply installed in a company’s server hall and ready to be used. openEHR have made a set of tools to ease storing, retrieving and searching in electronic health records. For example there is no openEHR database that is ready to be deployed. The using company has to set up a database themselves, which can be a non-trivial task. Information and general advices on how to choose an appropriate Database Management System (DBMS) for the clinical data can easily be found from other developers using openEHR around the world in form of slide shows, workshops and articles.

The openEHR Foundation has defined a set of specifications in which a reference model together with a language for “clinical models”, called archetypes, separates the models.
from the software. The specifications also contains a query language to enable search and
indexing of the archetypes. With the separation of the archetypes from the software the
archetypes can be created so that both software developers and clinical experts understand
how they are structured. This simplifies the development since the software developer can
concentrate on the application and the clinical expert can focus on what data the archetype
should contain.

5.1.1 Archetypes

In 2007 the European standard ISO13606 was created. The standard states the Archetype,
how it is created and shared containing clinical information to provide quality healthcare.
Each archetype specifies the key attributes of information in patient records for the main
information they store. Information such as who, when and where are not needed to be
stored in each archetype.

The archetypes uses external terminologies such as Systematized Nomenclature of Medicine
– Clinical Terms (SNOMED-CT), which is a systematically organised collection of medical
terms which can be computer processable, together with some other collections. These
terminologies builds the foundation for the archetype, the archetypes are then written in the
specified Archetype Definition Language (ADL). The argument for using archetypes is
that these only need to be created one time and can then be used within the system at any
point and reused between system implementations. There will only be created one archetype
of blood pressure. This archetype contains all information about the blood pressure for a
patient at a given time. Multiple blood pressure archetypes can then be used to make con-
clusions about a patient’s health with blood pressure data from a number of measurements
over time.

The archetypes can be used in trees. The patient record acts as the root node of a tree, then
the archetypes branches out and all together they keeps track of all measurements, notes and
values of a patient's health. One archetype could be “Routine check” which then contains
archetypes of blood pressure, hydration level, EKG etc, which creates the tree of archetypes.

Structure

In Figure 5 the general structure of an archetype is shown. The archetype is built in ADL
which really just is a “glew” syntax for three other syntaxes. Constraint for of ADL (cADL),
Object Data Instance notation (ODIN) and First Order Predicate Logic (FOPL).

The archetype is then built specifying a set of tags. First is the tag specifying which type
of ADL and release used in the archetype. Then follows information about language, short
description and definitions. Under the language tag all languages are listed to which the
archetype is translated. Then follows the description tag where all short descriptions is
listed, all translated and defined to the language tags.

These two follows the ODIN Object Data Instance Notation which is the object syntax
Figure 5: The general structure of an archetype

which is a completely generic syntax that is used to define the most in the archetype.

The following is an example of ODIN collected from the language tag in the Blood Pressure archetype originally created by Sam Heard in 2006:
The definition follows the formal constraint definition cADL. With this fields and parameters is specified in it’s own syntax stating types, intervals, units etc. An example from the Blood Pressure archetype again, this time the definition of Pulse pressure:

```
... ELEMENT[at1007] occurrences matches {0..1} matches { -- Pulse pressure
  value matches { C_DV_QUANTITY <
    property = <[openehr::125]>
    list = <
      ["1"] = <
        units = "mm[Hg]"
        magnitude = <|0.0..<1000.0|>
        precision = <|0|>
      >
    >
  >
} ...
```

Worth to note is that C_DV_QUANTITY is deprecated and have been removed from ADL 2.
ELEMENT[at1007] is a tag referencing to a translation, each language has a list of translations so each archetype can be exported to any of the translated language. In this case the tag matches to the English translation tag (which uses ODIN):

```plaintext
...  
["at1007"] = <
  text = "Pulse pressure"
  description = "The difference between the systolic and diastolic pressure."
>  
...
```

The example contains the text name that it matches to and a short description of the tag. Then the element specifies that the Pulse Pressure should occur 0 or at most 1 time in the archetype, this with a quantity has a range from 0.0 to less than 1000.0 with the unit mm[Hg].

The third syntax type is FOPL which states the rules for the archetype in first order predicate logic. This to express constraints of the data within the archetype that is an instance of an underlying information model.

**Complete example - Guitar**

In Figure 6 an example of a complete archetype describing a guitar is presented. First the ADL version is specified, followed by the definition and terminology.

The original language is specified as the language code `en` which is English. Then follows the definition, in the definition the terminology is used. The definition defines an instrument that has `id1` that refers to a name: guitar and a short description: string instrument. The instrument have then the properties:

- **Size**  the size is specified to be within the interval 60 - 120 cm.
- **Date of manufacturing**  where the year and month of manufacture must be specified
- **Parts**  lists the parts of the guitar, which is `id2` - the neck, and `id3` - the body. The neck is made of the material `ac1` which can either be timber or nickel alloy. and the body is made of `at3` which is timber.

The terminology is then presented. As said and done the terminology is used in the definition. The terminologis is identified with unique tags, as `id1`, `at3` and so on. The tags then specifies a text, used as name, and a short description in this case. At last a value set is specified. This containing the tag `ac1` which is a set of tags, in this case `at3` and `at4`.

---

17 [http://www.openehr.org/releases/AM/latest/docs/ADL2/ADL2.html#_an_example](http://www.openehr.org/releases/AM/latest/docs/ADL2/ADL2.html#_an_example)
5.1.2 Current Usage

openEHRs archetypes solution is used in Sweden in the Cambio COSMIC health care system, currently used in 8 regions where Upssala and Örebro is two of the larger ones. The system is also used within NHS in the United Kingdom, Australia, Denmark and Brazil.

In the following the advantages and disadvantages are listed.

http://www.cambiohealthcare.co.uk/
Advantages

- Archetypes defines how to store clinical data into structures.
- The ADL language together with the AQL makes searches for clinical data easier for research and health care.
- Avoids vendor lock-in where the user chooses how to implement the system and how it is used.
- Freedom in usage and implementation makes the archetype approach highly customizable and adaptable for the end users.

Disadvantages

- The freedom creates more work for the developers creating all customizations to fit the end users (nurses, doctors, therapists).
- With archetypes the storage, handling, retrieval, creation etc. more sensitive to security breaches since the developers developing the current system needs to implement the security features needed for a patient records system instead of a central implementation.
- The archetypes can be confusing for end users when used in “raw” format.
- Relatively old technique, newer standards exists

5.2 FHIR

FHIR is a tool for structuring clinical data into predefined resources, similar to openEHR’s archetypes. Also as openEHR, FHIR is not a ready-to-deploy system. The FHIR standard is developed by Health Level Seven (HL7) which is an international organization that develops standards in the healthcare area.

FHIR is recommenced by the IT-section at VLL to use in new healthcare applications. A solution today could be to use a standalone server system with no integrations against the digital infrastructure of the Swedish national systems where doctors use either a desktop program or web portal to use the system to monitor patients and monitor the rehabilitation process. The communication between the client application should use the FHIR protocol to be as easy to integrate when the national system architecture becomes available to external healthcare companies such as Brain Stimulation.

5.2.1 Resource

All resources in a system has a known identity, this identity can be addressed with an url. A resource contains structured data items which are all describes in the definition of the type. The resource can be defined in either eXtesible Markup Language (XML) or Javascript

[https://www.hl7.org/fhir/]
Object Notation (JSON) depending of the used type in the system, if the system uses both data structuring methods both definitions are used.

Structure

Below follows an example of a name attribute from the general patient resource example.

```json
...
"name": [
  {
    "fhir_comments": [
      "Peter James Chalmers, but called \"Jim\"
    ],
    "use": "official",
    "family": ["Chalmers"
    ],
    "given": ["Peter", "James"
    ]
  },
  {
    "use": "usual",
    "given": ["Jim"
    ]
  }
]
...
```

The name attribute first specifies an array of comments for the name attribute. This example contains one comment which is the full name of the patient and a reference to what he is called. Thereafter follows official statements of the patient’s name, as “Peter James Chalmers”. And a usual name used in every day’s life, “Jim”.

Interactions

Standard interactions for resources are:

- **read** Reads current state of a resource
- **vread** Reads current state of a specific version
- **update** Updates a existing resource

21 [https://www.hl7.org/fhir/patient-example.json.html](https://www.hl7.org/fhir/patient-example.json.html)
22 [https://www.hl7.org/fhir/http.html](https://www.hl7.org/fhir/http.html)
delete  Deletes a resource

history  Retrieves the update history of a resource

Standard interactions for the whole system:

conformance  Collect a conformance statement from the server

history  Retrieve update history for all resources

search  Preforms a search for all resources, can include filters

5.2.2 Security

The security in FHIR needs to be implemented and different design choices can be done. The customization level of the system is high, this makes the final product fit the end customer’s requirements better.

The usage of the HTTPS protocol is optional in the system, but all exchange of real healthcare data should use some sort of encryption, i.e. Secure Socket Layer (SSL) or Transport Layer Security (TLS). Most of the interactions requires authentication, and all of these interactions are subjects to either RBAC or ABAC and some times both. All of these techniques is presented in chapter 4.

5.2.3 Implementation

FHIR utilizes a Representational State Transfer (REST) framework for easy access via the Hypertext Transfer Protocol (HTTP) protocol of resources. Each resource has the same set of operations called “interactions” making the managing of resources simpler. The implementer of the FHIR system can choose which interactions that should be available and what resources to support. However the server shall provide information of which resources and interactions are supported, this is called a “conformance statement”. FHIR supports different versions of the same resource, i.e. FHIR has built in versioning system making it easy to gather the changes of a resource over time.

When using FHIR through a RESTful Application Programming Interface (API) together with the XML and/or JSON it is easier to integrate more hardware solutions to the system, such as smart phones, tablets or other devices. Desktop applications, web applications and web portals can easily use this data formats. It’s possible for easier implementation of different client applications for the patients to use to access their own records in the way they choose.

Existing tools

For implementation of a healthcare system utilizing FHIR there are some tools available to use. FHIR clients are available for download and further implement or integrate into external applications. FHIR clients for Java\(^2\), C#\(^3\) and JavaScript\(^4\) are all available through:

\(^2\)[https://github.com/cnanjo/FhirJavaReferenceClient](https://github.com/cnanjo/FhirJavaReferenceClient)

\(^3\)[https://github.com/ewoutkramer/fhir-net-api](https://github.com/ewoutkramer/fhir-net-api)

GitHub.

Other tools available are an official FHIR validator which is used to validate resources in a system. This validator is implemented in Java. There are also an icon pack containing icon at different resolution to make FHIR applications look the same. At GitHub there is also a fully implemented FHIR server presented, this server is written in Pascal.\(^2\)

5.3 HIP

Health Innovation Platform (HIP) is developed for developers who want to create applications and services for the Swedish healthcare system. HIP supports making of healthcare services, for example assisted decision systems, and residents services, for example information systems which can be a system presenting all scheduled appointments, journal entries and coming appointments for a resident. With HIP it is also possible to create applications that uses open data supplied by the Swedish healthcare system.\(^3\)

The HIP has a SDK, called HIP SDK which helps the developer in the development of an application for the Swedish healthcare. When retrieved the sufficient access to the healthcare APIs the SDK eases the connecting and retrieving data from the central healthcare system containing the patient records.

HIP SDK uses OAuth 2.0 and SITHS certificates to connect securely to the healthcare APIs. SITHS is a identification service that contains both physical and electronic identification for service IDs.\(^4\) The SITHS certificate must be applied for to the Swedish healthcare system, then issue a valid certificate that is then installed, followed by a small configuration in the HIP SDK and then is ready to use. When the certificate is configured by the HIP SDK, it handles the rest, all connections where the certificate is needed.\(^5\)

5.3.1 How to use HIP SDK

The HIP SDK is easy to use for programs developed in the Java language together with Maven to download and handle the dependencies and needed libraries from the HIP SDK. In Figure an example code for collecting a fictive person’s care contacts.

First there is a check whether or not the person have gave their consent that the application have the rights to retrieve medical information about themself. The application then gets an answer from the NTjP telling if they are eligible and authorized to retrieve information about that person. If so, the example then makes a new connection to NTjP to collect all care contacts of that person. When the request is complete and a response is retrieved the application can use the information collected from NTjP.

5.3.2 Current Usage

HIP is used in the Swedish healthcare today. The platform “Mina vårdkontakter” is built upon HIP. A resident can log in to the platform with an electronic identification (e-ID), and

\(^{2}\)https://github.com/grahamegrieve/fhirserver

\(^{3}\)http://www.hip.se

\(^{4}\)http://www.inera.se/siths

Figure 7: Example code of connecting to the NTjP with the HIP SDK in the Java language for retrieval of information about the fictive person “Tolvan”.

then see information from his/her patient record, schedule a time to see a doctor, and many other important and useful services. The HIP platform connect to NTjP that performs information exchange easily and effective between connected systems in a secure way. Access is granted through a service contract, the contract regulates the information exchange with other systems and services.

In the following the advantages and disadvantages are listed.

Advantages

- Used to build the national platform “Mina vårdkontakter”
- Makes all connections to the national healthcare servers securely without need of configuration and built in authentication.
- Personal data can be “washed away” in the gateway to minimize malicious user to extract personal data
- Authentication with SITHS-certificates, used and considered secure for today’s eHealth systems in Sweden.

Disadvantages

- The HIP SDK is only used in Sweden for the Swedish healthcare

Not currently available for commercial use in external companies

Makes a dependency to Java language

5.4 Swedish Healthcare Systems

This section contains a shorter explanation of the digital infrastructure of the Swedish national healthcare system and its regional sub systems.

The Swedish healthcare systems is built on a foundation of regional systems where all regions have their own sub systems where patient records are stored. The base information about patients are stored for each region independently from other regions. These regional systems there are around 20 currently active systems in use. However, some regions uses the same sub systems but have their own deployments. This means that two regions could use the same system architecture but runs independent from each other.

The national Swedish healthcare system currently developed and maintained by Inera AB continuously gathers information about patients from the regional sub systems. The national system utilizes “tjänstekontrakt” which is similar to services where these services gather information about patients from all regional sub systems into one national service. Applications can use these services to retrieve information such as medical history, make time bookings at local health clinics, etc. This functionality is built into the platform “Mina vårdkontakter” at http://www.1177.se.

The regions have some way to integrate healthcare applications directly against the systems but this would not be feasible to do since many different systems are currently used by the regions in Sweden. This would lead to different integration implementations for the regional systems.

Today there is a module called, “Hjälpmedelstjänsten” attached to the national systems where residents themselves can supply the system with own data about themselves. Residents can by blood pressure monitor or a pulse meter and upload the data to the existing module. All monitors and meters must follow an ISO classification before it is allowed to store information to the database.

This module sound very promising for integration against healthcare applications to store information for Swedish patients into the national system. This is however not the case according to professionals at VLL, where the data uploaded cannot be trusted in the healthcare and therefore is useless. This module is not recommended to use when developing new telemedicine solutions for a product like RehAtt.

Currently the new platform “Stöd och behandlingsplattformen” (SoB) is in development and when finalised external companies could use that platform for their products to use integrations against the Swedish national healthcare system. The platform will be able to store data directly into the regional systems which have not been easy before. This allows the telemedicine system to store training data directly into the regional systems which can later be propagated into the national systems.

According to the IT-section at VLL FHIR will be the used framework of data structuring

30 http://www.inera.se/
31 http://www.inera.se/TJANSTER--PROJEKT/hjalmmedelstjansten/
5.5 Conclusions and Implications for the Design of RehAtt

The HIP SDK is the application framework to use to make integrations to the infrastructure for the Swedish healthcare platform 1177. This framework would be the way to go to make integrations in the future when it becomes available for external healthcare companies like Brain Stimulation. It’s downside is that it is only available for integration to Swedish healthcare system. This means that integration against other countries infrastructure would require different implementations.

This would probably be the case anyway, since the literature study gave no findings of global standards of data formats, integration methods, authentication methods, etc. of national healthcare systems.

openEHR and FHIR are two data structuring formats with supplementary tools for data searchability and other utilities. Both are similar where data is stored into predefined profiles containing information, clinical measurements, etc.

openEHR is used today in different systems used in Sweden, UK and Australia. That makes the framework attractive. On the other hand in Västerbotten and VLL the framework is not used and the more modern and light weight REST-based FHIR framework is the recommended one to use when developing applications to be used in future.

To make a telemedicine solution most suitable for the future for now, it seems to be the best solution to use the HIP SDK to make integrations against the Swedish national healthcare digital infrastructure together with the FHIR framework for data structuring. However HIP SDK is not yet released for companies to use.

The conclusion to be drawn from this study a solution built today would utilize the FHIR protocol from data structuring within the system and later integrate the system against the national healthcare systems with the HIP SDK or with direct data transfer with the FHIR protocol into the national systems. This would most certainly be done through SoB.

With the new SoB platform it would be possible to store patient-specific data directly into the regional and national systems with supplied software tools. If this could be the case the stored data would be protected by the same principles as the ordinary EHRs and other clinical data. The data would also be accessible for doctors from the EHR system they use in their every day work which would be positive for the learning of how to use the new RehAtt system by using familiar tools.
6 Requirements of the Design

The requirements for a telemedicine system for the current RehAtt platform will be presented in the following sections, starting with the legal requirements from Swedish laws and European directives, followed by the system requirements categorized and organized into subsections.

6.1 Legal Requirements

Within eHealth systems, security and privacy are among the most important aspects since the systems handles vital and personal data about patients. With the level of importance of keeping the data secure there are laws and directives stating important principals requiring e-health actors handling, storing and transmitting personal data. 1995 the European Council enacted the Data Protection Directive, 95/46/EC.

Legal restrictions for systems in Sweden depends on to whom is the system developed for and by who is it used. There are two greater candidates when building systems that is to be used in the Swedish health care system. Health care systems, which targets doctors, nurses and patients and handles patient data and provides health care services, the Swedish law “Patientdatalagen” (2008:355) PDL, is the applicable law.

The second candidate of legal requirements is for residents systems, the “Personuppgiftslagen” (1998:204) PUL, is the applicable law.

In Sweden, when planning to use cloud resources and store, handle and transmit personal data certain rules must be followed. In this case the patient data counts as personal data and the telemedicine solution must follow the guidelines from The Swedish Data Protection Authority when using cloud resources while processing personal data.

These guidelines states that the one using cloud resources is the controller of the personal data even though the processing is done by the cloud service provider. It is the controller who has the responsibility to make sure that all applicable laws such as PUL and other applicable laws for the current field of use, in this case the PDL.

Recently the European Parliament approved new rules regarding personal data stored and used by IT-system. The rules includes the rights of a resident to be able to:

• A right to be forgotten
• "Clear and affirmative consent" to the processing of private data by the person concerned
• A right to transfer personal data to another service provider
• The right to know when your data has been hacked
• Ensuring that privacy policies are explained in clear and understandable language
• Stronger enforcement and fines up to 4% of firms’ total worldwide annual turnover, as a deterrent to breaking the rules

These rules entered into force after the publication in the EU Official Journal, at May 4th 2016. All member states of the European Union must within two years transpose the rules into national law which means by May 6th 2018 the directive must be an applicable law in member states.

Regarding the data created while training there is a possibility to class the data as working material instead of entries into the patent medical journal. In Sweden the The National Board of Health and Welfare (Socialstyrelsen) makes the judgement that temporary memory entries that is used for information to be added to the medical journal is classed as working material.

The working material has lower requirements when it comes to storage and content. Personal data could be removed from the working material so it becomes untraceable to a patient and the working material can be destroyed after it has been added to the medical journal. However, if the temporary memory entries are stored and organised for a longer time, The National Board of Health and Welfare expects that the entries would be treated as medical journal entries.

6.2 System Requirements

A scenario for a patient starting the rehabilitation platform for his/her daily rehabilitation at home is provided, which has served as the foundation of the requirements identification. The resulting requirements are summarised in Section 6.2.2.

6.2.1 Scenario of a Patient Using the Software

Patient starts the computer and the RehAtt training program. The computer connects to the Brain Stimulation online server system for storing training results. The patient trains from one hour in the different games and the software uploads the results as the games is finished and the data is stored in the server system.

At a different time a doctor wants to check in on the patient and analyse the training done this day. The doctor connect to the web platform though the installed browser on the system.
When logged in the doctor can see an overview of all patients assigned to him/her. The doctor chooses to browse the training data from the patient. From there the doctor can later do journal notes to put into the hospitals medical records of the patient.

If the doctor finds that the training needs to be changed he/she can insert an entry for the user that during training next week they will have a meeting through the built-in video conference solution to communicate to each other face-to-face and there give feedback on the training if something needs to be changed.

The doctor should also be able to directly connect to a patient who is running their training for communication and to observe the ongoing training session, and get a “live feed” of data.

If the server system is down or non-reachable for the patient during a training session the results will be stored locally on the training computer and later uploaded to the server system when it comes available again. The doctors can however only get data from the server system unless they are directly connected to the training computer used by the patient in a training session.

### 6.2.2 Identified System Requirements

All the system requirements of the RehAtt™ telemedicine solution will be presented in the following subsections organised into tables for overview based on the category for the clients, server system, security and the admin panel. The requirements will be organised in the tables using the RFC2119 standard for classification.

The technical requirements on the telemedicine solution from the **doctor client**, e.g. functionalities for doctors to interact with training results, analyse data to make adjustments in the rehabilitation process, etc. is specified in Table 2.

#### Table 2 Requirements for the doctor client of the system

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>To access the system the doctor can log in securely to the system</td>
<td>MUST</td>
</tr>
<tr>
<td>Doctors can only access files from patient they treat</td>
<td>MUST</td>
</tr>
<tr>
<td>A doctor can connect to a patient to monitor the training process</td>
<td>MUST</td>
</tr>
<tr>
<td>Doctors can access the training result history for a patient</td>
<td>MUST</td>
</tr>
<tr>
<td>All transmission of data between the doctor and the system must be secure</td>
<td>MUST</td>
</tr>
<tr>
<td>Personal data can only be decrypted by doctors who cares a patient</td>
<td>MUST</td>
</tr>
<tr>
<td>All data stored locally by doctors must be done with security in mind</td>
<td>MUST</td>
</tr>
<tr>
<td>A doctor can connect and monitor the training in real time (live)</td>
<td>SHOULD</td>
</tr>
<tr>
<td>A doctor can communicate with an online patient face-to-face</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Access to the system is not limited to one computer for one user</td>
<td>SHOULD</td>
</tr>
</tbody>
</table>

The requirements of the **patient client** of the system is specified in Table 3.

The requirements of the **server solution** is listed in Table 4 with all identified requirements with security and availability.

The general requirements regarding security within the system is specified in Table 5.
### Table 3 Requirements for the patient client of the system

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>The patient can communicate to a connected doctor</td>
<td>MUST</td>
</tr>
<tr>
<td>The communication must support voice communication</td>
<td>MUST</td>
</tr>
<tr>
<td>Training could be done offline</td>
<td>MUST</td>
</tr>
<tr>
<td>Training results can be uploaded when connected to Internet</td>
<td>MUST</td>
</tr>
<tr>
<td>The program must be easy to understand for patients</td>
<td>MUST</td>
</tr>
<tr>
<td>The program will not stop if the Internet connection is lost</td>
<td>MUST</td>
</tr>
<tr>
<td>The software handles multiple users for a single computer</td>
<td>MUST</td>
</tr>
<tr>
<td>The software isolates the patients’ data from each other</td>
<td>MUST</td>
</tr>
<tr>
<td>Communication with the doctor should be a video feed (face-to-face)</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Access to the system is not limited to one computer for one user</td>
<td>SHOULD</td>
</tr>
<tr>
<td>The patient must not be able to see it’s own results</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Doctors can let patient to see their progression</td>
<td>MAY</td>
</tr>
</tbody>
</table>

### Table 4 Requirements of the server system

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>No data could be accessed by non-authorized users</td>
<td>MUST</td>
</tr>
<tr>
<td>Data from the database can only be retrieved by the server software</td>
<td>MUST</td>
</tr>
<tr>
<td>All transmission of data to and from the server must be encrypted</td>
<td>MUST</td>
</tr>
<tr>
<td>The server system scale well for the number of users</td>
<td>MUST</td>
</tr>
<tr>
<td>Data stored contains no personal data in clear text</td>
<td>MUST</td>
</tr>
<tr>
<td>The server system can always be accessed in one or more locations</td>
<td>SHOULD</td>
</tr>
<tr>
<td>The servers should be redundant in case of failure</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Training data gathered in the system can be used in future research</td>
<td>SHOULD</td>
</tr>
<tr>
<td>The usage of the system can be monitored by Brain Stimulation</td>
<td>MAY</td>
</tr>
</tbody>
</table>

### Table 5 General security requirements for the system

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both patients and doctors must authenticate to access the system</td>
<td>MUST</td>
</tr>
<tr>
<td>The authentication must be secure in practice and transmission</td>
<td>MUST</td>
</tr>
<tr>
<td>Data storage within the system must be secure</td>
<td>MUST</td>
</tr>
<tr>
<td>The storage must protect against leakage of data to unauthorized users</td>
<td>MUST</td>
</tr>
<tr>
<td>Transmission of data between parties must be secure and encrypted</td>
<td>MUST</td>
</tr>
<tr>
<td>Data stored in system have a creator (patient, training data)</td>
<td>MUST</td>
</tr>
<tr>
<td>The data must be searchable dependent on the patient</td>
<td>MUST</td>
</tr>
<tr>
<td>The system must follow current legislation in Sweden and EU</td>
<td>MUST</td>
</tr>
<tr>
<td>Doctors and patients should use roles to access resources</td>
<td>SHOULD</td>
</tr>
<tr>
<td>The system can trace which user requested a specified resource</td>
<td>SHOULD</td>
</tr>
<tr>
<td>The system must follow current legislation in USA</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Usage of integrations against the digital national healthcare systems</td>
<td>MAY</td>
</tr>
</tbody>
</table>

The requirements for the **administration functionalities** in the system is specified in Table 6.
Table 6 Requirements of the administrator part of the system

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is one system administrator owned by Brain Stimulation</td>
<td>MUST</td>
</tr>
<tr>
<td>The system admin can create hospital admin accounts</td>
<td>MUST</td>
</tr>
<tr>
<td>The hospital admin can create doctor and patient accounts</td>
<td>MUST</td>
</tr>
<tr>
<td>Hospital admin can assign doctors to patients</td>
<td>MUST</td>
</tr>
<tr>
<td>Hospital admin can remove created accounts (Doctor &amp; patient)</td>
<td>MUST</td>
</tr>
<tr>
<td>System admin can remove created accounts (Hospital admin)</td>
<td>MUST</td>
</tr>
<tr>
<td>Admin can login to admin panel</td>
<td>MUST</td>
</tr>
<tr>
<td>Admin can monitor the usage of the system</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Brain Stimulation can monitor the usage of all hospitals</td>
<td>MAY</td>
</tr>
</tbody>
</table>
7 System Design

The design of a telemedicine system, or any system to be used within healthcare situations, always needs to prioritize the security of personal data about patients. With that in mind a system design for a telemedicine solution for the RehAtt has been created. The design chapter is constructed in a top-down approach starting with a general overview of the system in Section 7.1 and breaking it down to smaller pieces, in Sections 7.2–7.6 where security, transfer of data, database design, user roles and the system’s flow are presented.

7.1 System Description

The brain Stimulation telemedicine system for the rehabilitation will mainly consist of a server system that could be deployed in cloud or hosted by the company themselves on appropriate hardware. The server system consists of a RESTful API web service for communication together with a database storage of user information, training data and roles for all users. A brief illustration of the system design is presented in Figure 8.

The rehabilitation program will store data locally and upload the data to the server system when the service is accessible. This will mostly be done right after a game is finished. If the server is unavailable the data will be stored locally and uploaded at a later time. Doctors can retrieve data from the servers with the data from training sessions to analyse if the training is going as planned or if it needs to be adjusted.

A lot of design choices for authentication and authorization. The easiest way is of course the username/password authentication with authorization tokens used for a session together with the HTTPS protocol for transmitting data between client and server.

Figure 8: System description of the Brain Stimulation telemedicine system
There is a concern of how useful a username/password authentication could be since the patients who are going to be using this system suffers from neglect. Most of the patient will probably have a spatial paralysis in one body half, so they will not be able to fully use their right arm and their finger functionality will be decreased making it hard to enter regular passwords on traditional keyboards. These patients also have reduced awareness of stimuli from one side of the space.

However this may not be enough of security for a eHealth system. The system should atleast use a two factor authentication (2FA). Where the username and password is paired together with another authentication method like smart-cards, certificates, biometric information.

The web service and the heart of the server system will be implemented in Java together with the Spring framework to be the foundation for the RESTful web service. The choice of Java i strengthen by most of the integration tools against other eHealth systems uses the Java language, HIP SDK is one example.

The DBMS software chosen to use is PostgreSQL which is the largest and the most advanced open source DBMS. It is also chosen because most of the Cloud Service Providers has PostgreSQL as an option out of a selection of relational database systems.

### 7.1.1 Patient Client

The client program is today implemented using Python, there needs to be three modules implemented to the existing software to get the telemedicine solution to work. A brief system design can be seen in Figure 9. The first module is the database module where all results can be stored locally before upload to the server.

The second module is the transmission module where the software establishes the connection to the server securely for sending training data. The last module is the video conference face-to-face module letting patients and doctors to communicate with each other. In this module Skype, Adobe Connect or an own solution utilizing Web Real-Time Communication (WebRTC) will be used.

### 7.1.2 Doctor Client

Regarding the doctor’s side of the system the solution could either be to implement a web portal or a standalone desktop program for the doctors to use to monitor the progression of the rehabilitation process. The web portal will always be available for doctors in all hospitals around the world, given that the IT-policies allows usage of external platforms in the daily work.

At VLL the IT-policies allows installations of external software on the computers used within the council’s network so a desktop program could also be used in this case. Other country councils within Sweden may not have the same generous IT-policies the same reasoning applies to healthcare organisations abroad. IN this

In this design choice both the web portal and a desktop program could be viable. Both have pros and cons presented below.

---

1. [http://spring.io/](http://spring.io/)
2. [https://webrtc.org/](https://webrtc.org/)
Figure 9: Brief description of the modules needed for the patient’s side in the telemedicine system

Web portal pros

- Can be used regardless of the operating system on the client computer
- Hosted directly from the server system
- No interference with IT-policies regarding installation of software
- Easy to make a mobile application
- Easy to make updates and changes to the client software
- Easy tools for web to make video conference functionality (WebRTC)

Web portal cons

- Cannot be used offline with downloaded data
- Potential greater downtime
- IT-policies which does not accept external web services for patient data
- Support of many different browsers

Desktop program pros

- Gets installed on every computer that the doctors use
- Allows offline usage with previously downloaded data
- There exist some functionality that could be used in the desktop program

**Desktop program cons**

- Support of different operating systems
- Harder to roll out updates for the desktop program
- IT-policies against installation of external software on client computers
- Harder to make a mobile application
- Could possibly be hard to make a integration with web camera for video conference

The safest one to choose would be to make a web portal which the doctors use to monitor the rehabilitation process and to communicate live with a patient with a video call where the doctor and patient meets “face to face”. A web portal would just utilize the RESTful API to gather the information from the server system in the same way a desktop program would do. The video conferencing functionality is achievable through WebRTC.

### 7.1.3 Server System

The server system is designed to use standardized external software, such as Java and PostgreSQL, to allow easy cloud deployment to enable high availability and to be scalable easily. Another feature in the design is that the server system is designed to both be deployed locally on the local hospital to only be used there, as well as it is possible to make a global cloud solution where all the hospitals using the telemedicine system.

![Diagram of server system](image)

**Figure 10:** Brief description of the modules used in the server system
7.2 Security of the System

In today’s medical systems the security is the most important aspect of the system. No malicious user should be able to extract any information from the system about a patient or about it’s medical diagnoses. Therefore the security mechanisms needed for these system needs to be robust and well tested.

The communication between parties within the system needs to be secure. This is done by using encryption and decryption of messages. The authentication of users is important to allow users their access rights. Then enforcing the authentication through authorization.

7.2.1 Encryption

Transmission of data will use the HTTP with the SSL/TLS extension, commonly refereed to as HTTPS. This makes the connection encrypted end-to-end and no one in between the connection point can read the traffic sent between two parties.

Stronger encryption could be necessary to meet requirements of keeping the patient data secure for a longer period of time. In some countries EHRs must be legally secured for 30 years. These cases motivates a stronger encryption method since it is proven that both RSA and ECC encryption is broken in a quantum computing environment[1]. This motivates the usage of the NTRU encryption method, this is then combined with the AES-256 method to create a strong and quantum resilient encryption of data.

In Sweden EHRs must be saved for 10 years since the last note into the EHR. This applies to both patient records (paper form) and the electronic records[3].

An other alternative to the NTRU-AES encryption combination would be to use a technique presented in literature by Amin et al. and Li et al.[2], [15]. Both of which suggests a cryptographically strong authentication scheme utilizing biometric information such as fingerprint or iris scanning. This to both authenticate users and for users to authenticate the server as well as making a secure exchange of session keys for encryption of messages sent between users and server.

This solution requires additional hardware to the system and risks complicating the system for people with reduced attention and partial paralysis which neglect patients commonly have. The solution also requires a slightly different system architecture to use since the scheme mutually agrees upon a session key used for encryption, instead of relying on SSL to handle that for you.

7.2.2 Database Security

To achieve security of the database a number of procedures can be done. First is to only allow access from the local machine or network. No one can then access the database directly from a remote location, to access the database the option is to use a Secure Shell (SSH) connection and then connect to the database for maintenance.

To allow the server software to access the database a new user role for the software needs to be created. The role should have the sufficient access attribute such as read and write.
By creating a separate role for the application to use creates an extra level of security since the user only can read and write to the database unlike the superuser which can do all of the user function in the DBMS.

One idea could be to have two separate roles, one for the patient client which only can insert results to the database and one role for the doctor client which only can read from the database.

### 7.2.3 Authentication

In healthcare systems the most important aspect is not to grant access to user who should not have access to the system resources. Therefore the authentication and authorization processes are very important. Because of this the authentication most often utilizes a 2FA. This could be done by a username/password login together with a one-time pin-code delivered from a SMS or smart-phone application that is connected to the user account in the system. Examples of smart-phone application would be the Steam Guard[^4] and Google Authenticator[^5] and the RehAtt authenticator would work in the same way as these two examples.

The method of authentication could, or rather would, differ depending on the country the system is used in. In Sweden with a integration against the digital infrastructure of the Swedish national healthcare systems there is a possibility to integrate the authentication against the healthcare platform 1177. The authentication is then performed by using BankID[^6] or one-time codes sent via SMS to the patient’s phone.

In other countries these functionalities may not be available and a integrations against the national systems may not be feasible. Instead more traditional authentication methods may be sufficient. These methods could be username-password, identity with biometric information, signed certificates, etc[^8].

The authentication method to use in the RehAtt platform is to be a username/password login for the local machine where the rehabilitation is performed. The rehabilitation machine then uses a shared user identity to authenticate to the server system.

The authentication process utilizes a challenge-response protocol where the authentication request starts with a challenge proposal, the server answers with a challenge. The requester then performed the challenge. To make a hash of the challenge together with the shared identity. The hash of the challenge is then analysed by the server and if correct the server issues the authorization token to use for future communication.

### 7.2.4 Authorization

When the users have authenticated with the system they will receive a JSON Web Token (JWT) that is used in the authorization process in future calls. This token is used in the Authorization header in the HTTP messages. The server will then extract the token on incoming calls and check it against the stored ones in the database to identify the user making the call and check whether or not the user has access to the resource or not.

The implementation of the server system could utilize libraries that handles the authoriza-

tion process automatically, such as Spring Security for the Spring framework. The process of

### 7.3 Transfer of Data

The data sent will be in JSON-format which is one of the most used data representation formats of web services and web applications. This is an easy to understand format for both humans and machines.

All communication with the server system is authorized by the authorization string acquired in the login phase. This string is sent with all messages in the HTTP header `Authorization`. Here follows a code snippet from a HTTP header with an authorization token.

```plaintext
... Authorization: Bearer eyJhbGciOiJSUzUxMiJ9.eyJzdWIiOiJiNjcz...
...```

Below follows an example of a request body sent to `/session/start` with HTTP-method POST. The HTTP body specifies the start time of a training session, upon a start request a random unique id is chosen and returned in the response to the user to use for the session. In the case of a start request like this, the start time of the session is specified in the request.

```json
{
    "start_time": "09:34:42"
}
```

Using XML format could also be viable since many of the systems utilizing web services uses XML as the data format of transmitted data [14]. However since the FHIR system allows both of XML and JSON and by professionals said to be a successor to the older openEHR format the JSON data format will be chosen to be used in the design of the system.

When the training session is started by a patient, the patient starts the rehabilitation process and plays the games. When a game is finished the results are uploaded to the server system. The request body is sent to `/game/{game_name}` with the HTTP-method POST.

As similar from the session example the games finished request contains the authorization pair to authorize the patient in the request header. The training session id the game belongs to and the connected meta data are sent to the server in the request body. The meta data in this case is the start time of the game, the number of levels played until the game was completed, the total time of finishing the game and the mean reaction time of catching the objects in the game.

```json
{
    "start_time": "2016-05-10 14:34:32.345234+02",
    "time_played": 34,
    "levelsPlayed": 7,
```
7.4 Database Design

A first draft of the Brain Stimulation telemedicine system database schema is presented in Figure 11. The design is not final and the database can be split into two to increase the security of the personal data.

The personal data stored in the server system in the Patient relation must be encrypted and the ones capable of decrypt the information must be the doctors who has the Cares relation to the patient. This lowers the threat level since the data cannot be traced back to a single user without the decryption of personal information. The l_name, f_name and SSN attributes will be encrypted while age, sex, neglect and user_id will still be accessible.

The training computers used by the patients will also have a database storage of information. In the local database the data yet not uploaded to the server system due to no connection to the server system. The database schema for the local databases is similar to the schema for the server system, and can be see in Figure 12.
7.5 User Roles

The users in the system will be divided into primarily two roles, patients and doctors. The doctors can however gain additional access to be “admin doctors”. The choice of using both role-based access control together with a touch of attribute-based access control is that eHealth systems must have a hierarchy of access levels. The base is the role-based where the roles are patient and doctor. Some doctors will gain the attribute admin of a hospital and thus gains additional access levels.

7.5.1 Patient

The patient accounts have the possibility to create training data. This data is created when playing the games in the training program. The data is uploaded to the server after a game is finished. The patient cannot retrieve any patient data. Not even their own.

7.5.2 Doctor

Only doctor accounts can access patient data. However the doctors are restricted to only be able to retrieve the data of patient they care. Each hospital has a number of admin doctors, these could access all patients data for their own hospital. The doctors are not able to perform training and create training data.

7.5.3 Admin Users

There will be one account in the whole system that belongs to Brain Stimulation that is used to create Hospital entries and the belonging doctor admin account for hospitals in the system.

The admin doctor accounts then administrate it’s patient accounts and sets up the Cares relation, issue new patient and doctor accounts, etc.
7.6 System Flow

The following subsections will describe the internal calls of the platforms of what is done when using the different clients and the server.

7.6.1 Patient Client

Below follows a number of code snippets of pseudo-code explaining the flow and what processes being executed when the RehAtt rehabilitation system is started and used with the telemedicine extension.

First a snippet explaining the general flow of the program. When started the user must login. If the user has access to the system a training session is initiated and created. Then the user plays games and when finished the training session is closed.

```python
start_program():
    login()
    token = initiate_training()
    while(playing):
        start_game(gamename, token)
    end_training_session()
```

The user must login to use the system, this is done by supplying the username and password. The login function will utilize a 2FA to increase the security in the login phase of the system. This can be done by either additional biometrics data from a fingerprint- or iris scanner, or via an external smartphone application or SMS delivering an one time code to be used in the authentication.

```python
login():
    username, password = get_username_password()
    status = check_credentials(username, password)
    if authorized(status):
        grant_access()
    else:
        show_login_error()
```

When initiating the training session the program starts the communication with the server system to retrieve a session token to use in the local system. If successful the token is stored and used in the system for the rest of the training session. If no connection to the server system is established a local token will be generated and used instead.

```python
initiate_training_session():
    token = get_server_training_session_token()
    if token is set:
        store_token(token)
        return token
    else:
        token = create_local_token()
        store_token(token)
```
A user starts the game from the menu, when a game is started the scene is set up and the game starts. At the end of the game the result is recorded and the game finishes.

```python
start_game(gamename, token):
    play_game()
    record_game_result(token)
    end_game()
```

When storing game results the result is first stored locally in the local database on the machine. Then the software tries to upload the result to the server system. If successful the result is stored in the server system database and then the local result could be removed.

```python
record_game_result(token):
    store_result_locally()
    if successful(upload_result_to_server()):
        remove_local_save()
```

### 7.6.2 Server System

Insert server side flow of request with diagrams and pseudo code

When a request is received by the server system through the REST API the following events are performed for most of the API end points, below follows a short code snippet of pseudo code and a figure describing the internal calls upon incoming requests, see Figure 13.

```python
receive_request(req):
    authenticate_request(req)
    check_user_rights(req)

    if has_rights():
        execute_request(req)
        return_response()
    else:
        return_error()
```
Figure 13: Internal calls of server upon incoming requests
8 Implementation of the Design

The telemedicine system is designed to be suitable for 2 programmers for 3-6 months to get the system up and running. The system can then be extended to be more robust and create integrations against national healthcare system for the countries where possible and needed.

In the demo implementation a version of the server system is implemented in Java using the Spring library to create a server with a REST API. Combined with this a database schema is created. The local patient client has been extended to allow storage of results in a local database and transmission of data to a server. The doctor client is built in to the analyse functionality in RehAtt. The following subsections contains more information about the demo implementation for each part of the system.

8.1 Server System

The server system is implemented in Java using the Spring library as stated in the design chapter. The server has a REST API for the clients to make requests to. The server can handle calls from the rehabilitation platform to insert data into training sessions for a user. In the demo the level of security is not satisfied for a fully developed and ready to launch healthcare application, by for example not handling user passwords in the server system. The login requests are sent to the server with a user id and password, but the passwords are not stored or checked, simply ignored. A user is granted access upon giving a correct user id, which is a UUID of version 4.

The server has a PostgreSQL database for storage of results, user accounts, etc. The database uses the proposed schema presented in the design in section 7.4.

The authorization of the system is performed with the library JSON Web Tokens for Java or Java JWT (JJWT). These tokens can be verified and has an expiration time and after that the token becomes invalid. The tokens are signed by a key which could be checked by the server to ensure the issuer is in fact the server system.

8.2 Patient Client

For the existing RehAtt software some development has been done to make local storage of results to a local database instead of saving results into text files in user unique folder on the local storage media. Both for making the storage more sophisticated and organized as well as simpler to retrieve and create new result entries. For this a database module is

[https://github.com/jwtk/jjwt](https://github.com/jwtk/jjwt)
built in Python and simply integrated to the existing software. The database handles all communication with the local database through the pyodbc library together with the ODBC driver.

For transmission of results to the server system a Transmission module has been implemented that handles all communication with the HTTPS protocol. The task of the module is simply to create the functionality of uploading results to the server system for the telemedicine solution, such as login and upload of results.

After each game the result is stored locally and the program tries to upload it to the server system. For now all locally stored results which are successfully uploaded to the server system remains in the local database. It is possible to make so if the result was uploaded the result would be removed from the local database. When a result in the local database is uploaded this is indicated for the result in the local database. This is designed to be the start of a syncing process between the patient client and the server system. Due to time constraints this synchronize process has not been finished.

### 8.3 Doctor Client

The Doctor client is simply a modified version of the analyse functionality in the existing version of RehAtt. In the analyse window the doctor can use a login functionality to login to the server system and then retrieve results for a patient. The doctor can only retrieve results for a patient which it currently have the cares relation to. This relation is set to hinder doctor accounts to access patient records that the doctor doesn’t actively. This functionality is specified in the Cares relation where the doctors actively treats a patient is specified.

The connections to the server system is done by the Transmission module and utilizes the doctors login function in the module. If the login was successful the module handles the token in the same way as for the ordinary patient client.

In Figure 14 the modified analyse window is shown. The extensions made to the window is added features for doctors to login to the server system and retrieving patient data. This is done from the drop down file menu in the top left corner. The login is then performed in a dialogue box where the doctor enters username and password.

The doctor analyses the results by choosing which game, and for the Haptris game which rotation mode and level, to analyse in the right side panel. When the analyze button is pressed the corresponding results are analysed and the result will be plotted and shown in the left panel, as for the example showing the results for a test profile of the Haptris game with vertical rotation and level 1.

The blue line in the plot represents the score of the game and the red indicates the mean reaction time for the user to catch the falling block. The ideal result for a patient would be that the score would increase over time and the mean reaction time would lower over time of the rehabilitation.
Figure 14: The analyse window in RehAtt analysing data for a test user
9 Discussion

In the following sections shorter discussions about the thesis work of what have gone well, not so well, what I could have done differently and discussion about some key areas of the thesis work.

9.1 Current Research in the Field

Security for patient data records is a hot topic since more and more systems are being digitalized. In this process the storage, transmission and general usage needs to be secure without leakages of personal data to unauthorized parties. The research done in the field tends to focus on preserving the privacy of personal information, by in some cases even not store any personal information about patients to instead only use different attributes to identify a profile with a diagnosis.

The method used to find published material for the literature study gave good results which made the foundation for the evaluation of the integration tools and the security methods used in the design. Since the security area is very broad it is likely that not all relevant information have been found and used in the literature study.

9.2 Security in eHealth Systems

The security level in national eHealth systems are high and rightfully so. No one would like to have their clinical data or EHRs leaked due to an attack. However the level of security makes it harder for innovation from small companies and self employed business into the healthcare sector.

It is hard to find information about how Swedish national healthcare system is structured and how to integrate applications to the national systems. Better material is needed to learn about the systems, how they are used and information about services available. This thesis contributes with an introduction that may serve as such material.

9.3 Authentication

One method from the literature study is presented by Amin and co-workers [2]. However this method requires additional hardware to be included in the platform. The smart-card reader together with the distribution of the smart-cards makes this approach a bit less desired. The distribution will most certainly be handled by the hospital staff which leads to more administrative work, which is not desired. This is a method that offers a secure way
of mutual authentication and initiation of encrypted communication between two parties.

This creates a crossing where the advantages of the scheme must be weighted against the drawbacks with additional hardware and smart-cards.

At the moment a professional from Brain Stimulation delivers and sets up the rehabilitation hardware where it is supposed to be used. If the company would continue with this delivering system of the rehabilitation hardware, the professional could also handle the distribution of smart-cards and register them to the patients who are going to use the system. So if the company would continue the current delivery system the Amin approach with smart-cards and biometric information could be feasible to use in the telemedicine system.

This system would be used like the SITHS\(^1\) service provided by Inera AB which is a identification system used to login to computers and system, access eHealth services and national agencies, physical access to buildings. This system of identification is used in the Swedish healthcare for identification for doctors and other employees. This could be an option since it is already used in Sweden but it has the same drawback as HIP SDK as it could only be used within the Swedish healthcare services.

One approach for the Swedish healthcare system is to use the HIP SDK and integrate the authentication with “Mina vårdkontakter”, by doing this the patients authenticate with BankID\(^2\) with their own smartphone. This gives access to the whole healthcare platform at [http://www.1177.se/](http://www.1177.se/) where residents can get health advice, renew their prescriptions, etc.

### 9.4 Resulting System Design and Implementation

The outcome of this thesis is satisfactory, since a foundation has been created for a future telemedicine system. Before the continuation of the development a re-evaluation needs to be performed to analyse if the integration tools has changed, if new ones have come up and is used and if the ones currently developed (“Stöd och Behandlingsplattformen” and HIP SDK) are ready to be used for healthcare application such as this telemedicine system.

The demo implementation is today a minimum working example where the patient client, the existing RehAtt software, can store results to a remote server through the HTTPS protocol. The doctor client consisting of the existing analyse window from RehAtt with extensions for doctors to login to the server system and retrieve patient results to use in the analyse window.

The security in the demo implementation is poor since access is granted upon giving a correct user id and simply ignoring passwords in the server system. The local profiles for patients in the rehabilitation is however password protected in the login phase. A deployable telemedicine solution for RehAtt would utilize at least a 2FA with either smart-cards or a smartphone application as presented in the design chapter.

\(^1\) [http://www.inera.se/siths](http://www.inera.se/siths)
9.5 Integration to healthcare systems

Developing a healthcare application is not as easy as “regular” development, in the sense of the need to familiarize with how the complex infrastructure of the national healthcare platforms operates and the strict security policies when handling personal data of patients. The system lacks documentation for external companies to easily find how to use the systems and which services that are available to use and how they are used.

The healthcare sector is not an ordinary market for application as can be seen for smartphone applications or web applications. The high security standards, complexity of infrastructure and that it is so hard to find good documentation for the system slows down or even kills the innovation for this type of healthcare applications when the applications have a hard time to reach the market.

Within the thesis work I would have needed to start the arrangement of a meeting with professionals from VLL earlier in the project to get more time with the information gained from the meeting/interviews to incorporate the knowledge into the design. This report includes a section that introduces the digital infrastructure of Swedish national healthcare and is one of the main contributions of this thesis.

9.6 Work Environment

I have been working from the office that Brain Stimulation has, located at Umeå Biotech Incubator, which is a incubator of several life science and biotech companies. Since no other software developers are currently employed in any of the other projects at UBI, I have worked mostly on my own without the benefits of discussing ideas and solutions with colleague developers and programmers in the daily work. The workplace is fantastic with nice colleagues and a very nice familiar feel. But I have missed the everyday talk with other developers and programmers during work time.
10 Conclusions

Contributions made within the thesis work are: 1) evaluation of the tools used for integration against national systems, 2) the requirements for the telemedicine solution based on the literature study, interviews and cooperation with Brain Stimulation AB, 3) the design of a telemedicine system together with a demo-implementation, 4) the report also introduces the reader to the digital infrastructure of Swedish healthcare system.

Within the thesis work an evaluation of systems and software tools used in healthcare today where conducted. One system and three tools where chosen based on the literature study. In the in-depth evaluation openEHR, FHIR and HIP SDK where included. openEHR and FHIR is two protocols for clinical data structuring, these two where interesting to evaluate further to see if either one of these are suitable to use in RehAtt, with possible future integration to national systems included. HIP SDK where included since it is a tool that makes an integration to Mina vårdkontakter to make secure authentication against the digital Swedish national healthcare system.

Requirements for RehAtt, both legal and functional, where identified in collaboration with professionals from VLL and together with a medical expert from Brain Stimulation. From the requirements a system design where created to fulfil the requirements. The design is created to be deployed locally or in a cloud environment. However, when using cloud resources the legal requirements of personal data is significantly increased to assure that the data stays secret from other non-authorized parties.

From the system design a demo implementation where created to demonstrate the functionalities of the telemedicine solution for RehAtt with a server system and a doctor client.

The thesis report also contains an overview and introduction of the national Swedish healthcare systems which explains how the system operates and how the data is accessed and stored in the national and the regional sub systems.

To summarise, the aims of this thesis were achieved in a satisfactory way and the future work is described in the following section.
11 Future Work

If Brain Stimulation would like to pursue the development of a telemedicine system for the RehAtt software a re-evaluation of the systems and tools and a evaluation of possible new software to be used in the system needs to be done. This of course if the development of the telemedicine system is paused after this thesis work is finished and restarted in a few years. For the Swedish healthcare market a stop now and a re-evaluation in 2-3 years would probably be the choice for the best telemedicine system with integration for retrieving and storing medical data from/into the national systems.

The best choice for the product would probably be to pursue the development of at least a synchronize system where parts of this system is integrated allowing to remotely store profile results from diagnostics and rehabilitation into the same profile, which would be a significant improvement for the product. Today a profile is stored on the local computer, which leads to if a patient uses two different computers at a rehabilitation center for example, the results from the different sessions would be stored separate from each other.

The authentication and authorization methods may be revised in future, the spread of the product needs to be included. If the product is only used in Sweden one option to use is be the SITHS card with the associated card reader. For a international approach the method could depend on the country if a own type of smart-cards with the Amin’s approach is feasible or not.

One obvious task for the future is surely to implement the system and to start using it. Some parts like the security in transmission and handling of patient data needs to be revised together with lawyers to make sure that the patient data is handled correctly for the Swedish and international healthcare markets, regarding the security classification, encryption methods and access controls to the data.
References


[23] F. Wiemer and R. Zimmermann. High-speed implementation of bcrypt password search using special-purpose hardware. In 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14), pages 1–6, Dec 2014.
Appendix A - List of acronyms

2FA  Two factor authentication
ABAC  Attribute-based Access Control
ACD  Access Control Decision
ADL  Archetype Definition Language, used in openEHR
AES  Advanced Encryption Standard
API  Application Programming Interface
AVISPA  Automated Validation of Internet Security Protocols and Applications, a simulator for active and passive attacks on security schemes

cADL  constraint form of ADL
DBMS  Database management system
DLP  Discrete Logarithm Problem
ECC  Elliptic Curve Cryptography
eHealth  Electronical Healthcare systems
EHR  Electronic Health Record
FHIR  Fast Healthcare Interoperability Resources
FOPL  First Order Predicate Logic
HIP  Health Innovation Platform
HL7  Health Level 7 – develops standards within the healthcare area
HTTP  Hypertext transfer protocol
HTTPS  Hypertext transfer protocol Secure
IoT  Internet of Things
JSON  JavaScript Object Notation – Common format in data transmission I web services
JWT  JSON Web Token
NHS  National Health Service - Publicly founded healthcare in UK.
NTjP  Nationella tjänsteplatformen - National Service Platform (ISH)
ODIN  Object Data Instance Notation
openEHR  Healthcare system used in various countries, Sweden included
PDL  Patientdatalagen – The Patient Data Act
PHR  Personal Health Record
PUL  Personuppgiftslagen – The Personal Data Act
RBAC  Role-based Access Control
REST  Representational State Transfer, used in web services
RSA  Rivest-Shamir-Adleman (cryptosystem)
SDK  Software developer kit
SITHS  Identification system with eIDs
SNOMED-CT  Systematized Nomenclature of Medicine – Clinical Terms
SSH  Secure Shell
SSL  Secure Socket Layer
SoB  Stöd och Behandlingsplattformen
TLS  Transport Layer Security
VLL  Västerbottens läns landsting (Västerbotten County Council)
WebRTC  Web Real-Time Communication
XML  eXtensible Markup Language