DEGREE PROJECT IN COMMUNICATION SYSTEMS, SECOND LEVEL
STOCKHOLM, SWEDEN 2015

# Cloud-Based Alerting System for IP-Telephony

*A prototype development*

PER-JOHAN SIMON JAKOBSSON

**KTH ROYAL INSTITUTE OF TECHNOLOGY**
***INFORMATION AND COMMUNICATION TECHNOLOGY***

# Cloud-Based Alerting System for IP-Telephony

## *A prototype development*

Per-Johan Simon Jakobsson

2015-07-19

Master’s Thesis

Examiner and Academic adviser
Gerald Q. Maguire Jr.

Industrial adviser
Christer Ulfsparre

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

## Abstract

An increasing number of people in Sweden are having problems with their hearing ability. The three major tools to aid hearing-impaired and deaf individuals are: hearing aids, special telephony, and alerting systems. Both hearing aids and telephony have seen a huge technical development. Hearing aids have gone from huge ponderous devices to small delicate in-ear devices. Simple text telephones have evolved into total conversation telephones with audio, video, and text all operating in real-time. Although smart lamps and other alerting services not specifically made for hearing-impaired individuals do exist, the development of alerting system is unsatisfactory. The gap in technology is a huge problem and integration between modern products and alerting systems is getting harder. This thesis explores how to close this gap. The result of this thesis project is a prototype that provides the missing technological link between an alerting systems and modern smart devices. An eventual product should support all kinds of services, but the prototype is limited to solving the problem of connecting an alerting system to a modern total conversation telephones. The prototype was evaluated and based on the evaluation data a timeline was created. An overall positive response towards the product exists and the timeline had adding more third party services (such as Skype and FaceTime) as a high priority. The complete timeline as well as adding Signal Initiation Protocol support is left as future work.



### Keywords

Cloud-Based services, alerting system, IP-telephony

## Sammanfattning

I Sverige har antalet personer med hörselskada ökat de senaste åren. För att hjälpa de med hörselproblem finns det tre viktiga hjälpmedel: hörapparater, special telefoner och varseblivningssystem. Stora teknologiska framsteg har skett för både hörapparater och special telefoner. Hörapparater har gått från stora otympliga apparater till små nätta anordningar som man har i örat. Enkla texttelefoner är idag komplexa system som stödjer både video, ljud och text i realtid. Även fast smarta lampor och andra varseblivningsprodukter existerar så är utveckling för varseblivning speciellt gjorda hörselskadade och döva undermåliga. Gapet som skapats mellan moderna varseblivningsprodukter och varseblivning som hjälpmedel växer sig allt större. Denna rapport ska undersöka detta gap. Resultatet av detta projekt är en prototyp som tillhandahåller den teknologin som ska länka modern varseblivning och varseblivning som hjälpmedel. Den tänkta produkten kan användas för många olika tjänster men i detta projekt är den begränsad till total konversations telefoner. Prototypen har blivit utvärderad och en tidslinje, baserad på utvädringen, har skapats. Tidslinjen ska beskriva kommande tjänster och enheter som skall kunna användas tillsammans med prototypen. Det visar sig att den skapade prototypen blev positivt mottagen och att tjänster som Skype och Facetime skulle ha hög prioritering på tidslinjen.

### Nyckelord

Molnbaserade tjänster, varseblivning, IP-telefoni

## Acknowledgments

I would like to thank my whole family for supporting me through the years of study. Special thanks go to my mother (Reené Jakobsson) and my father (Per-Martin Jakobsson) and my siblings (Jakob, Adam, and Jossefin Jakobsson) as they helped me through the years. I also want to show gratitude towards Pierré Eriksson, Bengt Åkerblom, Karina Eriksson and Nanna Vollen for their support through my life in Stockholm.

I also want to thank my examiner and supervisor (Professor Gerlad Q.MaguireJr.) who guided my through the writing of the report and came with constructive feedback and tips on what to do next and how to improve my thesis. His insight and opinions have been very helpful throughout the project.

Lastly, I want to thank Omnitor AB for giving me the opportunity to work with them for this project. Specially Christoffer Fridbreg and Christer Ulfsparre which took their time to help me out , discuses ideas, and guide the project.

Stockholm, July 2015
Per-Johan Simon Jakobsson

## Table of contents

## List of Figures

## List of Tables

## List of acronyms and abbreviations

| | |
|---|---|
| 3G | Third generation |
| 4G | Fourth generation |
| AB | Stock Company |
| API | Application Programming Interface |
| ASC | Alerting System Controller |
| ASC-Cloud | Alerting System Controller Cloud |
| ASC-Admin | Alerting System Controller Administration |
| ASC-Device | Alerting System Controller Device |
| ASC-Server | Alerting System Controller Server |
| AWS | Amazon Web Services |
| CI | Cochlear Implants |
| CPE | Customer-premises equipment |
| DIP | Dual in-line package |
| DTLS | Datagram Transport Layer Security |
| EC2 | Elastic Cloud Computing |
| FM-radio | Frequency Modulation Radio |
| GPDL | Graphical Profile Definitions Language |
| GPIO | general purpose I/O/ |
| GSM | Global System for Mobile |
| HDMI | High-Definition Multimedia Interface |
| HRF | Swedish National Association for Hearing-impaired |
| I/O | Input/Output |
| IoT | Internet of Things |
| IP | Internet Protocol |
| IPsec | Internet Protocol Security |
| IPv6 | Internet Protocol Version 6 |
| IT | Information Technology |
| ITU | International Telecommunication Union |
| JSON | JavaScript Object Notation |
| MD5 | Message-Digest |
| MiniPC | Mini Personal Computer |
| MySQL | My Structured Query Language |
| NAT | Network Address Translation |
| PC | Personal Computer |
| PIC-micro | Peripheral Interface Controller micro |
| RDS | Relational Database Service |
| RF | Radio Frequency |
| RJ-45 | Registered Jack |
| S3 | Simple Storage Service |
| SD-memory | Secure Digital |

| | |
|---|---|
| SEK | Swedish Krona |
| SIP | Signal Initiation Protocol |
| SIP-Server | Signal Initiation Protocol - Server |
| SK2-SS | Sidekick II Strobe Light |
| SQL | Structured Query Language |
| SRTP | Secure Real-time Transport Protocol |
| SSL/TLS | Secure Socket Layer / Transport Layer Security |
| TC-device | Total Conversation Device |
| TCP | Transmission Control Protocol |
| TTY | Telecommunications Device for the Deaf |
| USB | Universal Serial Bus |
| USB-alert | Universal Serial Bus - alert |
| VPC | Virtual Private Cloud |
| VoIP | Voice over Internet Protocol |
| Wi-Fi | Wireless Fidelity |

# 1 Introduction

This chapter describes the specific problem that this thesis addresses, the context of the problem, and the goals of this thesis project. Section 1.5 describes the research methodology used in this thesis, while Section 1.6 delimits the scope of this thesis project. The chapter concludes with an outline of the structure of the thesis.

When a machine wants to signal its user it can show information on a display, play a ring tone or play a tune, or in some other way contact the user through their human senses. An alerting system is a system designed to notify its user. Human interaction software today frequently implements an alerting system. When mentioning alerting systems most people think of only tools for aiding hearing/seeing impaired people perceive information that others perceive easily. However, alerting systems are actually far more general (see for example [1]).

Today alerting systems are advanced and integrated with the Internet and often communicate via smart-phones and tablets [2]–[4]. Alerting systems created for hearing-impaired and deaf individuals usually use radio and line carrier communication, rather than being Internet based system [5]–[10]. This thesis project explores the technological gap between current alerting systems and the out-of-date alerting systems made for hearing-impaired and deaf people.

The project began with a background study of alerting systems and ended with the development and evaluation of a prototype alerting systems that takes advantage of the Internet. This prototype is an alerting system that combines a client and a cloud-based server. Potential users of the prototype are the current customers of Omnitor AB [11], these are people with hearing problems. The prototype should bridge old and new technologies pushing the development for new alerting systems.

## 1.1 Background

An alerting system is a system that takes an input and then decides how to alert the user. One example is a coffee machine that indicates when coffee is ready by emitting a sound, sending an e-mail message, or flashing a light to signal to the user. Would I rather have my coffee maker call my cell phone than make a beep sound when my coffee is ready? This may lead to my selecting one product rather than another or perhaps configuring the product according to my preference. One group of people that more or less depending on an alerting system to live a normal life are those individuals that are either deaf or hearing-impaired, but the development of these systems are lagging behind what is technically and economically possible.

Today there exist 1.4 million people in Sweden with a hearing disability that affect their ability to perceive what others are saying [12]. Some of these people were born with hearing problem or even deafness, but it is much more common for these problems to develop over time. More than half of people facing this problem are between 16-64 (54%) years old and the rest 65-110 (46%) [12]. For a person with hearing disabilities or hearing loss it is crucial to have tools to aid them.

Special Telephones, hearing aids, and alerting Systems are some of the most common tools for people with hearing problems. Both special telephones and hearing aids have had a rapid technological growth during the last 50 years. Telephones for deaf and hearing-impaired have gone from simple text-telephones too much more advanced total conversation* devices (TC-device) which support real time video, speech, and text. The development of hearing aids has gone from simple electronic devices to implants inside the ear, for a complete list of the advancement of hearing aids see [13]. Total Conversation is a concept promoted by the International Telecom Union (ITU). Total

* ITU-T Rec. F.703 describes the Total Conversation service.

Conversation concerns the usage of telephones for people with disability. Apart from sending audio and video, it is also important that the text is sent one character at the time with the same intervals as its typed (also known as “timed text”) [14].

However, the one technology that has not improved very much is alarm and alerting systems made for hearing-impaired and deaf people [15]. A simple alerting system for hearing-impaired and deaf people consists of a sender, for example a doorbell-button, and a receiver, the doorbell; but instead of playing a tune the receiver flashes a light or vibrates to inform the deaf user that someone has pressed the doorbell button. There are many products offered by companies that offer a variety of services, the most common ones are alarm clocks, baby-monitors, doorbells, and motion sensors [16]. After looking at these products, it is notable that many companies utilize wireless technology for the communication between the sender and receiver as a replacement for their previous wired solutions. The wireless technologies used are various custom 2.4 GHz solutions or 868.3 MHz / 433 MHz solutions, the latter being the same technology used in older garage door openers [5]–[10], [17].

Although the development of alerting systems specifically made for hearing-impaired and deaf people has been slow, other types of alerting systems have been rapidly developed. The Internet of Things (IoT) is a new concept where everyday objects (“things”) have a connection to an infrastructure (the Internet) enabling communication between them. Several IoT products offer some alerting-features. These systems communicate over IP via Wi-Fi, Ethernet, Bluetooth, and wide area cellular systems (GSM, 3G, or 4G). Utilizing the Internet enables several features that were impossible in earlier simple alerting systems.

IoT is a fuzzy term and often described in different manners depending on which entity has defined it. In the European Research Cluster on the Internet of Things report “Internet of Things Strategic Research Roadmap from 2011” the following description is given:

> *“Internet of Things (IoT) is an integrated part of Future Internet including existing and evolving Internet and network developments and could be conceptually defined as a dynamic global network infrastructure with self configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes, and virtual personalities, use intelligent interfaces, and are seamlessly integrated into the information network.”* [18]

## 1.2 Problem definition

Current alerting systems created for deaf and hearing-impaired people are underdeveloped and use old technology compared to the rapid growth of regular consumer alerting systems. There is a need for an alerting system that can both handle the older technology and modern IP based technology. This project will investigate the usage, development, and evaluation of such a system in order to answer the question:

> *Is it possible to develop a general cloud-based alerting system which is compatible with current alerting systems and that can evolve to and exploit newer IP-based systems? If so, how would such a system be realized?*

## 1.3 Purpose

This thesis discusses the needs of hearing-impaired persons, current alerting systems for hearing-impaired persons, development of the Internet of Things, and possible solutions to fill the gap between current alerting systems and the Internet of Things. Finally, the thesis presents the design, implementation, and evaluation of a prototype that offers a solution to the problem described in the previous section. The target group for this thesis is individuals interested in software development, networking, and alerting system for a specific group of people, in this case specifically deaf and hearing-impaired people.

## 1.4 Goals

The goal of this project is to create a prototype of a system for alerting hearing-impaired and deaf people. This system should be able to be remotely triggered. This high-level goal is split into several sub-goals:

- Perform a market investigation of current alerting system specialized for hearing-impaired or deaf individuals both in Sweden and internationally.
- Design an alerting system that uses a cloud-based architecture.
- Propose different system solutions that fulfill both Omnitor ABs and user's system requirements.
- Develop a prototype based on the most appropriate solution.
- Evaluate the prototype.
- Document and present the work.

The deliverables for the company are a part of reaching the overall goal and include:

- A working prototype meeting the minimal requirements and
- Documentation of the system to enable further development.

The benefits provided by this project are direct improvements in alerting system for hearing-impaired and deaf individuals and indirect improvements based upon educating other developers as to how they can exploit IoT, cloud, and networking technology in their development of alerting systems. The users of the final products will experience several benefits. Some of these benefits are increased choice of products from the market (the system can be used together with different brands of alerting devices), use their choice of alerting system with popular software (such as Skype), and remote support (where a support technician or service can connect directly to the device to carry out remote diagnostics and repair). The final product should work with any TC-telephone. For this project, the prototype is limited to working with Omnitor´s TC-solutions and products of common alerting system brands.

An ethical issue that occurs in this degree project is the usual ethical problem that exists when developing software. The user's data needs to be handled in such a way that no personal information is leaked, i.e., to preserve the user's data integrity. Additionally, intellectual property has to be respected; no source code should be used without the owner's permission. Use of encryption has to be done in a manner that follows the laws of the country where the device is to be used. [19]

Sustainability is an important factor in software development. However, this factor is usually overlooked in actual development efforts. In order to be able to deliver software that is more economical and environmentally friendly different aspects of sustainability have to be considering throughout the software's lifecycle [20]. While the degree project mainly focuses on the development process, notes about maintenance, production, and usage aspects have been added where appropriate.

## 1.5 Research Methodology

An in depth description of the research methodologies and methods are presented in Chapter 3, while the philosophical assumptions, research methods, and research approach will be presented here.

The philosophical assumptions concern the basis of the researcher's view of what is acceptable knowledge and what is reality. Philosophical assumption comes in three different main flavors: positivism, realism, and interpretivism. Positivism assumes that understanding reality can be achieved by applying natural science to all fields. Realism is a similar to positivism in that sense that it also assumes that the reality can be understood through natural science methods, but realism assumes that natural science can be used to figure out an explanation - rather than natural science directly giving an explanation. The interpretive assumption is that reality is created by detailed observation of people in natural settings; it is the complete opposite of positivism. [21] For this thesis project, I have chosen an interpretive assumption as the best match for this project, as reality is not based on a traditional scientific method, but rather has a qualitative aspect.

After making our philosophical assumptions, the choice of research method defines how the researcher will carry out the research. Popular research methods include experimental, non-experimental, descriptive, analytical, fundamental, applied, conceptual, and empirical research [22]. For this thesis project, I have chosen to use the applied research method, because the background study done focuses on the development of a particular application, specifically to create a prototype of a potential future product.

The next step is to decide upon a research approach, which covers how conclusions are drawn and how to establish if they are valid or not. The most common research approaches are inductive and deductive. The inductive approach starts from a specific situation and then forms general ideas and conclusions as the research goes on. Deductive is the other way around, starting by collecting general ideas to get to a specific situation. [22] For this thesis project, I have chosen the deductive research approach because the project explores different technologies until we reach a system design that perfectly fits the requirements from Omnitor AB.

There are two basic categories of research methods: quantitative or qualitative. A quantitative research method is used when dealing with numbers, data, and measurements such as scale, range, and frequency. Qualitative on the other hand involves examining, reflecting, dealing with feelings and opinions, and development. [23] I have chosen a qualitative research method as we are dealing with the user interpretation and experience of the system. Also a qualitative research method is frequently used when developing a new system [22], which is exactly what this project attempts to do.

## 1.6 Delimitations

The market analysis carried out in the background study is far from a complete picture of all products on the market and companies offering products in this market. The companies included in this analysis are either listed on the "Assistech portal for assistive technology" [16] or companies for which Omnitor AB is a retailer. For the Internet of Things (IoT) products, a list on Postscape [24] was used.

During the development of the prototype, the technology chosen reflects the knowledge and experience of both Omnitor AB and the author of this thesis. This does not mean that different architectures, hardware, and software were not considered, but rather that the technology alternatives considered were not exhaustive.

The prototype developed during this degree project is simply a proof of concept prototype. A proof of concept prototype is the simplest version of a potential future product, and it is missing both the industrial design and many of the features that would be found in actual products [25]. The prototype is not expected to be aesthetically pleasing, but should provide most of the basic functionality necessary to show the importance and usage of the system.

## 1.7 Structure of the thesis

Chapter 2 presents the relevant background, going deeper into the current situations of the tools used by hearing-impaired and deaf individuals. The background continues into a brief look into the current alerting system market, Omnitor ABs role, and finishes up with earlier work that is related to this project. Chapter 3 goes into the research methodologies, explaining how the project will be carried out, both on scientific and practical level. The chapter will give a detailed explanation on how data is going to be collected and evaluated. Chapter 4 states the steps taken in order to realize this project, design choices and implementation steps are explained in detail. The chapter also presents how the prototype was created. Chapter 5 gives an explanation on how the prototype was evaluated and presents the data. The data is analyzed and a conclusion is extracted from the result. The last chapter 6 gives a conclusion on this project and thesis, going into a deeper view of the result, and through limitation and what the next logical steps for continuing work on the project.

# 2 Background

This chapter provides basic background information about current alerting systems and their market. Additionally, this chapter describes the company Omntior AB and its need for the product and the new product's requirement(s). The chapter also describes related work with home automation and cloud-based services.

## 2.1 Technological gap

The need for tools to aid hearing-impaired and deaf people has increased during the last few years as the number of people with hearing problems has increased. In Sweden the percentage of hearing-impaired people has gone from 11.3% of the population in 1987, to 14.3% of the population in 2007, and is now 17.0% of the population [12]. According to HRF (Swedish National Association for Hearing-impaired), people with hearing disabilities exist of all ages. However, it is more common amongst older people (75-84 or 84+ years old) to have hearing disabilities, but the largest group is between 55-64 years old [12]. Also more than half of the people with hearing disabilities are working (ages 16-64) meaning that customizations of their workplace might be necessary [12].

Recently there have been rapid developments in products for people with hearing problems. There are three common aids: hearing aids, telephony, and alerting systems. Hearing aids have gone from a simple larger device embedded into glasses frames to ultra-thin delicate in-ear-devices. Telephones for the hearing-impaired were initially text based, providing a service called TeleTYpewrite (TTY), and via a relay-service could provide communication between a hearing-impaired person and a non-hearing-impaired person. Today these telephones support real time video, audio, and text – thus providing a TC-device (as described in Section 1.1). The one major category of aid that has not seen much development has been the alerting systems [15].

An alerting system is a system created to alert/inform the owner of an event in other ways than the original product allows, for example providing a flashing light for a telephone that normally signals an incoming call through an audio output (i.e., a ringing tone). Hearing-impaired and deaf people can use an alerting system to ease their everyday life, while avoiding the problems that might occur with ordinary products that have not been designed for these users [15].

### 2.1.1 Current Alerting Systems Technologies

To understand the technologies used today in alerting systems for deaf and hearing-impaired, a brief presentation of current products is helpful. Assistech [26] is a company specializing in finding appropriate assistive products for people with disabilities and special needs. Assistech is also a retailer of the most popular alerting systems, and their website has a wizard that helps user find and list these products. Some of the companies that Assistech mentions are: Bellman & Symfon [5], Clarity [6], Serene Innovations [8], SilentCall [9], and Sonic Alert [10].

By using the Assistech wizard to find the manuals of the different products, it is possible to determine what kind of technology the product uses. Assistech also states that an alerting system is built from senders and a receiver; where a sender sends a signal to the receiver that alerts the user in some way. Unfortunately, in most cases senders and receivers created by different companies cannot be used together, as they are not compatible with each other. One reason for this is that no standard exists for this communication. Assistech further claims that there are two kinds of technologies available for alerting systems: radio frequency and line carrier. Radio frequency technology sends signals wirelessly using radio waves where interference is avoided by altering the frequency or a numeric code for each device using a built-in dual in-line package (DIP) with several switches. This same technology was used in older garage door openers [17]. Line carrier technology make use of the home (or office) electrical circuits to carry the signal between the sender and receiver [26].

Table 2-1 shows a list of technologies used by alerting system companies / products as stated in each product's manual. These products either are described in the company's technical catalog, as in the case of Bellman& Symfon, or are a new product released by the indicated company. The first column states the company's name and the product that was examined. If a product is present, then the company's entire catalog was looked at. From this table we can see that most companies use FM-radio configured via DIP-switches to pair their senders and receiver; although different frequencies are used by some of these products. Only Serene Innovations does something different, as according to the manual [27] their product CA-360 Visual Alert System use Smart-Code, although what Smart-Code is has not been documented. Also worth noting is that Serene Innovation's product are not using Wi-Fi, although they are using the same radio frequency band as Wi-Fi. Furthermore, according to the manual you cannot put a CA-360 close to a regular Wi-Fi router, as they transmit and receive on the same frequency, but do not use the same protocol.

**Table 2-1: Different companies and one of their newest products and which technology it uses. Note that a question mark is used when the frequency is unknown.**

| *Company - Product* | *Technology* | *Frequency* | *Manual* |
|---|---|---|---|
| *Bellman & Symfon* | FM-radio configured via DIP-Switches | 868.3 Mhz | Technical Solution 1.4 [28] |
| *Clarity – AL10* | FM-radio configured via DIP-Switches | ? | Clarity Alert Master AL10 [29] |
| *Serene Innovations CA-360* | Smart-Code | 2.4 GHz | Model CA-360 Visual Alert System [27] |
| *SilentCall - SK2-SS Sidekick II* | FM-radio configured via DIP-Switches | 418 MHz | SK2-SS Sidekick II Manual [30] |
| *SonicAlert – DB100* | FM-radio configured via DIP-Switches | ? | DB100 instructions [31] |

The technologies presented in Table 2-1 either are old or do not following any standards (and sometimes they are both), but they are representative of alerting systems for deaf or hearing-impaired individuals. In contrast, one can look at alerting systems in the consumer market, where a very different picture is revealed. In the world of Internet of Things (IoT), with Internet connected everyday objects, several high tech solutions can do the same things as the products of the specialized companies listed by Assistech. See for example the Wi-Fi connect video "doorbell" products: Ring* and Skybell†. Moreover, code exists for Arduino and other platforms to make your own such device see example: http://lifehacker.com/5908850/build-an-arduino-powered-doorbell-alert-system-that-sends-you-a-text-and-photo.

Internet of Things (IoT) is a new term that has many different meanings in the literature. However, the words Internet and things suggest that IoT involves some kind of connectivity to generic objects (things) via the Internet using standard Internet protocols. This means that IoT consists of a word-wide network of interconnected objects using standard Internet communication protocols. Today a central issue is how these different things communicate with each other to create autonomous behavior that benefits users [32].

* https://ring.com
† http://www.skybell.com

The market for smart everyday IoT objects is relatively new, but there exist many IoT products. For example, a Wi-Fi enabled light bulb that can be controlled from a computer or smart phone. Such a product is available from companies such as LIFX and Visualight [33], [34]. Home sensor systems integrate IoT products in the home to give the user more information about the home's status and power consumption [4]. Additionally, some companies such as Spark [35], are enabling other companies to develop new IoT products by facilitating rapid construction of prototypes using existing modules.

When comparing the alerting system products made specifically for individuals with hearing problems and IoT products we can see some strength and weaknesses of the former type of systems. We will compare the following attributes for the two types of systems: signal range, security, and usability. First of all the range, FM-radio has (roughly) a fixed range depending on frequency and output power, while IoT solutions connected to the internet give an effective range that is only limited by internet connectivity (although the first hop range to get from the device to the Internet might be limited by the communication technology used for this link). When Internet connectivity is missing, then new hardware might need to be installed. As Internet connectivity is built out, the range and accessibility of IoT products are increased. As to security, IoT products are much more advanced as they can run on top of Wi-Fi, GSM, and 3G, which all have built in link layer security. Moreover, the devices can also make use of IP layer security (via IPsec), transport layer security (via SSL/TLS, secure real-time protocol (SRTP), Datagram Transport Layer Security (DTLS) ...). The signals sent by FM-radio can be recorded and replayed, thus unintentionally or intentionally triggering alerting system receivers. The usability of IoT products differs between the different products. Different products use different ways to connect to internet, different higher layer protocols, and different applications to alter their settings. Alerting systems usually have at most some DIP switches that need to be altered, making them easier to install compared to some IoT products. On the other hand, alerting system components generally lack interoperability with components from other vendors, while IoT products can often work together or at least use the same signal receiver (typically a smart phone). IoT products can offer services that alerting systems for hearing-impaired /deaf cannot, as the former systems are connected to the Internet and hence can be integrated with many other systems - making them more flexible than the current alerting systems for deaf and hearing-impaired persons.

### 2.1.2 Omnitor AB´s new solution

The technological gap between modern day IoT devices and alerting systems for hearing-impaired and deaf people is an opportunity for companies delivering special telephones for the hearing-impaired and deaf – as these companies can potentially develop new products to fill this gap. Omnitor AB [11] is a company delivering modern IP-telephony systems with TC. New TC telephones run on tablets, smart phones and computers, enabling lightweight, compact devices with high quality video performance. The problem is how to connect a total conversation application running on a tablet (such as Samsung Tab S [36]) to an alerting system.

Today Omnitor offers two solutions for connecting a smart tablet: USB-alert [37] from Polycom and Bellman's Mobile Phone Sensor [38]. USB-alert operates by closing a relay that is controlled via a USB interface. To activate an existing alerting system the USB-alert is connected to an alerting system's transmitter that reacts to a closed a circuit (for example, a doorbell that actives when the button is pushed). This solution can be used with all alerting system that utilizes a sender that reacts to a closed circuit. A tablet using USB-alert must have support for USB-devices and must avoid problems with charging the tablet (when the USB-alert might be activated). Bellman's Mobile Phone sensor is a small egg like device with a camera on it that is supposed to balance on the screen of your mobile phone (or tablet in this case). When the tablet screen lights up due to an incoming call the small egg acts as an altering system sender by sending signals through wires to a Bellman alerting system receiver. Unfortunately, when using this product you are unable to use your tablet, because you cannot tilt or move it around, as the egg would then fall off. The Mobile Phone sensor is only compatible with Bellman´s alerting system products. Both products need wires to work, thus making the otherwise mobile tablet into a stationary device hence the user cannot move their tablet around.

Due to the weaknesses of current technologies, Omnitor AB required a new product to connect TC-devices on tablets with alerting systems. This new solution should not only resolve problems with connecting tablets to alerting systems, but should also serve as the missing link between smart IoT products and alerting systems. The idea is that this new product should bypass many of the problems of existing alerting systems problems, while also being compatible with them. This leads to the following requirements:

- Use standards (use standards as much as possible),
- Platform independent (the system should be able to be used regardless of the hardware or software the user uses for telephony),
- Smart (the system should be able to process data and just not be a simple circuit),
- Cloud-based (the system should have the majority of its logic in the cloud, rather than in the user's end device(s)),
- Built-in security (compared to traditional alerting systems, this system should implement some form of security),
- Be compatible with current alerting system (if a user already has an alerting system at home, then the device should be able to connect to it), and
- Be compatible with IoT devices (if a user would rather use a smart light bulb than an alerting system receiver, then it should be possible to implement this use case using the new product).

## 2.2 Related Work

There are several things to consider given Omnitor's interest in a cloud-based alerting system. The first is that currently there exist no similar products within the field of alerting systems that are specialized for the hearing-impaired and deaf individuals. Second, in other fields similar products may already be gaining popularity. For these two reasons, we need to examine related work.

### 2.2.1 Home Automation

Smart homes and home automation are two concepts that aim to offer users control over their home in order to increase their comfort, safety, and efficiency. One example would be to enable your refrigerator to know what is inside it, enabling user feedback and information about what products he/she needs to purchase. Another example would be to give the user control over all the lights in the house, as well as to program their behavior. [39]

A cloud based alerting system made for modern IP-telephony should work in the same manner as a very basic home automation system. When a total conversation phone receives an incoming call, other devices in the home could be activated. A lamp could be flashed or a traditional alerting system could be activated. The goal is that this new system should operate in the same manner as home automation system is triggered and activates or deactivates a device.

Andreas Hallberg and Oskar Lundh have written a paper on home automation as a an alternative solutions to environmental problems [40]. In their paper they discuss and show a simple implementation for an home automation system that uses a Smart Phone and Raspberry Pi [41] together with remote controlled electrical outlets to control lamps and other mains powered devices. Figure 2-1 shows their setup as used in a simple home automation system. The central device is a Raspberry Pi computer that runs server software written in Java. This server handles incoming connection requests, stores state, and can on command send out a radio signal through a radio interface to turn on and off the different electrical outlets [40].

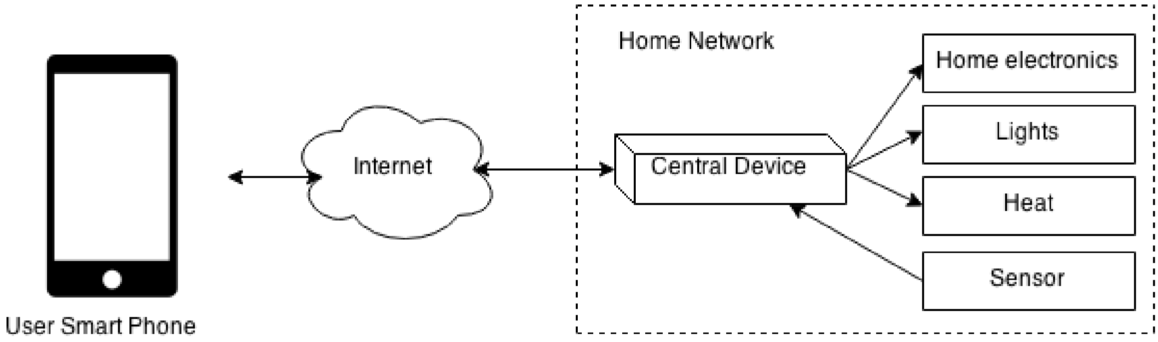


**Figure 2-1: Home Automation Setup. (Inspired by Figure 1 on page 11 of [40])**

The main advantage of using a Raspberry Pi as a server is that this computer supports many kinds of devices via USB, HDMI, RJ-45, and other standard interfaces. The device is credit card sized and energy efficient. The Raspberry Pi has a lot of support and been used in many hobby and professional projects. Several different operating systems are supported. The price of the Raspberry Pi is lower than many of its competitors, although some simpler devices, such as microprocessors are of course cheaper [41].

Using the Raspberry Pi as a server has both advantages and disadvantages. First of all the most obvious advantage is that by using an existing device there is no need for server hardware development. The main disadvantage is that all logic has to be implemented in this device, forcing it to store all of the settings and requiring greater performance from the device. Additionally, there may be problems when updating and altering settings if this process has to be done at every device. This is a particularly large problem if many devices are deployed. There is also a need to configure the firewall in the home router, especially if it uses Network Address Translation (NAT).

One of the alternatives to avoid the problems of NAT is to use IPv6, see for example the thesis project by Thor Hådén entitled IPv6 Home Automation [42] – where he realizes a home gateway to allow IPv6 control of devices attached to a RF controlled power strip. To securely reprogram IoT devices one should consider the methods proposed by Mussie Tesfaye, in his master’s thesis [43].

### 2.2.2 Cloud-Based Services

One important concept that needs to be considered thoroughly is cloud-based service. The term “cloud” is often defined depending on what kinds of services are delivered. Cisco Systems, Inc. defines a cloud in this manner:

> *“A cloud is a powerful combination of computing, networking, storage, and management resources, enabling a new generation of consumer and enterprise IT services that can be available on demand and delivered economically to any device anywhere in the world without compromising security or function.”* [44]

Many reports and papers have been written about clouds and cloud-based services. In the new solution desired by Omnitor, the cloud should be the backbone of the system, thus it will handle most of the logic and the data of the system. This reduces the computational and networking burden on each of the individual devices, while maximizing the interoperability – as different data format and protocols can be supported by the cloud, without the need to modify the data sources (senders) or receivers

Amazon Web Services (AWS) [45] is internet based cloud computing infrastructure created by Amazon Inc. in 2006. AWS offers a huge variety of cloud based services that are scalable and offers a pay-as-you-go method where users pay for exactly what they are using, thus making the service very cost efficient and removing the need for overprovisioning of each individual service [46]. The following brief explanation of the different products and how they are related was taken from the list of products on Amazon [47]:

| | |
|---|---|
| EC2 (Elastic Computing 2) | EC2 provides virtual servers inside the Cloud |
| VPC (Virtual Private Cloud) | Provides a virtual cloud inside the Cloud to isolate cloud resources from each other |
| CloudWatch | A service used for monitoring resources inside the Cloud |
| S3 (Simple Storage Service) | A fully redundant data storage infrastructure for storing data on the web |
| RDS (Relational Database Service) | Offers easy to setup database in the Cloud (MySQL, SQL Server, and many other databases are supported) |

### 2.2.3 Single-user Profiles for Alerting Systems

In an article by Doris Jung and Steve Jones [1], they introduce the concept of a Graphical Profile Definitions Language (GPDL). GPDL is a simplification of linking alerting system with user profiles that triggers on specific events. Even though there exist similar solutions with XML/XPath, GPDL is unique in that their target group is the general population rather than specialists. Their article provides one example of how GPDL can be used. A library is connected to a notification system and the users defines their interests so that the library will notify the user when a certain book arrives. A user can then use GPDL to set the preferences for his or her interests.

The solution presented by Jung and Jones is a very general solution and not locked to a specific area. Comparing the GPDL solution for creating user profiles to connect with alerting systems with the alerting system solution requested by Omnitor, there are some major differences. GPDL is an interface for users to define profiles that alert the user via an already established alerting system infrastructure and triggering services. Omnitor's new product is a link in the alerting system infrastructure connecting old technology with new technology, specifically by using Omnitor's Session Initiation Protocol (SIP) server to provide triggering events. In its simplest form Omnitor's new product does not need any user configuration, but when adding additional features to this new product user customization should be added and GPDL could be used for this.

## 2.3 Other related work

Related work that is used as inspiration and insight of how the system should be designed is of great value in this project. Omnitor's new product must be designed with the ease of access and usage in mind. The entities that need to be designed have to be identified.

### 2.3.1 Remote Residential Control System

A remote residential control system is a home automation system, but on a higher level. Such a system is used to control different devices that in turn make the whole system control things, such as energy management, security surveillance, and consumer electronic. Ming-Shaung Lang has written a thesis on “Remote Residential Control System” [48] where guidelines to creating such a system are described and evaluated. This thesis discusses different standards for device communication and network management. The thesis ends with a brief review of trends and upcoming technologies. The thesis was written in 2004, but still has information that holds today. The thesis describes information about technologies that *should* be considered when designing a remote control system.

### 2.3.2 Service in the Cloud

In the thesis “Security as a Service in Cloud for Smartphones” by Laksmi Subramanian [49] in 2011 there is a detailed description of a cloud based solution. The idea behind this cloud service is to bypass the problems with battery life and CPU power that smartphones have. This is done by shifting some of the calculations to the cloud, focusing on processing on what is relevant to improving the security of smartphones. The thesis shows how the cloud can be used to provide functionality that otherwise would greatly limit the operating lifetime of the device.

### 2.3.3 SIP as Service Provider

SIP, defined in RFC 3261 [50], is a protocol used for starting, stopping, altering, and creating sessions. SIP has many uses, for example: internet telephone calls, multimedia distribution, and multimedia conferences. Omnitor’s new product is supposed to react to calls that are initiated through Omnitor’s SIP-Server. These calls initiate sessions that carry audio, video, and text.

In a master’s thesis by Martin Altinkaya and Saman Ahmedi from 2001 there is a presentation of SIP as a protocol as well as a description of how to use SIP for interconnecting and as a service provider [51]. In their thesis, the similarities between TCP and SIP are drawn, explaining the initiation of IP-telephony calls. One of the main goals of this thesis is how to solve interconnection between different protocols. The thesis concludes that connecting several different protocols by using SIP is possible, but the task would require a huge amount of work.

## 2.4 Summary

After exposing the problems with the technological gap between alerting systems for the hearing disabled and modern IoT products, a new solution is required. The new solution requested by Omnitor should be seen as the first step towards modernizing alerting systems. When comparing entities such as current alerting systems, IoT products, USB-alert, and Bellman´s mobile phone sensor we can see both advantages and disadvantages with the different systems. Table 2-2 lists the current solutions to the connection problem: tablet to alerting system, with both advantages and disadvantages.

The same type of summary can be done for simple implementations of home automation systems. Table 2-3 shows the advantages and disadvantages of different parts of the implementation of a home automation system. Note that these advantages and disadvantages were chosen to be relevant based upon the new alerting system requested by Omnitor AB.

**Table 2-2: Advantages and disadvantages of different devices and their technologies.**

| Device | Technology | Advantage | Disadvantage |
|---|---|---|---|
| **Current Alerting System** | Radio-FM | Simple Implementation | Unsafe<br>Limited Range.<br>No compatibility |
| **IoT Product** | Wi-Fi, GSM, 3G | Secure, Reachable from Internet<br>Can be connected to using API | Wi-Fi needs to be setup<br>Hard to implement |
| **USB-alert** | USB to close circuit | Simple design<br>Easy to use. Compatibility with other products | Prohibits the movability of tablets<br>Problem with support and power charging |
| **Bellman´s Mobile Phone Sensor** | Camera | - | Render device useless while on it<br>Incompatible with other products |

**Table 2-3: Advantages and disadvantages of the implementation of home automation system.**

| Implementation | Advantage | Disadvantage |
|---|---|---|
| **Raspberry Pi as client controller** | Cheap and functional compared to other similar products<br>Supports many different devices<br>Supports many different programming languages | More expensive compared to microprocessors |
| **Run server on user device** | Do not need back-end server | Needs to manage all configuration and settings<br>Hard to control many devices<br>Needs firewall settings |

# 3 Methodology

The purpose of this chapter is to provide an overview of the research method used in this thesis. Section 3.1 describes the research process. Section 3.2 focuses on the data collection techniques used for this research. Section 3.3 describes the experimental design. Section 3.4 explains the techniques used to evaluate the reliability and validity of the data collected. Section 3.5 describes the method used for the data analysis. Finally, Section 3.6 describes the framework selected to evaluate the prototype created in the project.

## 3.1 Research Process

As this project uses a qualitative research method, the research process itself has several steps containing examining and evaluation, often from the perspective of feelings, opinions, and reflection upon what has been done. The basic structure of the research process begins with an empirical research on the topic of already existing technologies and related work that can be useful for the design/implementation phase. After the empirical research phase, there will be a design phase in which different options are presented and evaluated. This evaluation is done by comparing the advantages and disadvantages of different designs with the list of requirements given by Omnitor (see Section 2.1.2). Following the design phase there will be an implementation phase. In this implementation phase the system is constructed resulting in a prototype. The final phase is the evaluation phase. The evaluation phase is the most important phase as it deals with collecting data from test users and interpreting this data. Based on the interpreted data, a detailed timeline for when new features could be introduced in updates or in a series of similar products is estimated. Figure 3-1 shows how the different phases relate to one another.

The empirical study is a straightforward phase as it simply looks at today's solutions for alerting systems for hearing-impaired and deaf people. In addition, studies of Internet of Things (IoT) solutions will be done so the gap between these two technologies will be exposed. Related work, such as home automation systems and previous work on cloud computing is included in this phase. The goal of this phase is to gain a reasonable perspective on the current market, what technologies are used, and possible solutions that could be applied in the design phase.

In the design phase, some major decisions must be made. These decisions will affect the outcome of the whole project. The network architecture is to be decided upon; this is necessary, as different network architectures will affect the system's design. The hardware for both the server and the client have to be decided. Lastly, the communication has to be designed, both in terms of protocol and data format (including individual fields and their encodings). All of the decisions in the design phase will have an impact on the implementation phase as the choice of hardware, protocol, and architecture may limit the system on many different levels.

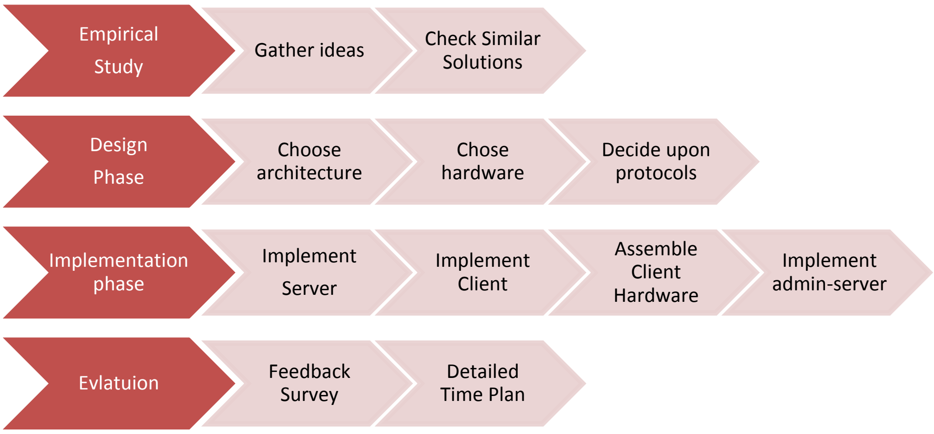


**Figure 3-1:** **The Research Process and its different phases and a small explanation**

Although all the different phases are equally important, the implementation phase is the largest in that sense it is expected that this phase will consume more time than the other phases. There are four different parts: the server, the client, the client hardware, and an administrative server. The client hardware needs to be assembled - unless there is a complete product at hand. For example, if the client hardware requires specific accessories to perform basic alerting system operation, then a peripheral will be used or another platform has to be considered. A basic alerting system operation could for example close a relay in order to activate the alerting system or turn on a light. The server, the client, and the administrative server have to be implemented through software development. This software development will use ideas from agile methods mainly scrum.

Scrum is an agile software development method that is drastically different from the waterfall model. In the waterfall model, development is streamlined going from point A to point B in a straight line. The waterfall model leaves little to no room for improvement as development takes place, hence only with great effort can such unplanned tasks be performed. The agile development methods tries to counter the problems of the waterfall model by putting emphasizes on collaboration, constantly having functioning software, exploiting flexibility, and adapting to emerging business realities. Agile methods are guidelines, while Scrum is an implementation of agile software development methods. Scrum provides concrete steps, specific improvements, and distinct roles. [52]

In the project the improvements adopted from scrum will be a scrum board, sprint iterations, and scrum meetings. The scrum board will be a collection of prioritized tasks that can be moved around, altered, and fixed. The board will be the main tool to ensure that development is on time and to decide what features are added or not. The sprint iterations divide the working period into short periods of about 2-3 weeks. Dividing the development into sprints increases flexibility. Each sprint begins with the selection of the tasks that together can create a minimal working version of the prototype. At the end of the sprint, the minimal version should be treated as a small release, which should be reviewed and critically evaluated. It is during this evaluation phase of the sprint that changes can be made and features can be added or removed. New changes can be applied to the next incremental version that will be implemented in the following sprint. During the scrum meetings tasks will be discussed and time for each task estimated. The scrum meeting will be held at least once per sprint, more meetings might be added if a particular problem occurs.[53]

The last phase is the evaluation phase in which the system will be tested by test users. These test users will be users with particular needs for such systems or users that have previously used USB-alert or Bellman’s Mobile Phone Sensor. After the test period is over, these test users will be given a simple survey (see Appendix A) where they can review the system as well as add information about it and in which direction they think the development should continue. The test users will be from different areas of Sweden: Stockholm, Karlstad, Uppsala, Örebro, and Linköping. Although the geographical distances are large, the number of participants is limited to one or two users at each location. It is important to have test users in different locations as different rules apply when it comes to aid distribution. Geographically distributed users will give a fair evaluation of the new product. The sparse number of participants is due to the limitation of hardware that can be produced for this project. Another limitation is also finding testers that uses Omnitor’s TC-device and can give high quality feedback. The feedback will be interpreted and used for creating a detailed timeline for when new features can be introduced.

The time for the project’s implementation will be divided accordingly (For the day-to-day work the scrum board is used). See Table 3-1 for duration of each of the phases.

**Table 3-1: Duration for each of the four phases**

| | |
|---|---|
| Empirical Study | 4 weeks |
| Design Phase | 3 weeks |
| Implementation Phase | 6 weeks |
| Evolution Phase | 7 weeks. |

## 3.2 Data Collection

The data will consist of feedback received from the test users. The format of the data should include some qualitative answers to a set of questions - along with answers to some open questions (that will enable the test users to write a small text as an answer). Some questions will have several alternative answers, but every question should have a field where the user can enter their thoughts in their own words. The data collection method, questionnaire and case study, fits this kind of evaluation. Both can deal with smaller groups and suit a qualitative research method. The questionnaire method supports both closed and open questions; hence, a questionnaire fits the data collection needs of this thesis project. Case studies are normally used together with the case study research method.

### 3.2.1 Sampling

Sampling will be done through an online web form where users can choose alternatives and fill out answers to open questions. The answers will be saved in a database and later examined. This database will be encrypted and the web form will be provided to the user using SSL/TLS to provide a secure channel, enabling the user's input to be confidential and the data to be safely stored. The questions should answer the following questions:

- Type of user (Ordinator that orders the product[*], Installer that installs the product for the user, User that actually uses the product itself)
- How easy is it to use the product?
- Which of the product's properties are important?
- What services should be added in the future?
- What other smart devices should be compatible with the product?
- Any other comments (There might be something that was never considered)

### 3.2.2 Target Population

The target population is limited to people who need to replace an USB-alert or Bellman's Mobile Phone sensor. This group will be distributed over major cities across Sweden, specifically: Stockholm, Karlstad, Uppsala, Örebro, and Linköping. Each user will be provided with a prototype and the user has the opportunity to use it actively for two weeks before they answer the questionnaire. After two weeks, the user is expected to have been able to receive enough calls to build an opinion of the prototype.

[*] The ordinator is the person that handles the purchasing and ordering of aids for people with disabilities.

## 3.3 Experimental design/Planned Measurements

To determine the research strategy, several different qualitative methods were examined. Both exploratory research and case study research methods were considered, as both are recommended for qualitative research, but neither of them fit the needs of this thesis project. Exploratory research was rejected as it uses surveys to gain insight into the problem and rather than exposing solutions, it focuses on finding key issues of the problem. A case study was also rejected due to use of an empirical investigation that did not fit the artifact development part of this project. [22]

The research strategy chosen for this thesis project is action research as it mimics the agile method used to develop software, more specifically in our case developing a small portion of the complete hardware and software system. Figure 3-2 shows the cycle used for action research, the steps are: take action, observing, evaluating, and critical reflection. [22]

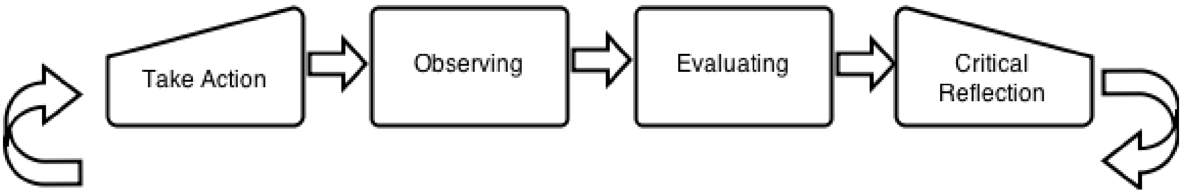


**Figure 3-2: Research strategy of action research**

When developing software with agile methods there exist some cyclic steps that work much in the same manner as action research. Figure 3-3 shows the entire software development life cycle. A user story is created and a plan for what it should contain in each release is discussed. When the release-planning phase is complete, then the software development process enters a loop of development and small releases, followed by feedback on all changes. Once a minimal product is created, the prototype system can be put into use although the loop of development and small releases may continue. Compare this to action research: the action taken is the development phase of the loop while observation, evaluation, and critical reflections are similar to the small releases. [54]

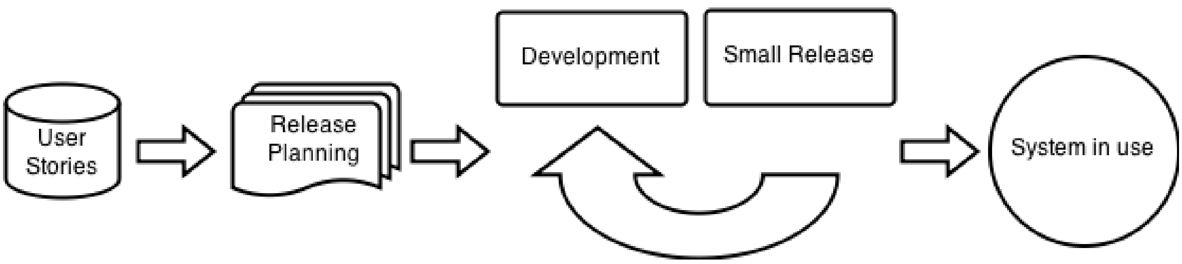


**Figure 3-3: The Software development life cycle. (Inspired by Figure 1 on page 2 of [47])**

The research is driven by the requirements from Omnitor AB. Sets of ideas are proposed for a small release, then these ideas are sorted. Following this, an action is taken, leading to one of these ideas being implemented. Once this implementation is complete, then the result is presented and observation and evaluation take place. Before selecting more ideas to implement, critical reflection is performed. To ensure that the research is going in the correct direction changes and alterations can be proposed during the critical reflection phase.

## 3.4 Assessing reliability and validity of the data collected

The data collected will consist of answers to open questions where the respondent can write in their own words and answers to closed questions where the respondent is given several alternative answers to a question. From this data, the system is evaluated and a time line created for the future implementation of additional features. The interpretation of the data is also used to answer the problem statement.

### 3.4.1 Reliability

The reliability of the data is based on conformability and dependability. The data must have been collected in good faith without personal bias that could alter the result. The tester benefits from being given a truthful answer to the survey, as if the prototype is successfully introduced into the market they might be offered such a product in the future, while if the prototype is a failure then the feedback could be used to improve the product or lead to any further development of it being discontinued. Omnitor would not benefit from exerting undue influence, as it is their interests to gain high quality feedback. Conclusions will be drawn using logical reason based on the collected data and the reasoning should be easy to follow and understand.

### 3.4.2 Validity

To ensure that the data is valid the respondent needs to confirm that the data is interpreted in the correct way. This will be done by presenting the results to each of the respondents, enabling the user to correct misinterpret data when they read the result. To provide validity to conclusions and result, the reasoning should refer to the data in such a manner that the reasoning can be recreated.

## 3.5 Planned Data Analysis

The mixed qualitative and quantitative data gathered through the data collection method has to be analyzed. The only data analysis method considered for this report is coding as it enables the qualitative data be converted to quantifiable data and allows a greater picture to be drawn from the data [22]. To do this conversion, labels and grouping will done to the data and similar answers will be mapped together into groups. The result will be presented in diagrams and the conclusions will be drawn from these results.

As the data set is small, the coding can be done by hand with the assistance of some tools. The diagrams will be generated using Microsoft Office Excel*. Excel was chosen because of the small amount of data and processing that needs to be done. Another reason Excel was chosen was that Omnitor provided the software. To draw diagrams, the free online application draw.io will is used.

* http://www.microsoftstore.com/store/mseea/sv_SE/pdp/Office-Hem-och-Sm%C3%A5foretag-2013/productID.263156400

# 4 Realizing a cloud-based alerting system

A system for solving the problem of connecting new TC-Devices to alerting systems for hearing-impaired has been implemented. The new system is an Alerting System Controller (ASC) and is made up of three entities: a device located in the user's home (the ASC-Device), a server inside the Cloud (ASC-Server), and an administration web-interface (ASC-Admin). To realize Omnitor's new product (ASC-System) several design choices had to be made. In addition, software supporting the design choices has been implemented.

## 4.1 Network Architecture Decision

The first decision to be made before any actual implementation is done is to decide what kind of network architecture the new system should use. As stated in the requirements, the service is supposed to be cloud based, although eliminating this requirement and considering non-cloud solutions should also be evaluated.

Four different architectures were considered: local, direct connection, cloud-based, and cloud-connected. All of these architectures have the same goal: when communication is initiated between two parties, the receiving party is the alerting system that should be activated. This activation should utilize a device placed at the receiving party's location (typically their home). This device will be called the Alerting System Controller Client (ASC-Device), while the cloud component will be referred to as an Alerting System Controller Cloud (ASC-Cloud).

Figure 4-1 to Figure 4-4 shows the four different architectures. The entities in the figures are static and reoccur in more than one example, but in some architectures, they are missing. Before introducing the proposed solution there are four already established entities: Alice´s TC telephone, Bob's TC telephone, Omnitor's SIP-Server system, and Alice's current alerting device. In these examples, Alice is going to receive a call from Bob; when an incoming call occurs, she needs to have her alerting device activated. It is assumed in the example that the alerting device is of a common brand.

### 4.1.1 Local Architecture

A **local** architecture refers to the local connection between Alice's TC telephone and her ASC-System. Figure 4-1 shows the static entities and the ASC-Device located at the user's premises. Although not shown in the picture, there is a chance that both Alice and Bob are behind a firewall running in the router that implements their local area network. For this local setup to work without altering settings in this home router (or routers), Alice's TC-device needs to be on the same local network as the ASC-Device. An example call flow follows the steps shown in the figure: (1) a call is placed from Bob to Alice, the call passes through Omnitor's SIP-Server. Alice receives a call from Bob and her TC-device notices the incoming call through an application implemented in this device (2). The TC-device contacts the ASC device, which then activates the alerting system (steps 3 and 4).

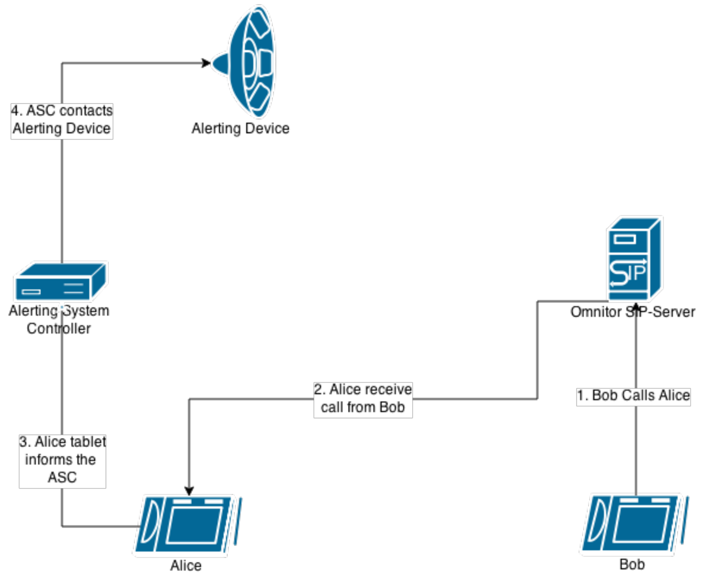


**Figure 4-1: Local architecture.**

### 4.1.2 Direct Connection Architecture

The **direct** connection architecture gets information directly. This information is sent directly to the ASC from Omnitor's SIP-Server. Once again, it is important to consider that there may be a home firewall as without configuring this firewall, the firewall could prevent this setup from working. This is of course avoidable by having the ASC-Device talk SIP with the SIP-Server. Figure 4-2 shows the direct connection setup. This time an incoming call from Bob goes to Omnitor's SIP Server (step 1). This SIP-Server then sends the call information to Alice's TC-device and establishes a connection to the ASC to send the alerting information (steps 2 and 3). The ASC device activates the alerting device (step 4).

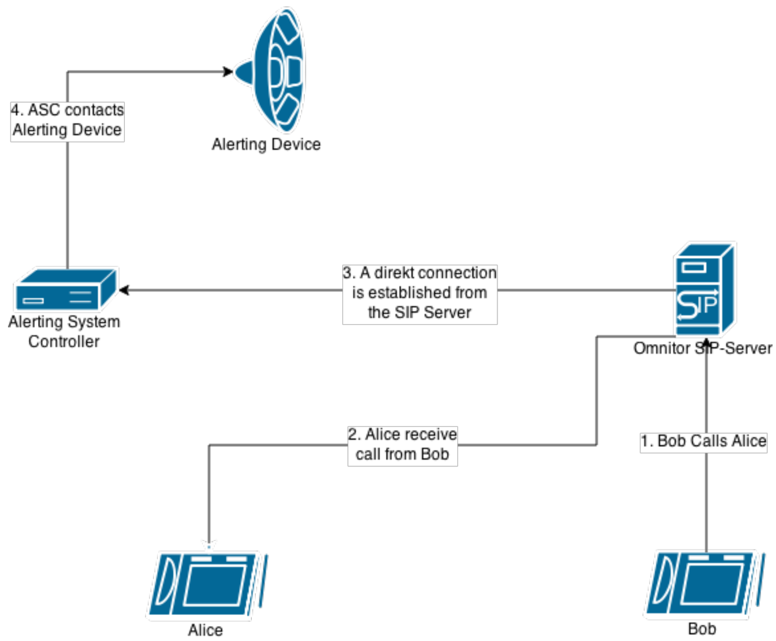


**Figure 4-2: Direct connected architecture.**

### 4.1.3 Cloud-Based Architecture

A solution that is a little more complex would be to use the same basic architecture as the local case, but with a back-end cloud added. This kind of solution is referred to as **cloud-based** and is presented in Figure 4-3. As in the local architecture, the cloud-based architecture also requires the ASC device and Alice's TC-device to share the same local network, as otherwise the home router might prevent them from communicating. When a call is placed by Bob to Alice, it goes via the Omnitor's SIP-Server and Alice's TC-device receives the incoming call (steps 1 and 2). Alice's TC-device contacts the ASC device that establishes a connection to a server running in the cloud to determine what to do (steps 3 and 4). The cloud signals back to the ASC to activate Alice's alerting device (steps 4 and 5).

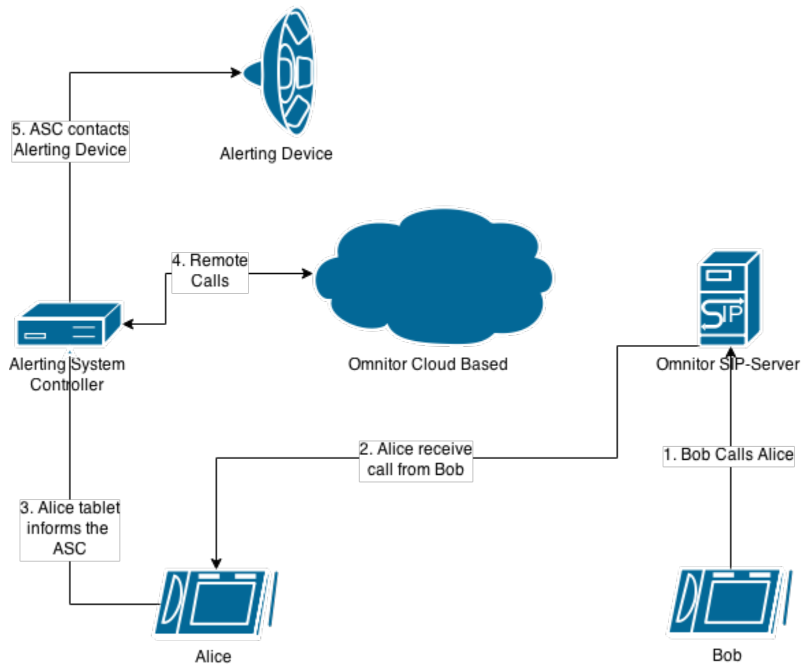


**Figure 4-3: Cloud-based architecture.**

### 4.1.4 Cloud-Connected Architecture

The last solution adds a cloud to the direct-connect solution. This **cloud-connected** solution is very different from the previous versions as its *independent* of router's firewall. The setup does not require Alice's TC-device and ASC-Device to be connected to the same network. Figure 4-4 shows this setup is very similar to the direct-connect version, but with a cloud in the middle. When Bob initiates a call to Alice via Omnitor's SIP server (step 1 and 2)., the SIP server sends an extra message to the cloud service (step 3). As the ASC-Device has previously established a connection to the cloud service, the outgoing connection is able to penetrate the local router's firewall. Over this connection, the ASC-Device receives a command from the cloud and activates the alerting device (steps 4 and 5).

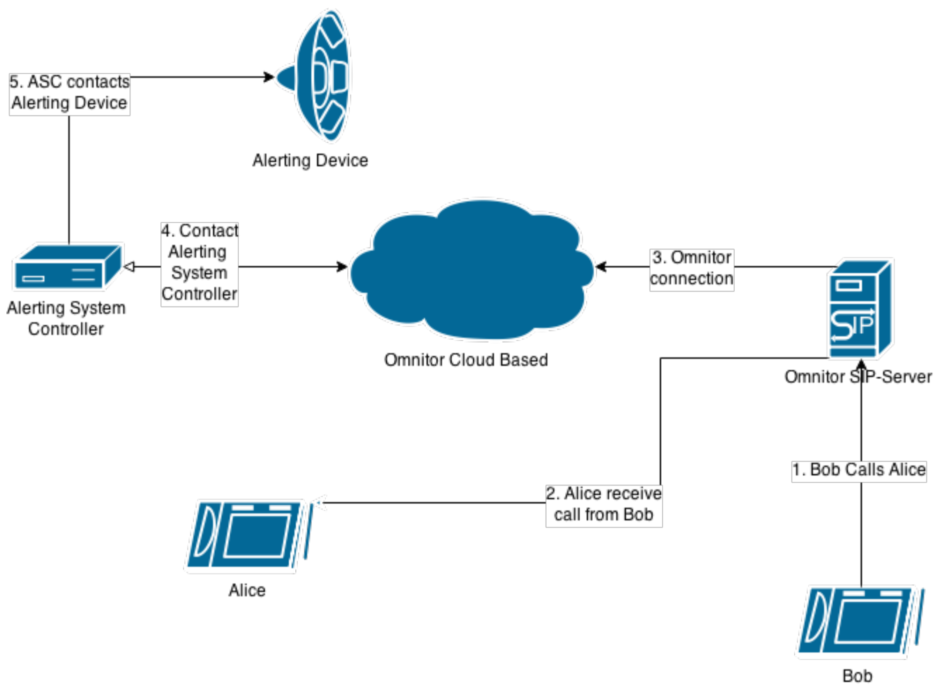


**Figure 4-4: The cloud-connected architecture.**

### 4.1.5 Summary

A quick comparison between the four architectures quickly exposes their advantages and disadvantages. Table 4-1 shows the advantages and disadvantages of these four different architectures. The first column indicates which architecture is evaluated and the columns to the right list the advantages and disadvantages of this architecture. With these advantages and disadvantages in mind, the cloud-connected network architecture was chosen because of its flexibility. It is also worth noting that the need for a cloud implementation is listed as a disadvantage, but only from a network architecture perspective. As in practice the usage of a cloud includes a large number of advantages from a usage perspective, as it gives better control and a remote control and monitoring center can be used to manage, change, and update the system *without* the users needing to alter any of their ASC-Devices.

It is important to keep in mind that the network architecture should not depend upon Omnitor's current resources (such as the SIP-Server), but rather be an independent system. The reason for the system to be independent is to increase support for third parties and prevent creating a system that only can be used by Omnitor. The ASC-System should be a complete separate package that can be handed over to other parties.

**Table 4-1: Advantages and Disadvantages of the different architectures.**

| Setup | Advantages | Disadvantages |
|---|---|---|
| **Local** | No back-end needed<br>Can penetrate firewalls | Must be on same network as TC-device<br>Needs implementation in the user's TC-device |
| **Direct-Connect** | No back-end needed | NAT router problem if SIP is not used. |
| **Cloud-Based** | Can penetrate firewalls | Needs implementation in the user's TC-device<br>Need for extra implementation for the cloud |
| **Cloud-Connected** | Can penetrate firewalls<br>Is network independent | Need for extra implementation for the cloud |

## 4.2 Platform Decisions

In the cloud-connected architecture, there are three custom entities: the ASC device, the ASC cloud, and finally the signal from Omnitor's SIP server. Omnitor's SIP-Server already exists, so there is no need to decide upon a platform for that entity. The ASC-Device needs to be a small product that easily can be moved around and have modern connectivity such as USB, Ethernet, and/or audio/video. Omnitor does not want to run the ASC-Cloud on their own server(s), but rather expects this service to run on a popular cloud service provider.

### 4.2.1 Alerting System Controller Hardware

Several different hardware platforms were considered for the ASC-Device. Some properties have been given a greater weight than others have, but the most important properties are minimal work, model compatibility, and suitable inputs and outputs. Minimal work was a very important property as development time adds to the cost and if a solution is too expensive, and then it would not achieve high market penetration. Model compatibility concerns what models of the device are to be released and what the compatibility should be between the different models. For example, will there be a small or large amount of work when there is a new release of the hardware. Having suitable inputs and outputs is very important as the device must be able to connect to the Internet and open/close a circuit. Opening and closing a circuit could be done with simple relays or with external devices, such as the previous mentioned USB-alert.

The hardware that has been considered for Omnitor's new product includes a PIC-micro controller, miniPC, Raspberry Pi, CuBox, and Ethernet relay card. A Microchip PIC-micro controller is brand of programmable microprocessor [55]. Such a microprocessor can be programmed using a low level programming language and this microprocessor provides a wide variety of input/output pins. A miniPC is simply a small PC with all the benefits a normal PC has. Many different companies produce miniPCs, such as ASUS [56]. The Raspberry Pi is produced by the Raspberry Pi Foundation [57]. The Raspberry Pi is a credit-card-sized computer and this platform has been used in many different projects. Additionally, there are a large number for accessories for this platform. The SolidRun [58] CuBox is a powerful Android computer with all the standard input/output of a typical computer. Their aim is to provide high quality products at an affordable price. Lastly, there is an Ethernet relay card. This simple ~~a~~ card or device has several relays together with an Ethernet port. These cards/devices are connected to the network and remotely controlled over an IP network. Ethernet relay cards are manufactured by many different companies and are available for a wide variety of prices.

The PIC-micro controller is by itself the cheapest alternative by far, with prices from Elfa [59] starting at around 10 SEK. However, the ASC-Device also needs an Ethernet port, circuit board, connectors, cables, and relays to realize the desired platform. A disadvantage for the microcontroller-based solution is the huge amount of work that needs to be done. To make this into a solution there has to be a protective cover, soldering of the processor and other chips to a circuit board, connecting of wires, and testing the device. Moreover, if the alerting device to be controlled is powered by 220 Volt power, then there is a need for the device to be certified by a testing laboratory so that it can get a "CE" or similar mark. Because of the last reason it is highly desirable that existing devices with suitable type approvals be used.

The miniPC is a robust computer although it is often physically smaller than a typical PC. The price of a miniPC's can vary widely depending on model and vendor. Because a miniPC lacks programmable pins, there is a need for an external device such as USB-alert or similar accessory. Note that there exists miniPCs with programmable pins, but the most common versions of miniPCs are made to replicate a normal size computer. For examples of popular miniPCs *without* programmable pins see: [60]–[62]. A miniPC often possesses the full capabilities of a large computer, but is smaller in physical size [63].

The Raspberry Pi device is the same device as was mentioned in Section 2.2.1 with regards to the report by Andreas Hallberg and Oskar Lundh. As seen in their home automation project a Raspberry Pi is a good choice to control the home. The biggest problem with the Raspberry Pi is that extra accessories are needed. For example, Raspberry Pi needs a relay board to close circuits and a USB connected Tellstick to control NEXA devices. Another concern is that the Raspberry Pi will need a case, SD-memory, WiFi-dongle, and micro-USB power supply to be used [40].

CuBox is a competitor to the Raspberry Pi providing a lot of computing power in a complete Android device in a case. The only problem with the CuBox is that it is missing programmable pins; this means that a product such as USB-alert device must be used with a CuBox solution. The CuBox was created with media displaying in mind.

Ethernet relay cards come in different price ranges, but have the advantage that they do not need any additional work to fulfill the basic requirements, making them a perfect match to the requirements. The problem with an Ethernet relay is that it acts like a server and although APIs are present, a problem occurs when adding such a device to the cloud-connected architecture. In the cloud-connected architecture, the connection must be established from the ASC device to the cloud (to avoid the need to configure the router's firewall), rather than the other way around. However, there exist Ethernet relay cards that are Internet enabled with a programmable or pseudo programmable interface, for example the SIP relay module created by Stentofon Baudisch*. These devices are SIP enabled and can traverse home routers, the only problem with these devices are the lack of extendibility. Compared to a computer, the Ethernet relay modules do not contain a USB interface or audio output, thus limiting the number of new features that can be added to a single device.

* http://www.stentofonbaudisch.com/produkte/sip-relaismodul

Table 4-2 shows the different platforms with their advantages and disadvantages. The PIC-micro controller and Ethernet relay cards seem very static compared to the more generic miniPC, CuBox, and miniPC, as all of these devices have several extension ports. The miniPC is overprovisioned in that it would have a lot of CPU power and many features used that are likely to never be used. Both the CuBox and MiniPC often need an USB-alert or similar device to interact with accessories (such as relays) making the Raspberry Pi the best fit. In addition, the wide support of the Raspberry Pi will reduce development costs.

**Table 4-2: Advantages and Disadvantages of different hardware platforms for the ASC device.**

| Hardware | Advantage | Disadvantage |
|---|---|---|
| **PIC-micro controller** | Full Control<br>Cheap alternative | Huge amount of work<br>Needs a lot of accessories<br>Low level programming |
| **MiniPC** | Robust with computing power<br>Can run Windows/Linux<br>Easy to support | Needs USB-alert<br>Overprovisioned |
| **Raspberry Pi** | Widely used, supported<br>Model Compatibility | Needs accessories |
| **CuBox** | Small and powerful | Needs USB-alert |
| **Ethernet Relay Cards** | Simple and contains all parts needed.<br>No work needed. | Lack other extensions ports<br>Cannot easily be extended |

### 4.2.2 Alerting System Controller Cloud

Omnitor does not want to run the cloud server locally, but rather expects to utilize a popular cloud service provider. Omnitor currently uses both Salesforce[64] and Amazon Web Service (AWS) [45], hence they already have a lot of experience with cloud service providers. Amazon's AWS service was recommended to me for a dynamic project, such as the proposed product. AWS provides a flexible, cost-effective, scalable, and easy-to-use cloud-computing platform. There already exist many services that utilize AWS, this adds compatibility and crowd support of the system [46].

The services that will be used to realize the ASC cloud are EC2, VPC, CloudWatch, S3, and RDS. EC2 will run virtual instances of the type t1.micro, one of these instances is used for the ASC-Server, and another for the administration of the ASC-Server (called the ASC-Admin). CloudWatch is used to store log files from the ASC-Server. Larger amounts of information that should be shared between the ASC-Server and the ASC-Devices are stored in S3. RDS will be of the MySQL type and will be used by both ASC-Server and ASC-Admin to get/set the status of the ASC-Devices. Figure 4-5 shows the complete ASC-Cloud (both inside the VPC and outside).

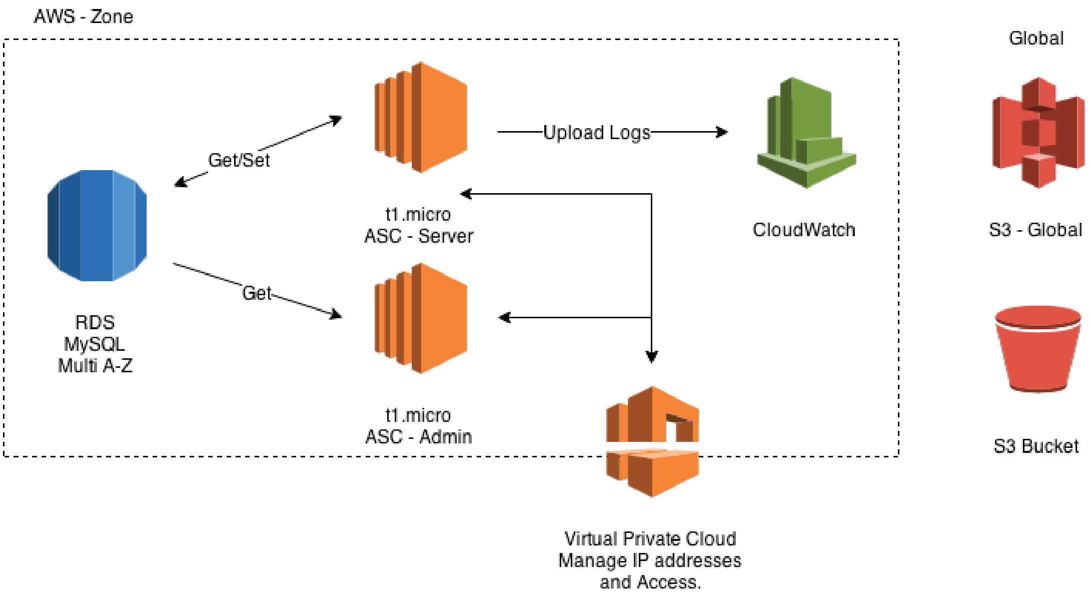


**Figure 4-5: Shows the different amazon services and their relationship to each other.**

For the entire ASC system to work there must exist a communication protocol between the three entities: ASC-Server, ASC-Device, and Omnitor's SIP server. It is important to distinguish between the needs of the ASC-Server to ASC-Device communication channel and the SIP server to ASC-Server communication channel. For the sake of simplicity, the ASC-Server to ASC-Device channel will be called the closed protocol channel, while the SIP-Server to ASC-Server channel will be called the open protocol channel. These names were chose since the closed protocols are hidden from the public protocols, while the open protocol should be opened for generic services (not only SIP).

The closed protocol channel gets its name from the fact that communication between the ASC-Server and ASC-Device should be hidden, un-modifiable, and secure; in other words this channel is closed to the public (more specifically all other parties than the two communicating parties). The reason to ensure that this is a closed communication channel is that no one should be able to spoof a message and activate the user's alerting system without permission first passing through the ASC-Cloud. To ensure this security, standard web security (SSL/TLS) is applied to this channel.

While the closed protocol channel is encrypted and hidden from the public, the open protocol channel should be usable by anyone, as not only Omnitor's SIP server should be able to use the ASC-Server. For example, other services should be able to connect to the ASC-Server in order to deliver alerts. This will facilitate future development of new services that can help to fill the gap between alerting systems and modern Internet of Things (IoT) products. Moreover, there is a need for some security measures to protect even this channel in order to prevent spoofing of messages that are sent from a malicious server (for example, a server that pretends to be Omnitor's SIP server).

To provide security for the open protocol asymmetric key cryptography could be used so that messages from the sender are signed, enabling the receiver to prove that a message is indeed from the correct source. This method would require some extra handling of the certificates and public keys that are to be distributed to third parties. Implementing the security of the open protocol should be in-line with which existing key cryptography Omnitor uses, but is outside the scope of this thesis project. Luckily, the security of open protocols can be overlooked as they messages from the SIP-Server to the ASC-Cloud are carried over Omnitor's network; hence, we can assume that appropriate security is provided for this traffic. In the future, security must be added to the open protocol to avoid unauthorized activations of a user's alerting system.

### 4.2.3 Protocol and Message Format

A protocol and message format for the open and closed channels needs to be selected. The usage of the protocol and format will help to detect errors during parsing when the receiver receives a message. The messages sent through both channels have to be generic and easy to alter when new features are added in the future. The protocol and format on both the open as well as the closed channel should be the same in order to reduce the development time and to lower the complexity of the system. The open channel should make it as easy as possible for sources to send messages, making the implementation on SIP-Server and third party platforms simple. A minimal implementation of the open communication channel should reduce later compatibility problems with software, as third parties should be able to write their own software with ease. The closed channel messages may also needed to be extended over time because of updates of both the ASC-Device and ASC-Server (due to scripts and software developed by third parties). A solution would be to release an API, but the specification of such an API is avoided in the context of this thesis in order to keep the system simple.

The most obvious choice for a protocol and message format would be to use SIP, because the existing Omnitor SIP server already speaks SIP. Building upon the established SIP infrastructure, can reduce development time – especially as existing software can be adapted to act as proxy/server/… etc. The use of SIP would help with firewall/NAT problems and communication would occurs in the same manner as for the TC-Device. In addition, this would enable the same communication protocol to be used throughout the whole system. Third parties not speaking SIP could be provided with a gateway that converts their outgoing messages (or requests) to SIP on their way to the ASC-Server. This would create a very open environment where many third parties could join in developing a rich market of services (and devices).

There are some problems with using SIP as the protocol. The new product provided by Omnitor is expected to be a complete system and should be able to be transferred to another company or organization without previous SIP knowledge. A company with no established SIP infrastructure or any previous SIP knowledge would have some problems when setting up their ASC-Server and ASC-Devices. Companies that delivers smartphones applications, web-based applications, and other simpler systems may prefer a simpler protocol to decrease the time that they need to spend figuring out how to add a SIP header.

When SIP is used as the protocol, then the commands that the ASC-Server sends to the ASC-Device have to be included inside a SIP message. This can be done using the SIP message extension defined in RFC 3428 [65]. This SIP message extension adds a MESSAGE method to SIP, and allows the inclusion of a message in the body of a SIP message. Figure 4-6 shows an example of what this use of the SIP message extension packet might look like if it were to be used for the ASC-System. In this figure, we can see that the SIP message is both a message and the reply message. In this example the ASC-Server tells the ASC-Device to keep a relay 1 closed for 100 milliseconds and then open it for 200 milliseconds and that this should be done 15 times.

```
MESSAGE sip:user2@domain.com SIP/2.0
Via: SIP/2.0/TCP user1pc.domain.com
Max-Forwards: 70
From: sip:user1@domain.com;tag=49583
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 18

action=close relay 1
on_time=100ms
off_time=200ms
rounds=15
```

**Figure 4-6: An example of how SIP could be used when sending messages between ASC-Server and ASC-Device. This figure was inspired by Message F1 (figure 1) in [65]**

In a cloud-connected network architectures, without an established SIP infrastructure, the only advantage of using SIP is to lower the development effort if existing software can be used. This reduction in development time is lost when the code that converts a signal from a third party to SIP has to be implemented. Thus, the advantage gained by using SIP for this particular case has to be weighed against the simplicity of using plain TCP. For this project, it was decided to use TCP sockets.

SSL/TLS may be used to provide security for a TCP connection. The TCP connection between the ASC-Device and the ASC-Server is established once and then maintained for an extended period. To prolong the lifetime of a TCP connection the KEEP_ALIVE flag can be set, causing TCP to periodically send a small keep-alive message. A normal home router implementing NAT can choose to close down a connection that, even if there is very little traffic on the network, if it cannot determined if a connection is closed or not. The behavior for a prolonged TCP connection through a NAT is defined in RFC 5382 [66].

Using these settings, we have secure communication via which we can create new sessions, alter a current session, and detect closed sessions (KEEP_ALIVE gets a timeout), all through TCP. This renders the use of SIP for this single function unnecessary. With either a TCP socket stream or SIP packets, the information needs to be structured. Figure 4-6 shows how the data is structured as a field and a value separated by rows and equal signs. Two common text-notation formats are JavaScript Object Notation (JSON) (as used in [40]) and the EXtensible Markup Language (XML) (as mentioned in [48).

Both JSON and XML have libraries for several different platforms; this helps parsing and lowers the amount of time spent on the implementation. There have been studies that compare the efficiency of the two communication schemes [67], but for this case the flexibility and simplicity are the most important factors. W3schools[*] claims that JSON is an easier-to-use alternative to XML, making it more readable to humans and easier to parse. The intuitive nature of JSON makes is self-describing and more fitting for software development, thus a perfect fit for this project.

## 4.3 Assembling the Prototype

The design of the prototype was decided to be a simple final product. Some features of the prototype device are essential, as it must have a feedback system, an Internet connection, and relays. The

[*] http://www.w3schools.com/json/

feedback system could be as simple as some diodes that indicated the most basic information. Internet connectivity is provided either by a Wi-Fi adapter or by an Ethernet Interface. A generic relay module board provides four relays. The device must have as few as possible ports exposed to users.

Figure 4-7 shows the assembled prototype ASC-Device from the front. Figure 4-8 shows the assembled prototype from the rear. This prototype has three diodes marked “power”, “WAN”, and “eC”, in front indicating whether the device has electrical power, Internet connectivity, and whether the device is connected to the ASC-System. The backside of the ASC-Device has a cable to connect it to an USB power source, an RJ-45 Ethernet jack, and connectors for the four relays. As more than one prototype was made, variations in design and color exists, but the basic features are always the same.


**Figure 4-7: Shows the ASC-Device prototype as seen from the front.**


**Figure 4-8: The ASC-Device prototype as it from the behind**

A Raspberry Pi directly controls the diodes and relays. This is possible as the diodes have built in resistors and the relays are already on a board with the necessary components. The Ethernet is extended to the edge of the ASC-Device's box by using a short Ethernet cable and a female-to-female RJ-45 adapter. The reason for the extension cable is to enable the use of the USB devices to be connected *without* exposing the USB ports to the user. This configuration can be seen in Figure 4-9.

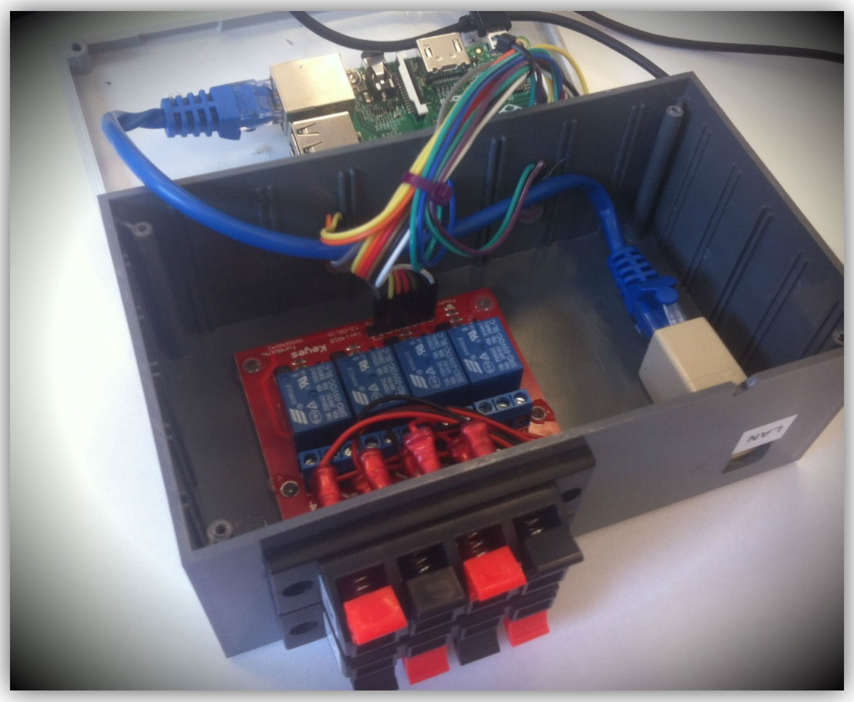


**Figure 4-9: Inside ASC-Device prototype as seen from above**

## 4.4 Software Development

The ASC-Device, ASC-Server, and ASC-Admin use a custom protocol to communicate. The software to implement these three entities was developed in Java for the ASC-Device and ASC-Server, while Hypertext Preprocessor (PHP) was used for the ASC-Admin. Java was chosen because of its multiplatform properties and due to the existing expertise for this language within Omnitor AB. PHP was used for the administrative interface as it facilitates integration with HyperText Markup Language (HTML) and it is well suited for a web-based administrative interface.

The ASC-Device was implemented as client software using Java together with some bash-scripts (Linux terminal script) to control the device. This client software identifies the box based upon the unique Raspberry Pi ID and this ID is used when connecting with the server. This ID enables the server to identify each ASC-Device. When the operating system of the ASC-Device boots it starts the client software. A script tests for Internet connectivity against a known server and if there is connectivity, then it turns on the Internet connectivity diode. The client software creates log entries in log files. These log files can be requested by the ASC-Server. The script that automatically starts the client software will restart the client software in case it terminates or crashes (The amount of crashes is outside the scope of this project as it would take an extended amount of work to figure it out).

The left part of Figure 4-10 shows a flow-chart describing a simple version of the client software used in the ASC-Device prototype. Following the chart from the top to bottom, the first phase is where the client software first starts. Next, it reads a configuration file. This configuration file contains the server's domain-name and the TCP port number to which the client software will connect to the ASC-Server. This configuration file also contains a field for the unique ASC-Device ID. This ID is inserted by a startup bash-script. The current software version number and the information necessary to perform a software update are also stored in this configuration file. After successfully reading the configuration file, a separate thread is launched. This thread checks for software updates and downloads an update if necessary. Next, a connection to the server is established. If this connection establishment is unsuccessful, then the client will try again, using a back off algorithm. This back off algorithm causes the software to wait longer before each successive connection attempt. This is done in order to limit the load on the server. After a connection is successfully established, the ASC-Device identifies itself to the server using its unique ID and waits for an OK message from the server. If an OK message is received, then the ASC-Device turns on the "eC" diode and can start to receive commands from the ASC-Server. These commands are left unspecified in this thesis as these commands depend upon the specific software release that is running in the client. From a design point of view, these commands should be very general and it should be easy to add new commands. The initial commands that have been implemented for the prototype can turn one of the four relays on and off according to a specified pattern.

The ASC-Server is a multithreaded TCP-server that listens on two ports. One port is used for the ASC-Device to ASC-Server communication and the second port is used for communication from third parties to the ASC-Server. This server uploads its logs files to Cloudwatch where they are stored in an RDS MySQL database. A Dynamic Name Service (DNS) lookup of the server's hostname is used to learn the IP address of the server. The ASC-Server is started manually. However, manually starting the server should be rare, as the server is designed to run for an extended period of time.

The right part of Figure 4-10 shows a flow-chart describing a very simple version of the server software used in the ASC-Server. Similar to the client software, the server software starts by reading configuration information from a configuration file. This configuration file tells the server software which two ports and IP addresses the sockets should be bound to. A separate thread listens for incoming connections on each of these sockets. Messages from these sockets are placed into two separate queues. Messages are extracted from these queues and processed in the main thread. The reasons why two queues are used are for efficiency and priority control. Different actions are taken depending on which socket the message was received on. Because the JSON format is used, it is easy to distinguish one message from another, even though it is not very efficient. By using the JSON format, the messages have to be parsed twice, once from the TCP stream and then into a JSON object representation.

If a message is received from an ASC-Device, then this message is examined to see if it contains an ID. If it does not contain an ID, then an error message is delivered to the sender and the message is discarded. However, if the message contains an ID, then a database is updated to record the fact that this ASC-Device is connected. After the database is successfully updated, an "OK" message is sent back to the ASC-Device. Now the next message in the queue is examined.

A message from a third party application should contain a list of ASC-Device IDs. This is the list of ASC-Devices that should be activated (to generate an alert for the user) when requested by the third party application. For each ID in the list, the server first checks if the specified ASC-Device is online, if not - it will continue to the next ID in the list. If the indicated ASC-Device is online, then the server performs the actions specified with the list of IDs. An action is identified by a numerical value that represents a specific task that the ASC-Device should perform. Each action value is translated into a command that is sent to the indicated client. This translation is done via looking up a database entry (potentially based upon both the ID and the numeric action value). Using a database gives fine grained control over what commands can be initiated. Additionally, different third parties can be restricted to different sets of commands that they can request the ASC-Device to execute. Moreover, some ASC-Devices might not be able to perform a given command. If the action value that is requested for a specific ASC-Device is allowed *and* can be performed, then this action value is translated into a command and sent to the specific client. After this, the action for the next ID is processed. This continues until all of the IDs in the list have been processed. The next message in the queue can now be processed. It should be noted that the third party interface could be extended to support SIP messages, though this is left for future work - as the prototype is intentionally kept simple.

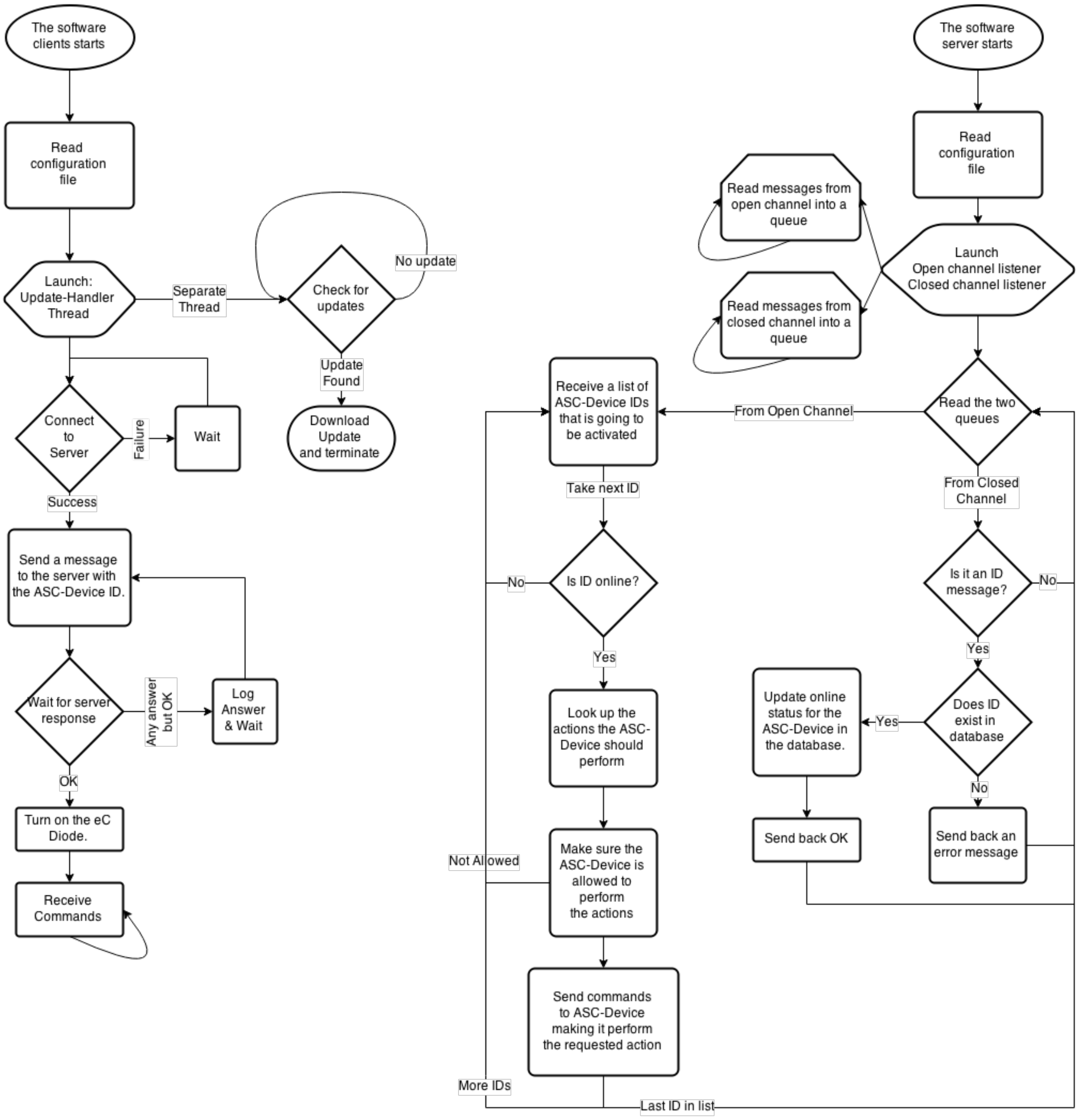


**Figure 4-10:** **A basic flow-chart of the two main programs in the ASC-System: the server and client software. The client is on the left, while the server software is on the right.**

The administrative interface enables the administrator to read logs files and other data from the database. Via the administrative interface, the administrator can send test messages and commands to specific ASC-Devices. Overall, this administrative interface provides the administrator with an easy to use interface that they can use to control the system. This administrative interface can be accessed through a web-browser, via scripts, or via programs.

## 4.5 Protocol Messages

The messages that can be sent via the ASC-System are defined in JSON format. This format was selected, as it is easy to extend in the future. One set of messages is defined for the open channel and another set for the closed channel. The open channel messages are fixed and designed to be simple with as few options as possible. This is done to make the message structure easy for third parties to follow, thus lowering the chance failure and minimizing problems of interoperability. The closed channel messages have to be flexible and extendable with control fields and are used for communication in both directions (both ASC-Device to ASC-Server and ASC-Server to ASC-Device). As the closed channel is used inside the ASC-System, there is little danger with making the structure more complex and contain several options. Note that the messages used to communicate with a particular ASC-Device have to match the software installed in the specific ASC-Device.

Figure 4-11 shows the structure of the messages sent on the open channel, these messages are always sent from the third parties (or Omnitor's SIP server). The first field is the **action-id**. This action-id indicates a predefined action that is stored inside the ASC-Server's database; these actions define what the closed channel messages should be. The second field **box-ids** are a list of all ASC-Device that should be affected by this action. Both of these fields must always be present and contain a value, otherwise the message will be dropped.

```
{
        "action-id":   "actionnumber",
        "box-ids":     ["id1,"id2"]
}
```

**Figure 4-11: The JSON structure of an open channel message. The messages are exchanged between third parties and the ASC-Server. The full structure is required for every message.**

Figure 4-12 shows the structure of the messages that are sent over the closed channel, these messages can be sent in both directions. The message contains six fields: **gpio**, **radio, action**, **settings**, **result**, **join**, and **status**. The **gpio** (standing for general purpose I/O/) field is used to define which programmable pins that should be used; the **action** field defines how the pin should be used. The **action** field can either be activate, toggle, on, or off. Activate let's the defined pins toggle on and off at intervals for a specified number of rounds (i.e., iterations), these properties can be defined using the **settings**. Toggle changes the status of a pin to the opposite of its status (on becomes off, off becomes on). On/Off sets the status of a pin to on/off. The **result** field is used to give responses back to the other party, for example telling the ASC-Client whether a join message was successful or not.. By making the ASC-Client or ASC-Server send a log message to the other party, this will force the message to be logged. Another field used when joining is **join**; this field contains the ASC-Device ID and is always sent from ASC-Device to the ASC-Server. The **status** field is used to either request the status of an ASC-Device or send the ASC-Device's status to the ASC-Server as a reply. For example, the system status could indicate which local IP address the ASC-Device has.

```
{
        "gpio":          [1,2,3,56,33…etc],
        "radio":         [1,2,3,4,5]
        "action":        "activate"/"toggle"/"on"/"off",
        "settings":      [interval, rounds],
        "result":        "feedback string",
        "join":          "Device ID",
        "status":        ["127.0.0.1"]
}
```


**Figure 4-12: The message that can be sent through the closed channel. Messages goes between ASC-Device and ASC-Server. Only parts of the structure is required for a full message.**

The **radio** field works in much the same manner as the **gpio** field, but instead of activating a programmable pin, it activates a predefined radio signal. This radio signal can be used to control an electricity outlet, such as those provided by NEXA [7] or other devices supported by the hardware. The hardware in the protoype is a Telldus Technologies* Tellstick. More complete instructions on how to use the Tellstick with the Raspberry Pi can be found in a report by Anderas Hallberg and Oskar Lundh in their thesis about home automation [40].

* http://www.telldus.se/

# 5 Analysis

After the development of the prototype, several copies of the prototype were constructed. Each of these copies had a slightly altered design, but the basic functionality was the same. These prototypes were loaned to test users who were supposed to test the system for two week, after which they were presented with an evaluation form.

In this chapter, we present the data learned from these forms. This data was then analyzed and the results will be discussed in this chapter. A time line for introducing future features will also be presented. This time line is based on the results from the analysis of the evaluation forms.

## 5.1 Major results

There were seven completed (web) forms. Some of these forms were filled in by several persons working together, rather than just a single person. These participants were an ordinator, installer, or users. The ordinator is the person that handles the purchase and ordering of aids for people with disabilities, the installer actually installs the ordered aids for the user. The user is the person that actually uses the aid itself. The reason why all these entities were asked to evaluate the product is that even if a product would fit a specific user, the user will not be able to receive the product if the ordinator cannot prescribe the aid. If a product is too much work for an installer to install, then the product might be abandoned in favor for another easier to install products.

The questions were grouped into three categories: the current solution, future solutions, and other. The current solution questions deal with the participants' view of the prototype and the features that they found useful. The future solution questions deals with what the participants want to see next in the ASC-System. The “other” questions were present if there is something else the participants would like to provide as feedback. All of the answers will be evaluated based on the questions and then presented in their category.

The (web) form allows the user to pick one or more answers to some question. The reason why the user could choose more than one answer is that when using a limited number of participants it would be hard to draw conclusions if all of them picked different alternatives. Allowing the participants to pick several alternatives makes it easier to find common alternatives that many participants picked and thus ease the job of prioritizing the answers (when the answers overlap).

Appendix A shows the complete survey (web form) as it was presented to the test users. As the survey was conducted in Sweden, the survey form is written in Swedish. The persons that completed the form have knowledge about the current situations for hearing-impaired and deaf people, giving them insight into the questions presented in the survey. Because of the specification and language of the survey, a short presentation of the questions and their alternatives is given in the following paragraphs.

The first question enables the test users to state which role he/she has (i.e., an ordinator, installer, user, or “do not know”). The following question asks the user to rate the prototype (as representing a potential product) from one to ten where one is a lesser product and ten is a superior product. This second questions also allows the test user to give a comment on their rating. The third questions asks which important properties the new solution should have. The prepared answers to this question were Wi-Fi, Ethernet, platform independent, relays, radio through Tellstick, security, simplicity, and different colors (different colors for the box that the ASC-Device is in). The test user may check none, all, or just some. Questions 4 and 5 differ from the previous questions as both of them concern future work rather than an evaluation of the current solution.

The questions 4 and 5 concern what service and devices should be added/supported. Both questions have an open ending where the test user can add their own service or device they want the system to use. Again, the user may check none, all, or some of the pre-provided answers. The sixth and final question is an open question that enables the test user to give free form input on anything connected to the potential new product.

If an open question was answered by a test user, then these answers were grouped together with similar answers. The grouping was done by looking at keywords and clustering similar opinions. The groups consist of the established alternatives or a newly created group for a specific free-form answer. If the answer to the open questions was more of a comment on their answers, rather that actually bringing up something new, this answer was noted but nothing extra was added to the concluding diagrams. In Appendix B the complete answers is listed, a one represent **True** (the user checked that alternative) the number zero represent **False** (the user did not check that alternative).

We can see that only one user answered although in some cases were a "team" answered, but not everyone in the team had the same position, some were users, some were testers, and some were ordinators. In such a case, the team picked the most fitting answer. As the test users grouped themselves into teams, the test users types are unreliable.

### 5.1.1 Current Solution

Questions 2 and 3 dealt with the current solution and the overall impression of the product. Question 2 asks the user to rate the product from one to ten. A rating of one means the product is a bad product and something the user would never use / buy for themselves. A rating of ten means the product is an excellent product that the user would recommend to others. In addition, the user could place a simple comment on their rating; if there was something important they wanted to add.

The users seem to agree that the product is a rather good product. The different test users ratings are presented in Figure 5-1. From the figure we can see that the lowest rating is 5 and highest is 9, while the other scores are divided between those. The mean value for the ratings is approx. 6.86 and the median is 7.0. Even though the product is overall likeable among the test users, there are reasons why a higher score was not given.

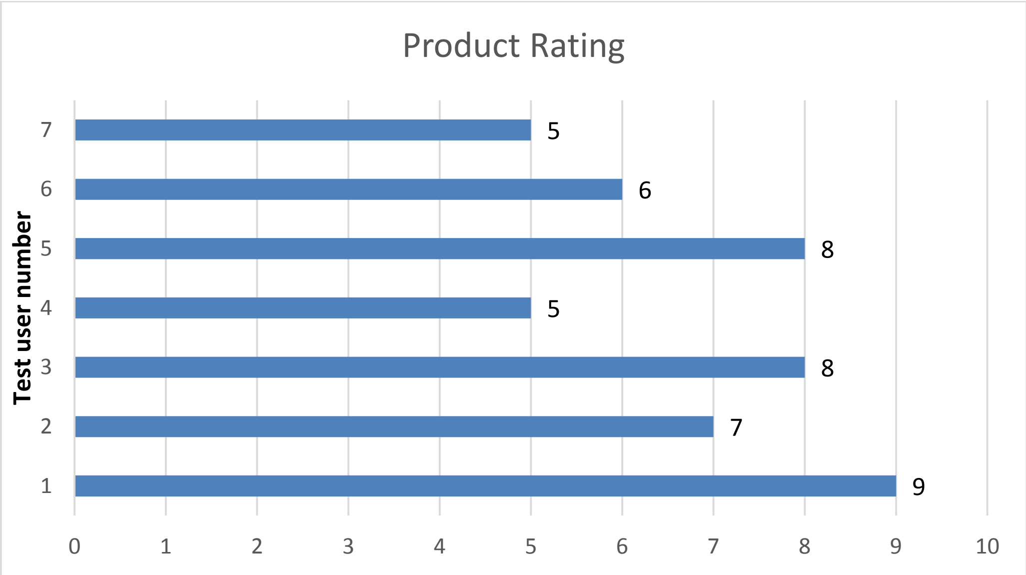


**Figure 5-1: Test users rating of the prototype as a product. One is bad and ten is good. The mean value is 6.86 and the median is 7.0.**

Looking at the comments given for these ratings, we can divide them into groups: missing service support, missing device support, bad design, and nothing new. Missing service/device support is a complaint concerning the fact that some features are missing (either a specific device or service was unsupported). Bad design is a group of complains about the design of the box. Nothing new simply reflects that the test user does not think the new product makes any change compared to the existing products.

The first comment (going from the top of those listed in Appendix B) was that the test user would like a flashlight to be attached to the ASC-Client together with additional options; furthermore, the ASC-Device looked messy when many cables are connected to it. The comment falls into the bad design category group, as if even if the comment concerns a missing feature, it comments on the ASC-Device itself, not device support. Following this first comment there was a short comment that states that the user did not notice any difference compared to the old USB-Alert system. This response falls into the "nothing new" group. The third comment is a perfect fit to the both missing device and missing service support, as it stated that a higher grade would have been given if there were more support. The last comment focused on problems with missing connection services to other SIP-companies, thus this comment falls into the missing service category. Table 5-1 shows the results of the mapping these comments into groups.

**Table 5-1: Mapping of the rating comments into categories. The left column is the category, while the right column shows how many times this type of response was given (mapped).**

| ***Category*** | **Times mapped** |
|---|---|
| *Missing Device* | 1 |
| *Missing Service* | 2 |
| *Bad Design* | 1 |
| *Nothing New* | 1 |

Figure 5-2 shows the result, concerning the WiFi and platform independent features. This shows that these features were those the test users liked the most. Another feature that got a high rating is the LAN feature. This is very interesting as using WiFi / LAN and being platform independent are some of the features that actually set this solution apart from the solutions presented by traditional alerting system provider companies. The lowest rating was given to multiple colors and radio, neither of which really affected the test users. The ASC-Device was simple to hide away after installation and the users used their existing alerting system (such as a Bellman). This might be the reason why those two features got such a low rating. Another interesting aspect was the relatively low rating of the four relays and that security that this solution provided. Both the relays and security are central parts of what makes this system unique compared to other similar systems, such as those made for hearing-impaired and deaf individuals. It is possible that the reasons for the relatively low ratings of these two features are that the user found them unimportant or very transparent and hence took them for granted.

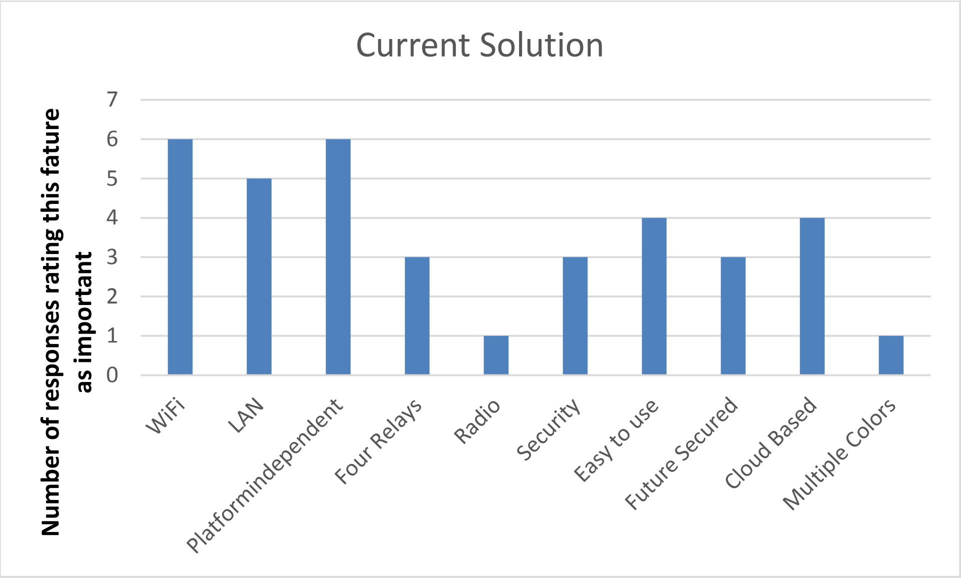


**Figure 5-2:** **The collected result that the tests users felt were important in the current solution, the solution that they got to see and tested.**

### 5.1.2 Potential Future Solutions

Questions 4 and 5 deal with potential future solutions and what the users wants to see in a future development of the ASC-System. The process of rating of future solutions follows the same structure as the rating processes of the current solution. The test users where presented with different service alternatives: Skype (the system triggers on incoming Skype calls), SOS-Alarm (the system triggers on alarms), E-Mail (the system triggers on incoming e-mails), Lamp-Settings (allows the user to decide in what manner the relays will close), other SIP networks (add other SIP systems to the ASC-System), Medicine Reminder (the system triggers when it is time for the user to take a dose of their medicine), and other. As in the current solution, these questions included open questions, where the user could add their proposal for a future service.

Figure 5-3 shows the result of the answers to question 4. These results indicate that Skype and Other SIP networks were the most important features that should be added in the future. This is understandable as there is a greater chance that the user can add the ASC-System to their favorite video conversation software. The least picked alternative was the medicine-reminder, as this target group had no particular problem with medicine or their memory, making this service unnecessary for the test group. The comments provided by the users reached a rating of four, which is a very positive sign of the test users' interests in the product.

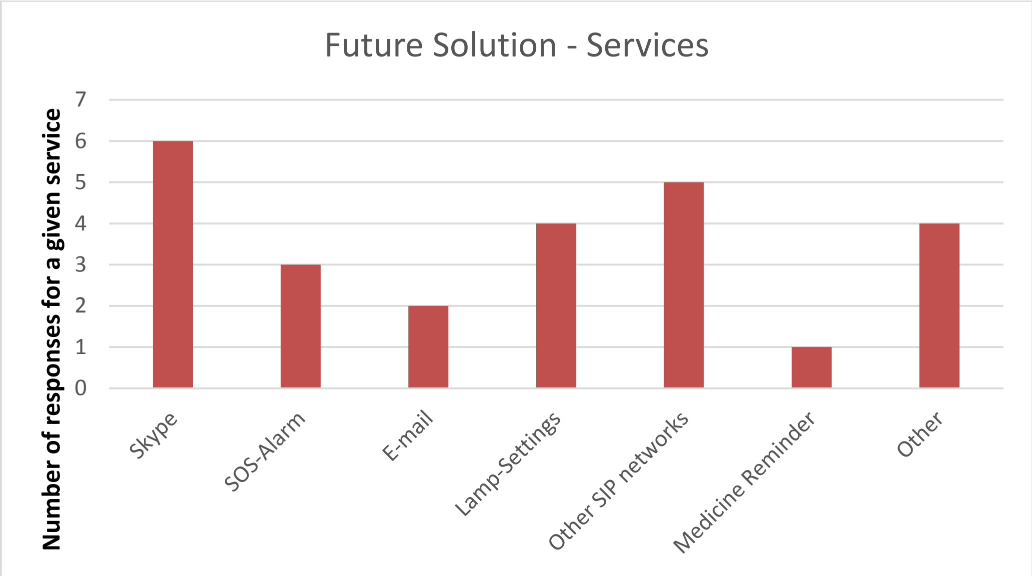


**Figure 5-3:** **The collected result of what the tests users liked about the future solution. This figure corresponds to question 4 about new services that can be added.**

The tests users own comments were straightforward and can be divided into three groups: FaceTime (Apple's video conversation service), totalsvararen (an answering machine for the deaf), and VMA (important message to civilians in Sweden). The first comments focus on totalsvararen, the second VMA and FaceTime. The last two comments where both to add FaceTime to the set of supported services. The reason why FaceTime was a popular is the same reason why Skype was a popular alternative. As FaceTime was request by so many individually, it is important to consider it as a future service. Table 5-1 shows these mappings.

**Table 5-2:** **Mapping of the future solution´s *part II* into categories. The left column is category the right column is how many times if was mapped**

| ***Category*** | **Times mapped** |
|---|---|
| *FaceTime* | 3 |
| *Totalsvararen* | 1 |
| *VMA* | 1 |

Figure 5-4 shows what kind of smart devices the user wants the ASC-System to support in the future. The two most highly rated devices where a smart door phone and fire alarm. At a first glance these results looks surprising, but remembering that the other products provided by Bellman, Sonic Alert, and others do not support a door phone nor fire alarm (see Assistech for a compatibility list [26]) these responses make sense. The other category received the lowest rating, although many of the test users left a comment. These comments were generally simply comments on their answers and not proposals for smart devices that could be supported by the ASC-System. Overall, there was a high rating for all the different answers apart from the "other" category. Interestingly smart light bulbs and clocks had the lowest scores. Remember that even if current alerting system companies (such as Bellman or Sonic Alert) can provide support for clocks and flashing lights, neither of these devices are smart. On the contrary, the clocks and flashing lights provided by those companies are just extensions of their current systems and utilize the same technology.

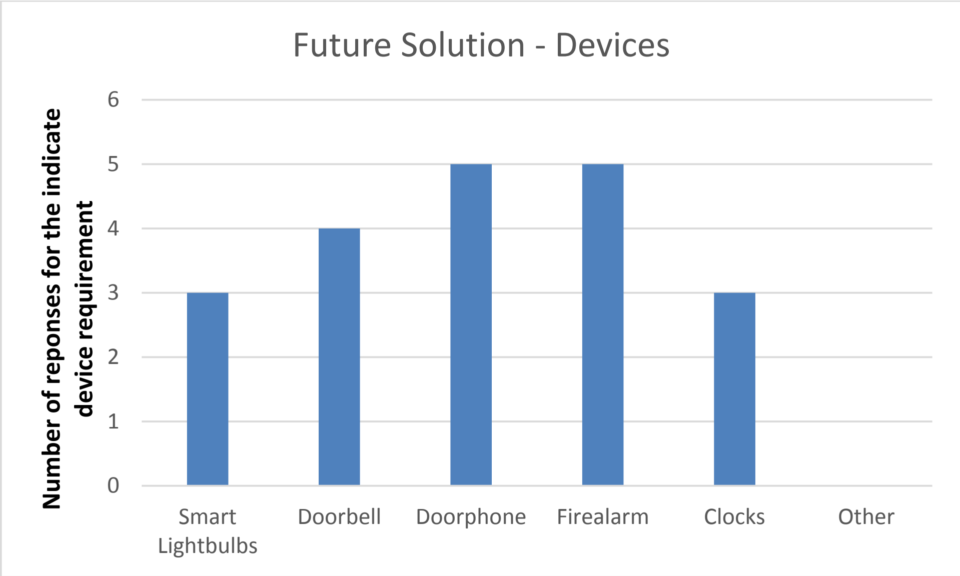


**Figure 5-4:** **The collected result of what the tests users liked about the future solution. This figure corresponds to question 5 about new devices that can be added.**

The three comments left on future devices either states the lack of interest the user has in the new products or simply explain their answer. Those who had no interests in connection with new devices pointed towards their current alerting system they have gotten used to and that they would continue to use. Through these comments, the importance of the compatibility with current devices is high-lighted. This makes the ASC-System requirement about compatibility even more important and shows that this compatibility should not be left out of future products.

### 5.1.3 Other

The last question was an open question where other information about the prototype could be given. This field is special, as it was not tied to any other questions. These answers were very general and were not coded and categorized into groups, due to the diversity of responses. Nevertheless, some evaluation of these responses must be done.

Luckily, the answers often regard changes in design and proposals for future cost, rather than comments on the current solution. The first comment goes into depth about the pricing of the device, something that will be rather useful when converting the prototype into a real product. Of course pricing is something that must be addressed, but is outside the scope of this project. The other three comments ask for design changes, such as having a more logical pattern for how the diodes are used. Another common complaint was the earlier mentioned lack of support from other SIP-companies and popular video conversations tools such as Skype and FaceTime. Once again, the need for Skype and FaceTime support is apparent. Another complaint concerning design choices concerned how to connect the Ethernet cable, hence this should be improved in a finished product as some users had problem removing the cable once it is inserted (due to the cable getting stuck in the Ethernet port on some design on the ASC-Client).

One comment mentioned that the use of a cloud service is a bad idea and the user would rather see the device sniff the incoming traffic. This is not possible with the system created in this project, and is probably not applicable in real life. Some SIP-Services utilize encrypted SIP messages, hence making traffic sniffing much harder. Creating a device that sniffs for SIP messages would also limit the solution's otherwise broad application, as it would only be able to handle SIP. An alternative solution could be to forward SIP messages from the TC-device to the ASC-Device and let it inspect those packets. That way, many problems could be overcome, but this approach is outside the scope of this project.

## 5.2 Future Timeline

As some of the results involve gathering information about what the test users wanted to be the next step for Omnitor's new product, creating a time line of when new features should be added is a good idea. The timeline will not give a complete picture of all future work, but rather gives an overview of when features could be added. (Note that this is similar to the ordering of when to introduce features as described in the master's thesis by Yi Fu and Ruimin Li [68].)

The timeline is divided between devices and services. Figure 5-5 shows the timeline given the services and devices that were prioritized by the test users. In this figure, services are colored green, while devices are colored blue. To the very left of the figure is time zero, and with time we move right along the x-axis. The y-axis simply spreads out the different features (particularly keeping the devices and services separated). The different services and devices were taken from the alternatives considered in the survey. The timeline was created by using the responses to the survey questions.

Looking at Figure 5-5 it is obvious that adding support for Skype / FaceTime is the high priority task. Many of the test users added it as a service they wanted to see in the future, and many of the open questions had comments about it. After a Skype/FaceTime service has been added to the system, and then the next task is to make it so that users can connect to other SIP-companies and their services while making use of the alerting system. This would enable the user to use another SIP-address then those provided by Omnitor. Following these two enhancements, the next important service is the Lamp Settings, SOS Alarm, E-Mail, and the lowest priority is the Medicine reminder server. These other services where not even close to the same importance as Skype/FaceTime and connection to other SIP companies.

When looking at the device part of Figure 5-5 (the blue boxes) we can see that no device was as important as the Skype/FaceTime service. The most important devices that users wanted to add to the ASC-System were fire alarms and door-phone. More importantly as a product differentiator, these types of devices are missing from products offered by competitors, such as Bellman and SonicAlert. Following these devices comes the doorbell (provided by most alerting system companies), smart watches, and smart light bulbs. It is important to remember that although the other devices were not as highly prioritized as the Skype/FaceTime service; these services still have a higher priority than the e-mail/medicine reminder.

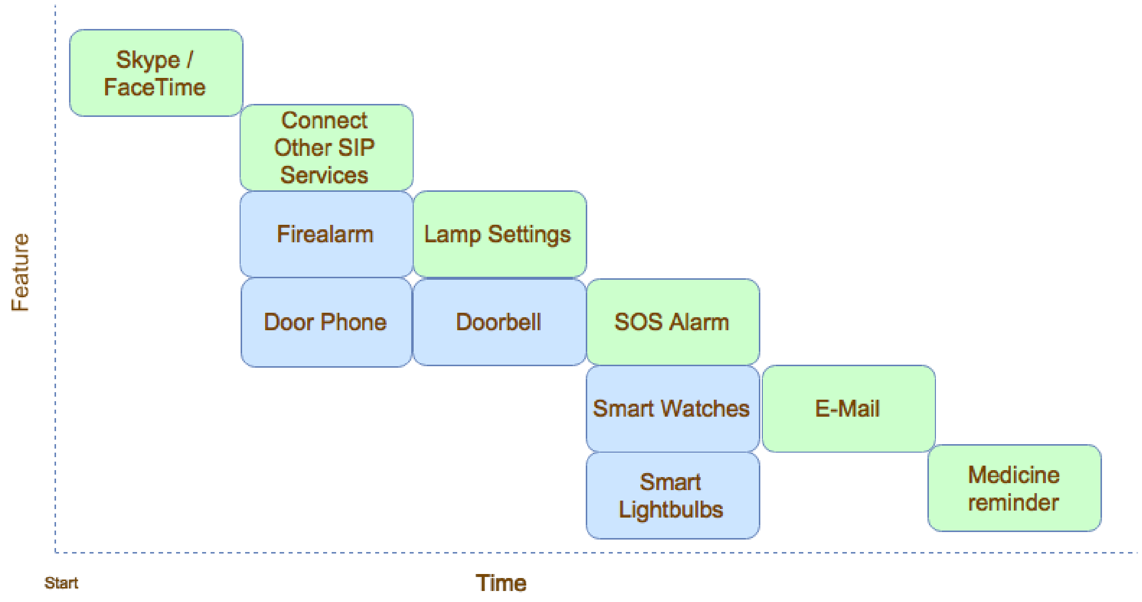


**Figure 5-5:** **The timeline for future releases of features based on user input. The two colors green and blue is used to differ between services and devices, green is services and blue is devices.**

## 5.3 Reliability Analysis

The reliability of the analysis depends on the reliability data. Following the same argumentation as in Chapter 1, there is nothing to gain for any parties by falsifying the data. The ordinators, installers, and users all want to have the best possible product, as they are the one who are going to use it. For this reason, I believe that they each gave honest feedback. Omnitor wants to know if the prototype should be further developed into a product, thus they also want honest and useful feedback. Omnitor would not gain anything by altering the data, as this would not change the opinion of ordinators, installers, and users. From this, we assume the survey responses were completed truthful.

The collected data set is small which can be a problem with regard to reliability. Although the data set is small, the data set was gathered from geographically different places, which is important for this kind of system, as there are different rules in different counties in Sweden when it comes to distributing aids to people with disability. For example: a different agreement on what kind of brands of aid should be used is not always the same in all counties. By gathering data from many of the different counties, a more general view was gained. This also addressed the case where a product suits one geographical region, but not another. This is not due to a technical reason, but rather restrictive local procurement policies.

Another factor that adds to the reliability is who was giving the feedback. From the survey results (Appendix B) we can see that there are 3 ordinators, 1 user, and 3 installers. This means that only one user actually gave feedback about the system, hence the survey may give a faulty picture of reality. However, some of these answers where answered in groups, where these groups might include a user. The survey erroneously failed to allow multiple types of test users to be indicated in the response, this mistake should be avoided in any future survey. All of the test users were either professionals or had been using alerting devices for a long time, which actually adds to the reliability of the data.

## 5.4 Validity Analysis

In contrast to the reliability of the analysis, validity does not depend on the data, but rather depends upon how the data were interpreted. Again, there is nothing to gain by deliberately misinterpreting the data to draw specific conclusions. Hence, I believe that the analysis gives a view of the real world, as it would be best for all parties involved. The analysis has been done with logical reason by using the variables provided by the data. There might be a difference in the results of the analysis if the same analysis was redone on larger dataset, although the reasoning would be the same. Even if the result might change with the analysis of a larger investigation, the same reasoning should be applied.

## 5.5 Discussion

Looking back at the goals of this thesis, when evaluating whether the sub-goals were fulfilled or not we can examine the summary given in Table 5-3. This table lists each of the project's goals and how the goal was fulfilled. From this table we can see that all goals actually were fulfilled, but this does not mean that the prototype is a finished product. The prototype is primarily a proof of concept, rather than a complete product. Some additional fine-tuning has to be done to transform the prototype into a product. The most important change may not be technical, but may be necessary from a marketing perspective. Although several devices based on the prototype have been created for the evaluation and even when system address a need, it is important to remember that a prototype is not a product, as it has neither a price nor a market name. In order to put a price on the system will require pricing of the end-devices and consideration of the cost to connect third parties to the system. Working out this pricing and a business model is a first step towards going from a prototype to product. A real product name that would be attractive is a second step (clearly, "ASC-System" is not a good product name). Additionally, there is a need for an industrial design of the ASC-Device and additional user interfaces are important next steps.

**Table 5-3: Lists the goals on how each goal was fulfilled.**

| Goal | Fulfillment |
|---|---|
| • **Perform a market investigation** | Was done and result is presented as a part of the background in theses thesis. |
| • **Design an alerting system that uses a cloud-based architecture.**<br>• **Propose different system solutions that fulfill both Omnitor AB´s and user's system requirements.**<br>• **Develop a prototype based on the most appropriate solution.** | Described in Chapter 4. |
| • **Evaluate the prototype.** | Described in this chapter. |
| • **Document and present the work.** | Done throughout this report. |

# 6 Conclusions and Future work

From the analysis, there is an overall positive attitude towards the prototype, although some features should be added before releasing a product. Section 6.1 summarizes my conclusions from this project. Some things could have been done differently to produce a better result, but those changes can be made in the future. The next step for this project would be to go from a prototype to a real product; hence, a great deal of work remains to be done. Section 6.3 summarizes this suggested future work. Section 6.2 describes some of the limitations of this project. Finally, Section 6.4 describes some of the social, economic, and sustainability aspects of this thesis project.

## 6.1 Conclusions

The goals of the project, as well as the requirements for the system have been fulfilled. The evaluation produced positive results, but several things remains to be done in the future. Many things could be done differently, and some changes would make the resulting prototype (ASC-System) even better. The reason for the smooth development and successful outcome was due to three key factors: Omnitor´s support, the use of agile software development methods, and helpful county councils.

Omnitor has throughout the project given insight and help with resources to develop the prototype. Having an entity that you can brain storm with is of great help and the project would not have turned out the same without Omnitor's input. Agile software development helped tremendously, having a board with tasks, time slots, and deadlines makes the development easier to control and plan. Without agile software development, I am unsure whether the prototype would have been finished on time. The county councils provided some of the test users, with expertise in the subject, and potential future customers. These persons could give feedback that was meaningful. Without the help of the county councils, the result may have looked very different.

It might be the case that alerting systems are not the only aid to have fallen behind in development compared to the consumer market. Anyone that is going to improve, or develop new aids, should try to follow some of the techniques shown in this thesis. For example, comparing the market of these aids to the consumer market, and seeking to build a "missing link" - that fills the gaps between the newer systems and the existing aids. There is a high probability that a new product would be unsuccessful if it could not be used with the existing systems. Another important decision was to keep the product simple. The result is that the system should be usable by anyone (any user or any company). Additionally, by keeping your logic in the cloud and creating an easy to use interface to the system, the result is a simple device for the user (as the device does not have any buttons, require any settings, or require more than connecting electrical cables to terminals and plugging in an Ethernet cable). Moreover, putting the logic in the cloud allows for a simple open protocol for use by third parties. It is expect that this will facilitate the evolution of an eco-system of products rather than a closed system.

If I were to redo this project again, I would consider using the SIP protocol for my entire communication, instead of my defining my own protocol. The result could change the behavior of the system and provide the user with many opportunities for customization. Another change I would do make would be to **not** use Omnitor's SIP-Server, but rather focus on making the system compatible with popular video conversation tools (such as Skype / FaceTime). Of course, Omnitor's SIP-Server could be added later based upon its priority in the proposed timeline.

## 6.2 Limitations

When it comes to what limited this project most, Omnitor was a limitation itself. Because Omnitor AB is a company and this project was done on their behalf, some decisions and design choices were made because of their preferences and these might not actually have been the best possible decisions and choices. It is important to remember that Omnitor was not only a limitation, but also funded the project and provided it with resources such as competence, hardware, and the idea for the whole project.

The limited numbers of participants in the survey limited the results and threatened the reliability of the data. However, it is important to remember that the respondents were professionals in the area and all of the test-users were recommended by these professionals. This limitation exists because of the limited number of professionals with sufficient time to install and test this device for two weeks and give feedback afterwards. Adding the geographical spread in deployments to the task further limited the number of participants.

## 6.3 Future work

There is some future work left to do to convert the prototype into a real product. As seen in the results, there is a very positive attitude towards the product; hence, the next obvious step would be go from a prototype to a product. However, several features needs to be added or implemented to make this step.

Before exposing the system to third parties outside Omnitor´s own network, the open protocol channel needs to be appropriately secure. For example, public key cryptography could be used and a key distribution infrastructure could be added. This would result in giving each third party a private key, this private key could be used to extract session keys embedded in messages encrypted with their public key. In addition, the private key could be used for signing, to ensure the message is sent from the correct source. There exist standards and applications that already perform these tasks. These should be adopted in any future product.

A fresh redesign of the prototype has to be made before releasing a product. Creating a more appealing design of the case as well, adding user manuals, and developing a business and pricing plan are important features that must be added by Omnitor before selling a product. If Omnitor wants to sell the whole system, a separate price has to be developed.

It is important to acknowledge that this ASC-System is simply a first step towards modernizing the alerting systems for hearing-impaired and deaf users. More features and functionality (examples can be found in Section 5.2) must be added before ASC-System would be recognized as a truly revolutionary product. Creating third party services is a costly and time-consuming effort, but an important building block in the future work. Integrating the system with current IoT-technologies is important, as this will fuse the two industries: specialty alerting systems and IoT products. This fusing would give users more options to pick the best technologies available, regardless whether the user has disabilities or not.

## 6.4 Reflections

This thesis presented the development of a proof of concept of an alerting system controller that tries to fill the gap between modern IoT technologies and aging alerting system specially made for hearing-impaired and deaf people. The prototype should be further developed into a real product to aid people with disabilities. Omnitor is expected to create such a product and add it to its inventory of products, especially focusing on selling it to users of their TC-Devices.

Because of the nature of economics, as Omnitor expands their business they stimulate the economic cycle in Sweden, thus the world. As more features are added to the alerting system, the target group is widened and more devices can be sold, stimulating the economic cycle even further. As with most products that uses electricity and computing power, the use of the product will have an environmental impact. Fortunately, the Raspberry Pi does consume very little electrical power, although the components that it consists of are based upon material that has to be taken from the earth.

Although the data stored has to be protected, as it should not be available to everybody, the data is not highly secret or sensitive. Apart from the integrity of the data, there is not much of a social or ethical problem that has be considered. It is important to note that as new features are added and users gain more customizable settings, both social and ethical problems will have to be re-evaluated.

## References


[1] D. Jung and S. Jones, “Interactive Definition of Single-user Profiles for Alerting Systems,” in *Proceedings of the 22Nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, New York, NY, USA, 2010, pp. 33–40.

[2] © 2014 Rest Devices, Inc, “The Mimo Smart Baby Monitor,” *mimo*, 29-Jan-2015. [Online]. Available: http://mimobaby.com/. [Accessed: 29-Jan-2015].

[3] Postscapes, “Wi-Fi Lights: A breakdown of IP enabled and data transmitting bulbs.,” *Postscapes Tracking the Internet of Things*. [Online]. Available: http://postscapes.com/wifi-lights. [Accessed: 29-Jan-2015].

[4] Postscapes, “Wireless Home Sensor Systems,” *Postscapes Tracking the Internet of Things*. [Online]. Available: http://postscapes.com/home-wireless-sensor-systems. [Accessed: 29-Jan-2015].

[5] “Bellman & Symfon,” *Design for ears*. [Online]. Available: http://bellman.com/. [Accessed: 26-Jan-2015].

[6] “Clarity® – Smart for Seniors™,” *Phone Technology for the Hard of Hearing*. [Online]. Available: http://www.clarityproducts.com/. [Accessed: 26-Jan-2015].

[7] “NEXA Electronics.” [Online]. Available: http://www.nexa.se/. [Accessed: 26-Jan-2015].

[8] “Serene Innovations.” [Online]. Available: http://www.sereneinnovations.com/index.php?route=common/home. [Accessed: 26-Jan-2015].

[9] “SilentCall Communications,” *Deaf Alerting Systems Smoke Detectors for the Deaf Blind Alerting Devices Deaf-blind Electronics*. [Online]. Available: http://www.silent-call.com/?intro=1. [Accessed: 26-Jan-2015].

[10] “Sonic Alert Specialized Hearing Solutions,” *Amplified Alarm Clocks, Phones, Listening Devices, and Signalers*. [Online]. Available: http://www.sonicalert.com/. [Accessed: 26-Jan-2015].

[11] “Omnitor - möjliggör bättre kommunikation mellan människor,” *Omnitor*. [Online]. Available: http://www.omnitor.se/. [Accessed: 23-Jan-2015].

[12] “Årsrapport 2014: Hörselskadade och hörselvård i siffror,” Hörselskadades Riskförbund, Stockholm, Årsrapport, 2014.

[13] Washington University School of Medicine, “Timeline of Hearing Devices and Early Deaf Education,” *Deafness in Disguise*. [Online]. Available: http://beckerexhibits.wustl.edu/did/timeline/index.htm. [Accessed: 26-Jan-2015].

[14] “Total Conversation,” *ITU*. [Online]. Available: http://www.itu.int/en/ITU-T/studygroups/com16/accessibility/Pages/conversation.aspx. [Accessed: 24-Apr-2015].

[15] M. A. Hersh and M. A. Johnson, *Assistive technology for the hearing-impaired, deaf and deafblind*. Springer, 2003.

[16] “Signalers for the Deaf and Hearing Impaired,” *Assistech Special Needs*, 2009. [Online]. Available: http://www.assistech.com/signalers/. [Accessed: 23-Jan-2015].

[17] “Houston TX Garage Door Repairs | Garage Door Repairs,” *Ahoustontxgaragedoorrepairs*. [Online]. Available: http://ahoustontxgaragedoorrepairs.com/. [Accessed: 26-Jan-2015].

[18] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, and others, “Internet of things strategic research roadmap,” *Internet of Things-Global Technological and Societal Trends*, pp. 9–52, 2011.

[19] “Ethics and software development,” 15-May-2006. [Online]. Available: http://www.ibm.com/developerworks/rational/library/may06/pollice/. [Accessed: 29-Jan-2015].

[20] B. Penzenstadler, “Supporting sustainability aspects in software engineering,” in *3rd International Conference on Computational Sustainability (CompSust)*, 2012.
[21] K. H. & R. Mckenzie, “InfoBarrel,” *InfoBarrel*. [Online]. Available: http://www.infobarrel.com/Explanation_Of_Three_Research_Paradigms-Positivism_Realism_And_Interpretivism. [Accessed: 29-Jan-2015].
[22] A. Håkansson, “Portal of Research Methods and Methodologies for Research Projects and Degree Projects,” in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering FECS'13*, 2013, pp. 67–73.
[23] Colin Neville, *An introduction to research & research methods*. University of Bradford, School of Management, 2007.
[24] “Top 10 Internet of Things Investment Rounds of 2014,” *Postscapes*. [Online]. Available: http://postscapes.com/top-10-internet-of-things-investment-rounds-of-2014. [Accessed: 19-Jan-2015].
[25] Nick : Editor at Large -Leardon Solutions, “What is a Production Prototype?,” *Product Development Solutions*. Available: http://leardon.com/what-is-a-production-prototype.
[26] “Signalers for the Deaf and Hearing Impaired,” *Assistech Special Needs*, 2009. [Online]. Available: http://www.assistech.com/signalers/. [Accessed: 23-Jan-2015].
[27] Serene Innovation Inc., “CentralAlert Model CA-360 Visual Alert System.” .
[28] Bellman & Symfon AB, “Technical Solutions 1.4.” .
[29] Clarity, a Division of Plantronics, Inc, “Clarity AlertMaster AL10 UserGuide.” .
[30] Silent Call Communication, “Silent Call Communication Sidekick II Receiver #SK2-SS.” .
[31] Sonic Alert, “Sonic Alert Wireless doorbell signaler model db1000 instructions.” .
[32] D. Bandyopadhyay and J. Sen, “Internet of things: Applications and challenges in technology and standardization,” *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011.
[33] “LIFX - Live a more illuminated life.,” *LIFX*. [Online]. Available: http://www.lifx.com/collections. [Accessed: 24-Apr-2015].
[34] “Visualight,” *Kickstarter*. [Online]. Available: https://www.kickstarter.com/projects/leifp/visualight. [Accessed: 24-Apr-2015].
[35] “Spark,” *Spark*. [Online]. Available: http://www.spark.io/. [Accessed: 24-Apr-2015].
[36] “Samsung, Galaxy, Tab S, tablet, super AMOLED, Wi-Fi,” *Samsung SE*. [Online]. Available: http://fb.se.samsung.com/consumer/mobile-devices/tablets/galaxy-tab-s/SM-T800NTSANEE. [Accessed: 24-Apr-2015].
[37] “USB ALERT,” *Polycom Technologies AB*. [Online]. Available: http://www.polycom.se/newLook/usbalert.htm. [Accessed: 29-Apr-2015].
[38] “Bellman & Symfon,” *Mobile Phone Sensor*. [Online]. Available: http://bellman.com/en/for-professionals/product-support/visit-notification-system/accessories/mobile-phone-sensor/. [Accessed: 29-Apr-2015].
[39] Ulrika Sundström, “Intelligenta hus,” *Byggahus.se*, 15-Sep-2008.
[40] Oskar Lundh and Andreas Hallberg, *Home Automation: An alternative solution to environmental problems using Raspberry Pi and Android*, Bachelor's thesis, KTH Royal Institute of Technology, TRITA-ICT-EX, 2013:131, Stockholm, 2013, urn:nbn:se:kth:diva-128510.
[41] “What is a Raspberry Pi? | Raspberry Pi.” [Online]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Accessed: 30-Apr-2015].
[42] T. Hådén, *IPv6 Home Automation*. Bachelor's thesis, KTH Royal Institute of Technology, TRITA-ICT-EX, 2009:28, 2009, urn:nbn:se:kth:diva-91502.
[43] M. Tesfaye, *Secure Reprogramming of a Network Connected Device : Securing programmable logic controllers*. Master's thesis, KTH Royal Institute of Technology, TRITA-ICT-EX, 2012:229, 2012, urn:nbn:se:kth:diva-104077.

[44] "Cisco Cloud Portfolio: Enabling the World of Many Clouds," *Cisco*. [Online]. Available: http://cisco.com/c/en/us/solutions/collateral/service-provider/service-provider-cloud-data-center-solutions/solution_overview_c22-693654.html. [Accessed: 30-Apr-2015].
[45] "Amazon Web Services (AWS) - Cloud Computing Services," *Amazon Web Services, Inc.* [Online]. Available: http://aws.amazon.com/. [Accessed: 05-May-2015].
[46] Md. Iqbal Hossain and Md. Iqbal Hossain, *Dynamic scaling of a web-based application in a Cloud Architecture*, Master's thesis, KTH Royal Institute of Technology, TRITA-ICT-EX, 2014:13, 2014, urn:nbn:se:kth:diva-142361.
[47] "AWS Products and Services - Global Compute, Storage, Database, Analytics, Mobile, Application, and Deployment Services," *Amazon Web Services, Inc.* [Online]. Available: http://aws.amazon.com/products/. [Accessed: 05-May-2015].
[48] M.-S. Lang, *Remote Residential Control System*. Master's thesis, KTH Royal Institute of Technology, IMIT/LCN 2004-03, 2004, urn:nbn:se:kth:diva-93044.
[49] L. Subramanian, *Security as a Service in Cloud for Smartphones*. Master's thesis, KTH Royal Institute of Technology, TRITA-ICT-EX-2011:143, 2011, urn:nbn:se:kth:diva-47886.
[50] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, and others, *SIP: session initiation protocol*. RFC 3261, june, 2002.
[51] M. Altinkaya and S. Ahmedi, *SIP in an Interconnector and Service Provider Role*. Master's thesis, KTH Royal Institute of Technology, 2001, urn:nbn:se:kth:diva-93274.
[52] "Scrum Methodology & Agile Scrum Methodologies." .
[53] M. James, "Scrum reference card," *CollabNet Inc*, 2010.
[54] M. Huo, J. Verner, L. Zhu, and M. A. Babar, "Software quality and agile methods," in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, 2004, pp. 520–525.
[55] "Microchip," *Microchip Technology*. [Online]. Available: http://www.microchip.com/. [Accessed: 05-May-2015].
[56] "ASUS," *ASUS USA*. [Online]. Available: http://www.asus.com/us/. [Accessed: 05-May-2015].
[57] "Raspberry Pi." [Online]. Available: https://www.raspberrypi.org/. [Accessed: 05-May-2015].
[58] "World's smallest mini computers for XBMC, Android TV & Linux," *SolidRun.com*. [Online]. Available: http://www.solid-run.com/. [Accessed: 05-May-2015].
[59] "Elfa Distrelec | Elektronik & Automation." [Online]. Available: https://www.elfa.se. [Accessed: 05-May-2015].
[60] "Mini PC: Intel® NUC," *Intel*. [Online]. Available: http://www.intel.com/content/www/us/en/nuc/overview.html?track=1&id=f48ca76b-2d82-4cf3-a9a1-2011abcca345&type=PER&val=&orderId=&valueCurrency=&promoCode=&ref=http://www.google.com/url%3Fsa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CB4QFjAA&ei=yZdXVYavEIPcUaSMgKgK&usg=AFQjCNGJ3ywX-HaZ8bb8haiGRNCvS9yfLA&bvm=bv.93564037,d.d24. [Accessed: 18-May-2015].
[61] "Apple - Mac mini," *Apple*. [Online]. Available: https://www.apple.com/mac-mini/. [Accessed: 18-May-2015].
[62] "ASUS Chromebox - Experience the Best Chrome OS Desktop," *ASUS US*. [Online]. Available: http://promos.asus.com/us/chrome-os/chromebox/. [Accessed: 18-May-2015].
[63] "What is a Minicomputer? - Definition from Techopedia," *Techopedia.com*. [Online]. Available: http://www.techopedia.com/definition/4615/minicomputer. [Accessed: 05-May-2015].

[64] "Demo: Introduktion till CRM - Salesforce.com SE," *Salesforce.com*. [Online]. Available: https://www.salesforce.com/se/form/sem/crm-display-2013.jsp. [Accessed: 05-May-2015].
[65] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle, "Session Initiation Protocol Extension for Instant Messaging." [Online]. Available: https://tools.ietf.org/html/rfc3428. [Accessed: 19-May-2015].
[66] S. Guha, K. Biswas, B. Ford, S. Sivakumar, and P. Srisuresh, "NAT Behavioral requirements for TCP," *IETF Request For Comments (RFC 5382)*, 2008.
[67] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of JSON and XML Data Interchange Formats: A Case Study.," *Caine*, vol. 2009, pp. 157–162, 2009.
[68] Y. Fu and R. Li, *From Set Top Box to Home Media Center*. Master's thesis, KTH Royal Institute of Technology, TRITA-ICT-EX-2014:144, 2014, urn:nbn:se:kth:diva-153655.

# Appendix A: User survey

Utvärdering

**Var god och svara på frågorna:**

**Fråga 1.** Jag är en:

Ordinatör | Installatör | Brukare | Vet inte

**Fråga 2.** På en skala mellan 1 till 10, där 10 är en ypperligt bra produkt medans 1 är en väldigt dålig produkt, hur bra tycker du Omnitors Nya Produkt är:

5

Om du vill kommentera ditt val på "**Fråga 2**" kan du fylla i det här:

**Fråga 3.** Bocka för de egenskaper du tycker är viktigast hos Omnitors Nya produkt:

- **WiFi --- Att enheten kan ansluta via WiFi**
- **LAN --- Att enheten kan ansluta via LAN**
- **Plattformsoberoende --- Att enheten kan användas av vilken hårdvara/mjukvara som helst**
- **Slutande relä --- Att enheten har 4 slutande relän**
- **Radiostyrda el-uttag --- Enheten kan styra el-uttag med radio**
- **Säkerhet --- Enheten krypterar sin data**
- **Enkel --- Enheten har inga knappar eller liknande**
- **Framtidssäker --- Enheten kan användas en tid i framtiden**
- **Molnbaserad --- Logiken ligger på internet inte hos brukaren**
- **Fler Färger --- NOTiFY kan levereas i olika färger**

**Fråga 4.** Omnitor har en vision att det nya systemet ska anslutas till tjänster, bocka för det tjänster du HELST vill att Omnitor ska lägga till i Omnitors Nya Produkt

- **Skype --- Brukaren får varseblivning när de rings från Skype.**
- **SOS Alarm --- Brukaren får varseblivning när SOS alarm får närliggande alarm.**
- **E-Mail --- Brukaren får varseblivning när det kommer ett ingående mail.**
- **Lamp-inställningar --- Brukaren/Ordinatören kan ställa in hur lampor ska lysa/blinka när lådan aktiveras.**
- **Andra SIP Företag --- NOTiFY system kopplas in på andra SIP företags nät. (T.ex. T-meeting / nWise)**
- **Medicin Påminnelse --- Brukaren blir påmind att ta medicin ett visst klockslag**

Har du någon/några andra tjänster du skulle vilja ansluta kan du skriva in dem här:

**Fråga 5.** Omnitor har en vision att det nya systemet ska ansluta enheter, bocka för det enheter du HELST vill att Omnitor ska lägga till stöd för i Omnitors Nya Produkt

- **Smarta Lampor --- Smarta lampor som kan skruvas up var som helst.**
- **Dörrklocka --- En dörrklocka så man kan ringa på hos brukaren.**
- **Porttelefon --- En porttelefon så man kan ringa på hos brukaren.**
- **Brandlarm --- Brandlarm så att brukaren kan se att det brinner**
- **Smarta Klockor --- Brukaren kan få information via en smart klocka.**

Har du någon/några andra enheter du skulle vilja ansluta kan du skriva in dem här:

**Fråga 6.** Är det något annat som du vill säga om produkten kan ni skriva det här:

**Spara**

Omnitor AB

## Appendix B: Detailed results

| user | goodorbad | comment |
|---|---|---|
| **installer** | 9 | |
| **installer** | 7 | Jag saknar en liten fast monterad blinklampa på lådan. Man borde kunna välja mellan att ha blixtljus (=kraftiga blixtar) eller lampljus (=lugnare ljussignaler). Om den fasta blinklampan finns med från början så räcker det med att ta med enbart den och en strömadapter på t ex resor och hotell. Supportens Notify-låda ser lite rörig ut med många sladdar inkopplade idag. |
| **ordinator** | 8 | |
| **user** | 5 | Gör inte så stor skillnad mot usb-alert till PC:n, slipper en kabel och det är skönt. |
| **ordinator** | 8 | Ger den en 8 av 10, dÅ jag anser att den fyller funktion fÖr smarta enheter. FÖr pc sÅ fungerar det lika bra med usb-alert. Om man ska rÃƒÂ¤kna in alla funktioner som kan tÄnkas tÄckas in i framtiden sÅ skulle den fÅ hÖgre betyg, men det har vi inte sett att det funegrar Än sÅ jag bedÖmmer utifrÅn den funktion ni idag erbjuder. |
| **installer** | 6 | Bra att den kan fungera trÅdlÖst mot telefoner och surfplattor, men dÅligt att den idag bara fungerar mot omnitorprodukter och Är molnanknuten. |
| **ordinator** | 5 | |

| WiFi | LAN | Platform-independent | Four Relays | Radio | Security | Easy to use | Future Secured | Cloud Based | Multiple Colors |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **1** | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0** | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |

| Skype | SOS-Alarm | E-mail | Lamp-Settings | Other SIP networks | Medicine Reminder |
|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 0 | 1 |
| **1** | 0 | 1 | 1 | 1 | 0 |
| **1** | 1 | 0 | 0 | 1 | 0 |
| **0** | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 1 | 0 |
| **1** | 0 | 0 | 1 | 1 | 0 |

| Other |
|---|
| |

| |
|---|
| **1) Det vore bra att fÅ varseblivning nÄr man fÅtt ett nytt meddelande i totalsvararen. Detta kan visas genom t ex en speciell lampa som signalerar om totalsvararen har fÅtt olÄsta meddelanden samtidigt som man fÅr epostnotifiering.<br><br>2) Det vore intressant med en notify-app till mobilen som talar om molnstatus i Notify-enheten (t ex aktiv, ej aktiv, att alla system Är ok) och vilka tjÄnster som Är aktiverade i den. Det skulle vara kul att kunna fjÄrrstyra olika enheter genom mobilen, t ex tÄnda och slÄcka lampor i hemmet och ÄndÅ ha kvar varseblivningsfunktionen parallellt. Om vi tillverkar en rolig och anvÄndbar produkt sÅ kommer folk kÖpa den.<br><br>2) "SOS Alarm --- Brukaren fÅr varseblivning nÄr SOS alarm fÅr nÄrliggande alarm." mÅste motsvara VMA-tjÄnsten i detta fall (dvs VMA=Viktigt Meddelande till AllmÄnheten). Ja bra att ha men inte nÖdvÄndigt. LÅg prio fÖr mig. Men om landsting och AF Är beredda att betala extra fÖr en "Notify Pro"-version som har "allt" sÅ Är det ok med en VMA-tjÄnst till den ocksÅ.** |
| |
| **Men om SOS-alarm ska vara av intresse mÅste det finnas en koppling till en bildskÄrm som visar vad som hÄnt varfÖr SOSlarmet har brytit ut.<br>Ãƒâ€“nskar hellre att NOTiFY indikerade pÅ VMA.<br>Stort ÖnskemÅl att NOTiFY skulle kunna indikera med lampor pÅ inkommande samtal till Facetime.** |
| **Facetime** |
| |
| **Brukaren fÅr varseblivning nÄr det ringer i mobilen, Skype, FaceTime...** |

| Smart Lightbulbs | Doorbell | Doorphone | Firealarm | Clocks | Other |
|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 1 | |
| **1** | 0 | 1 | 1 | 1 | "Smarta Lampor --- Smarta lampor som kan skruvas up var som helst." Ett "p" har tappats bort i rubriken ovan. |
| **0** | 1 | 1 | 1 | 0 | |
| **0** | 0 | 0 | 0 | 0 | Inte av intresse, vill behÅlla Bellmansystemet, invant och tryggt. |
| **0** | 0 | 0 | 0 | 1 | |
| **0** | 1 | 1 | 1 | 0 | |
| **1** | 1 | 1 | 1 | 0 | Befintliga saker i hemmet, som lampor etc. |

| extra |
|---|
| |
| **Landsting och AF borde betala ca 6500 kr plus moms som engÅngskostnad fÖr Notify-lÅdan . DÅ ligger priserna ganska nÄra Bellmans Visit-produkter. Men om vi startar en riktig Notifiy-molntjÄnst sÅ kan vi ta t ex 5000 kr + moms fÖr fem Års anvÄndning och ta samma pris eller nÅgot lÄgre pris en gÅng till nÄr fem År har gÅtt.  Jag tror att det Är mycket bÄttre fÖr oss att** |

| |
|---|
| **erbjuda tidsbegrÄnsade molntjÄnster pÅ t ex 1,3 eller 5 År som vi kan ta betalt fÖr.<br><br>Privatpersoner  kan kanske kÖpa Notify fÖr 2495 kr inkl moms som en engÅngskostnad och dÅ fungerar den som den ska gÖra i t ex 3 eller 5 År. Konsumentprodukter fÖr automatisering och fjÄrrstyrning av hemmet ligger i prisklasssen ca 2000-3500 kr exkl moms. NÅgra prisexempel kommer hÄr:<br><br>1. https://www.dustin.se/product/5010807457/smart-alarmkit<br>2. https://www.dustin.se/product/5010806220/zipabox-gateway-smart-home<br>3. https://www.dustin.se/product/5010811244/mydlink-home-kit---large<br><br> (TvÅ smÅ stavfel i texten ovan. dvs "remligt pris" = "rimligt pris", "framtidtsÄker" = "framtidssÄker".)** |
| |
| **Ãƒâ€"nskar att det gÅr att fÅ bellman-lampor att trigga pÅ inkommande samtal till Facetime.** |
| |
| **SvÅrt att dra ur nÄtverkskabeln, behÖvde anvÄnda ett verktyg fÖr att kunna komma Åt spÄrrflÄrpen pÅ nÄtverkskabeln. Borde finnas en testknapp fÖr varseblivningen i mjukvaran eller pÅ dosan. Ser hellre att dosan reagerar pÅ typ inkommande trafik Än att leverantÖrer ska ansluta sig till en molntjÄnst. Eller att notify sitter pÅ nÄtverket och att programvarorna skickar signal direkt via wifi till den nÄr man Är uppkopplad pÅ samma nÄtverk. DÅ varslar ju inte notify nÄr man inte Är hemmavid vilket Är bra. Konstigt upplÄgg med att en lampa ska blinka och den andra ska lysa fast nÄr allt Är ok. Mycket enklare om bÅda ska lysa fast eller blinka. Jag vet att vi testat en beta nu sÅ funktionen kanske kommer, men det vore bra om man sjÄlv kan fylla i vilken sip-adress som enheten ska varsla mot.** |
| **1) Tre rÖda lampor och en grÖn som blinkar/flimrar/lyser med fast sken. Vad betyder det?<br>2) Om den bara har stÖd fÖr Omnitors produkter sÅ Är den inte sÅ intressant.** |

TRITA-ICT-EX-2015:179