UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

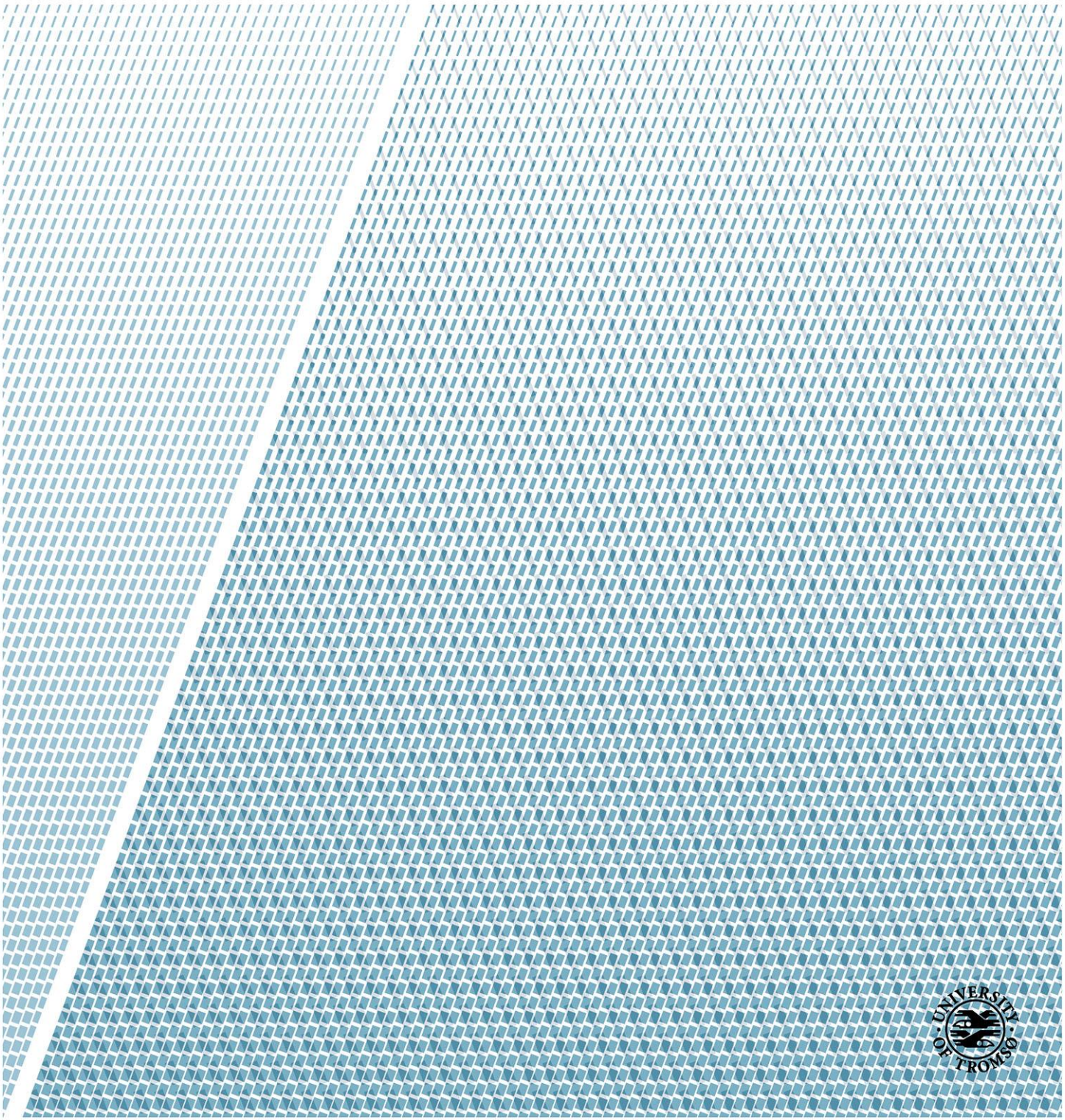
Faculty of Science and Technology
Department of Computer Science

Beyond the hospital walls - Extramural mobile communication with CallMeSmart

—

Bård Johan Hanssen

INF-3981 Master thesis – June 2015



Abstract

Physicians, nurses and other people who work at hospitals and in the health-care sector often move around a lot. As they also frequently need to contact each other, mobile communication devices like pagers and wireless phones are important tools for letting them maintain contact. This can lead to problems as personnel are constantly interrupted during their work because coworkers don't know whether they are available to talk when contacting them. CallMeSmart is a context-aware communications system that aims to reduce the number of interruptions that hospital employees receive. By using information related to users' location, schedule and general availability, the system can inform callers that the callee is busy and that the caller should probably not attempt to contact them unless the situation is urgent. This thesis proposes an additional tool for allowing users of the CallMeSmart system to communicate by introducing GSM functionality to support the IP based calling infrastructure that currently exists. This opens up possibilities for users to communicate with each other even when outside the hospital.

Preface

Going into the final year of my master's degree, I had some trouble deciding courses to take as the semester did not have have required courses. Knowing that there were several projects going on at and around the university I reached out to Professor Gunnar Hartvigsen hoping to hear if he knew of or had any ongoing projects that might be of interest. While there were several ongoing projects to choose from, CallMeSmart was the one that piqued my interest. Meeting with Terje shortly after led to me writing a capstone project related to the CallMeSmart project. Come the final semester, and there was not really any doubt what I would end up writing about for my master's thesis. Working on CallMeSmart has been both fun and interesting, and I wish the people surrounding the project the best of luck going forward. The passion for what they are doing is inspiring, and I have no doubt that they will do great things. Congratulations must go out to Terje Solvoll who initially began the project, as well as all the others who have worked on the project, as it was recently decided that CallMeSmart would begin a pilot project at the hospitals in Finnmark as a part of the Work Smarter project at the Northern Norway Regional Health Authority (Helse Nord). I wish them the best of luck going forward.

Acknowledgements

I would like to thank my advisors Terje Solvoll and Gunnar Hartvigsen for repeatedly giving me valuable feedback, encouraging me when I had trouble staying motivated, and for believing in me. I also want to thank Alain Giordanengo for being willing to help whenever I had question about the CallMeSmart system. This thesis would not be possible without their help.

Furthermore, I would like to thank my classmates and friends for their help,

support, willingness to put up with me, and just being great people in general. Their friendships have made these last few years go by incredibly quickly, and I will be sad to see them leave for what is clearly inferior geographical locations.

Finally I wish to thank my family for being supportive of what I do and always encouraging me to do what I want, as long as it is within reason, and sometimes even if it is not.

Contents

Acronyms

AD Active Directory.

AGI Asterisk Gateway Interface.

ARP Address Resolution Protocol.

BSOD Blue Screen of Death.

BYOD Bring your own device.

CH Correspondent Host.

CMS-VS CallMeSmart Virtualization Server.

DECT Digital Enhanced Cordless Telecommunication.

DTMF Dual-tone multi-frequency signaling.

EOL End-of-life.

FA Foreign Agent.

FXO Foreign eXchange Office.

GPL General Public License.

GPRS General Packet Radio Service.

GPS Global Positioning System.

GSM Global System for Mobile Communications.

GVU Graphics, Visualization and Usability.

HA Home Agent.

IETF Internet Engineering Task Force.

IMS Integrated Message Server.

IoT Internet of Things.

IR infrared.

JDK Java development Kit.

LDAP Lightweight Directory Access Protocol.

LTE Long-Term Evolution.

NFC Near Field Communication.

NST Norwegian Center for Integrated Care and Telemedicine.

OJS Open Java Server.

PBX Private Branch Exchange.

PDA Perosnal Digital Assistant.

PKI Public Key Infrastructure.

RTCP Real-time Transport Control Protocol.

RTT Round-Trip Time.

S3 Samsung Galaxy SIII.

S4 Samsung Galaxy SIV.

SCO Synchronous connection-oriented.

SELinux Security Enhanced Linux.

SIM Subscriber Identity Module.

SIP Session Initiation Protocol.

SOAP Simple Object Access Protocol.

SVN Subversion.

UIT The Arctic University of Norway.

UMA Unlicensed Mobile Access.

UNN The University Hospital of North Norway.

VoIP Voice over IP.

VoWiFi Voice over Wi-Fi.

VPN Virtual private network.

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background and motivation

in the early year of mobile communications, having a mobile phone did not necessarily make someone easier to contact. Early mobile devices were overly clunky and impractical for many to carry around. As a result, calling a landline could in many cases be as effective if not more as calling someone on their mobile phone. Pagers were often used as a more lightweight solution, offering the callee a way of knowing that someone wanted to contact them. Early pagers did not inform the wearer who made the call, which meant that the wearer would either have to guess who made the call or they could call the service center who could then inform them who made the call. As the technology advanced, pagers were able to show the number of the caller, as well as being able to display number codes and being used to send messages. While helpful, it did not help the fact that the callee would need to locate a phone in order to contact the caller if the user wanted verbally communicate with the sender of the page. As phones became smaller and offered more utility, the use of pagers began to decrease, and Norway dropped support for its pager networks in 2005 [3], though hospitals and some other public services still maintain their own pager networks for internal use. Today, cell phone coverage has improved tremendously since the early years of mobile communication, and line line numbers are in steady decline both in Norway and globally[64, 66]. With improved signal coverage, the main problem with mobile communication is often no longer to reach the desired callee, but rather knowing whether the callee wants to be contacted and if they are able to respond. A person sitting in a meeting would normally not want to be

disturbed, but they might be willing to make an exception depending on the importance of the call as well as the identity of the caller. In addition, unless the callee picks up and answers the phone, the caller is none the wiser as to why the callee doesn't answer.

1.1.1 Context-aware systems

A context-aware system is any system which uses any kind of contextual information in order to adapt itself to the situation. In the case of mobile communication, context can mean a lot of things, and several different definitions have been presented over the years[58, 61, 9, 20, 59, 52, 45].

In Mizzaro et al. [52] mention that almost any available information available during an interaction can be seen as context information, and goes on to mention some more commonly used types of information used in order to find context. Some of these types of information are presented below.

- Spacial information
- Temporal information
- Nearby resources
- Schedules
- Social situation and identity

Spacial information is perhaps the most frequently used type of contextual information, pertaining to a users location. It can be as simple as knowing whether a user is at home or at work, or more precise by specifying which office floor they are currently on. The benefits of having this kind of information is obvious. Trying to find someone is a lot easier when you know which building they are currently in. In addition, if a user uses a context-aware system in order to find someone else, the system can inform the user that the person they are looking for is currently in a meeting and that looking for them now is not ideal as they are busy and might not be able to talk to them anyway.

Temporal information can be of high value when collecting contextual data, and many ways of presenting this information has been proposed[82, 12, 31].

An example of how to use temporal information is to use it to check for colliding or overlapping events in calendars and schedulers. Other example uses involve combining temporal data with, as an example, sales data, and looking for connections between time of day and sales of certain items. Horvitz et al.[40] presents a system which allows for setting of thresholds for alerts on desktop and mobile alerts depending on urgency, availability, and other factors, including temporal data.

Knowing what resources are available nearby can also be used in order to help users in a variety of situations. This could be as simple as informing the user that a meeting room has been freed up or where the closest room with an available projector is. Taking this further, a system might recognise that a user currently located in a meeting room is about to make a presentation due to an event in that user's schedule, and as a result it automatically connects the users laptop to the projector and speaker system located in the room.

Having access to a users schedule can offer a lot of information. Knowing when they are going to meetings, seeing their doctor, or taking vacation can be very useful for providing relevant information without the user having to look it up themselves. An example of using this kind of information can be found in the interaction between Google's services. Gmail offers functionality which can automatically import event data to your calendar. Many applications also exist which automatically puts your phone on silent during meetings and other events by checking your calendar. Similarly a phone system can note that a callee is in a meeting and inform the caller of when they are expected to be done so that the caller has a higher chance of reaching them the next time they call, or it can present the caller with a set options as to what to do depending on the context of the callee as presented in Chihani[28].

The identity of the user as well as those attempting to contact them, or be contacted by them, is also a useful thing to know in terms of context. We mentioned earlier that a context-aware system could be used in order to reduce the availability of the user if they were in a meeting, but what if the one calling is the user's supervisor or someone with an urgent need to contact the user? This is mentioned in [77] amongst others, and in reality, people are often willing to be interrupted if the responsible party has a valid reason for it[48]. Identity can also be used in order to tailor a users surroundings. A user might have his office set up so that the lighting is adjusted based on the time of day, or having the temperature turned down and the lights turned off when they aren't there. Chen et al.[26] features more examples of how

various contextual information can be utilized.

For this thesis, the same definition of context that has been used to describe context for the CallMeSmart project earlier will be used. In [71], Solvoll uses the same definition of context as presented in [9]:

> *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*

context-aware system's are not a new phenomenon. While the term *context-aware* was first used to describe a system in 1994[59], the concept had already been in use for years. Schilits et al.[60, 58] gives a good overview of some earlier systems, the usage of context-aware systems, and design objectives that they believe should be considered.
One of the first system developed was the Active Badge System first deployed at Cambridge in early 1990[77, 37]. The system used lightweight badges which the users carried. These badges used infrared (IR) communication in order to send out a short signal every so often which could then be picked up by network sensors that were placed around the building in which the system was deployed. The information gathered by these sensors could be presented on a standard PC display showing users' name, location, and the probability of them being in the given location. The probability of a user being in their last location was based on whether the system noticed that the user was moving around recently. The system was primarily used by receptionists who used the presented information in order to forward phone calls. Though some were sceptical towards the system at first, it quickly became a useful tool, and many praised the system for making their lives easier in regards to responding to people trying to reaching them, though they wanted more control over when calls could be forwarded[77].

Another early system that used a similar system was the PARCTAB system develop at Xerox PARC in the early nineties[10, 78]. PARCTAB was a Perosnal Digital Assistant (PDA) whose applications mostly ran on remote machines and was primarily meant for in-building use. Like the Active Badge System[77] it used IR communication in order to communicate with these applications. While the Active Badge System had network sensors which only picked up signals from the badges, the PARCTAB system used a series of transceivers as the PDAs needed to both send and receiver data.

Another early use of context-aware system's was to create systems that could provide guidance to its users. One of the earlier systems that did this was Cyberguide[46, 8] which was developed at the Georgia Institute of Technology. Cyberguide acted as an electronic guide system for visitors of the Graphics, Visualization and Usability (GVU) Lab during open houses. Using awareness of the users position the system could provide the user with information about projects and demonstrations either when the user approached the demos, or when the user selected it on the devices interactive touch display. The Cyberguide project also included another guide system that covered a much larger area and wasn't limited to indoor use. Referred to as CyBARguide[8], it used the Global Positioning System (GPS) in order to find the location of the user. In a similar vein, the system described in Smailagic[67] also features navigation assistance as well as some other features that were not necessarily based on context awareness. As context-aware services can be difficult to develop and test, some have found creative ways of testing services while under development like using a modified game engine to simulate the real world[24].

In more recent times, some businesses and retail stores have started to take advantage of the possibilities that context-aware systems can provide. As an example, Estimote, a company which produces lightweight beacons which can be used for location tracking promotes the idea of combining beacons with an application on the consumer's phone which can notify the user of sales and other deals going on when the application detects that the consumer is near the store[1]. In addition, the application can also show the consumer where in the store the desired product is located. These kinds of systems can be seen as a continuation of guidance systems like Cyberguide mentioned earlier in[46]. Similarly the aptly named recommendation systems, systems that try to predict the users interest in an area, product, or event, have also begun using contextual information in order to offer better information to their users[54, 76, 11].

context-aware system's also share similarities with another concept known as ubiquitous computing, a term initially coined in the late eighties at Xerox PARC[79, 80]. The general concept is the idea of computers being everywhere and adapting to our presence and surroundings without us necessarily realising or having to think about it. An example of how ubiquitous computing is becoming more widespread is that more and more devices and items are becoming connected to the internet. This collection of devices is often referred to as the Internet of Things (IoT), and is often used when talking

[1] `www.estimote.com`

about ubiquitous devices. the IoT consists of billions of devices, and increases with every passing day. Cisco features a device counter on their site which keeps track of the number of devices connected to the internet, and estimates that the IoT will number more than 50 billion units by 2020 if the growth continues at its current rate[2]. An example of how the IoT might be of use could be an alarm clock that would be connected to the internet and used it to check the local weather and news. If an accident occurred between the owner's home and workplace, or the weather was making it so that traffic was moving slower than usual the alarm clock could wake the owner up earlier than normal so that they could still get to work on time. Going further, a user might have the alarm set so that if the weather is nice the alarm wakes them up earlier so that they can bike to work instead of taking the car. A simpler example might be a coffee machine that is is connected to the alarm clock and times it so that the coffee is ready as the user is getting up.

While the use of context-aware system's can potentially offer huge amounts of utility, many are still sceptical towards their use. To offer their services some context-aware systems often gather large amounts of personal data, and for many users the additional utility gained from the use of such systems do not outweigh the loss of privacy that the system generates. As a result of this, privacy is often referred to as the biggest concern when talking about context-aware systems[39]. This has lead to much research being been done in the area of protecting privacy in context-aware systems[39, 42, 65, 48, 18]. In [39], Hong et al. raise concerns about abuse and unease over the lack of control that users have over what information is gathered by these systems as some of the criticisms users have towards context-aware systems, an issue which is also discussed in Spreitzer et al.[72] and Barkhuus et al.[17].

Often, privacy is seen as a binary concept, where information is either available, or not at all. Examples of this includes using K-anonymity[74] and L-diversity[47], which is an extension of K-anonymity. While these properties are good in terms of preserving privacy, it can often reduce the amount of utility that can be extracted from the system. Other systems propose to treat privacy as a negative utility in order to increase the amount of information that the system can provide while still preserving the privacy of the users, as users often are willing to make exceptions to their privacy depending on the context of the person trying to reach them[48]. As earlier mentioned, people are often willing to be interrupted as long as there is a valid and urgent enough reason for it. As an example, someone hastily working towards a strict deadline might might be willing to be interrupted if someone needs

[2] `http://newsroom.cisco.com/ioe`

an urgent signature, but not if someone needs a confirmation of something that they could also get from someone else.

1.1.2 Context-Aware systems in hospitals

Communication in hospitals is often interrupt driven[29, 83], and in these environments interruptions can sometimes result in critical medical errors[38, 81]. In today's hospitals pagers are the primary tool for communication[57] and cause frequent interruptions[21, 43], e.g. the receiver of a page is not near a phone and so needs to locate one, or the sender of the page might not be near a phone when the receiver responds to the page. It's also not unusual for some physicians to carry several communication devices on their body[62, 57] depending on their role, e.g. head nurses or doctors on call often carry multiple devices. This can often end up reducing the effectiveness of alarms in what is known as alarm fatigue[63, 36, 32]. One alternative to using pagers has been to use mobile phones instead. While it has been shown that this can improve communications[51, 73, 34, 29], it has also been shown that phones can be more interruptive than pagers[62, 68, 69]. Part of the reason for this might be that workers often prefer to use synchronous forms of communication, even though it isn't efficient[55].

These things are important to consider when designing mobile communication systems for hospitals and the health care sector, as more than half of all medical information systems introduced fail due to staff resistance[13].

A proposed solution to reduce the problems associated with mobile phones has been to been to make use of the current context of the caller as well as the callee. Solutions have been presented where information like a users location, their role in the organisation, the timing of the delivery of information, as well as the state of devices used needs to be considered[35]. While the solution presented in [35] solves some of problems associated with mobile phones in hospitals, it also increases the number of devices that workers will need to carry, as it is not compatible with the different hospital infrastructures that exist. Studies have shown that when callers are provided with contextual information on the person they're calling it reduces the number of ill-timed calls[14].

1.2 Research problem and questions

For this master's thesis, the primary research problem is as follows; *How can we create a solution for an extramural communication service for the existing CallMeSmart system through a transparent Wi-Fi to Global System for Mobile Communications (GSM) switch?* This was expanded to the following subproblems:

- How can we provide all services offered by CallMeSmart on a public (GSM, General Packet Radio Service (GPRS), 2/3/4G) and private network (Wi-Fi at home).
- How can we maintain a communication channel when switching from a network to another (Roaming between intramural, public and private environments) without connection drops.

The research in this thesis is done as a part of the CallMeSmart project. The work being presented was done as a collaboration between The Arctic University of Norway (UIT) and the Norwegian Center for Integrated Care and Telemedicine (NST), as CallMeSmart is a product of NST.

1.3 Motivation

The current CallMeSmart system uses IP based calls and communications. The addition of GSM based communication for the system would allow users to move outside the range of the hospital network while still being able to communicate with other users in the system. This addition communication tool could also be used in order to provide a possible fallback solution in cases where hospitals might experience problems which could prevent the primary system from working. Network error which disrupt traffic can prevent IP based system from working entirely, potentially halting communications for days, as was the case at St. Olavs hospital in 2006[2] and at Akershus University Hospital in 2011[4].

Ultimately, the motivation for exploring extramural communication possibilities for the CallMeSmart system is to further improve the availablity of users when they should be available. Implementing GSM support aids this purpose by allowing users to reach users that are not able to be reached over the local hospital network.

1.4 Scenario

To illustrate a scenario where being able to hand off calls from one network type to another, we can imagine a nurse who is moving around the hospital. As she is making the rounds, she encounters a situation which leads her to making a call to a colleague. As her phone is connected to the hospital Wi-Fi and the CallMeSmart system, a /glsSIP based call is made. At some point the phone's connection to the Wi-Fi network might drop, be it due to a coverage dead zone, or perhaps for some other reason. Instead how having to wait until she regains her connection to the Wi-Fi network and making a new call to her colleague, a call can now be made over the GSM network instead. In order to restore the call the CallMeSmart system, which notes that the call was ended pre-emptively, performs a new call to the nurses' phone. Since the phone is no longer on the hospital network, the system uses its external trunk in order to make a call over the GSM network. As our nurse carries on, she eventually ends up getting a Wi-Fi signal back. As the call is currently taking place on the GSM network there is little need to switch back to an IP based call as long as there aren't any problems with the connection to the GSM network. As IP based calls are initialized quickly, an IP based call can be quickly established if the GSM call was to be dropped for some reason.

1.5 Research approach

The research performed was carried out three phases. Initially, research was made into possible ways of achieving the desired results as well as looking at what additional materials would be needed. After assessing what options were available, a choice was made on what solution to pursue. The second phase involved developing and merging the solution into the existing CallMeSmart system. In the third phase the implemented system was put through a series of tests in order to measure how well the system was working.

1.6 Scope and limitations

The scope of this thesis is to develop and test a external trunk solution for the CallMeSmart system that allows users to make calls and continue to communicate even if they are not located in range of the hospital's wireless network. This did not include sending of messages and alarms to users located outside the hospital. The system would also switch calls over to using the GSM network if noting that calls were prematurely ended, say in the case of a caller dropping their connection to the server as a result of moving outside the range of the wireless network.

Implementing support for sending and receiving of messages and alarms were not considered to be within the scope of this thesis, mainly due to the additional security requirements that would need to be met in order for these services to be allowed over external networks. As GSM is considered to be secure enough for calls, no additional security measures needed to be implemented.

1.7 Major results

The major results presented in this thesis is the implementation and testing of an external trunk for the CallMeSmart system that allows users to continue communications when leaving the coverage of the hospital's wireless network. The implemented solution has been tested and results have shown that the quality of calls utilising the external trunk handily meets the requirements for what are considered to be acceptable levels of quality.

1.8 Contribution

The contributions of this thesis to the CallMeSmart project is the research and implementation of a eternal GSM trunk that allows users to contact each other even if they are not connected to the wireless hospital network, as well as the ability to continue conversations that are interrupted as a result of either party losing their connection to the wireless network on the GSM network without user interaction.

1.9 Structure

The structure for the rest of the thesis is presented in the following table.

Chapter	Chapter contents
2 - Background	Introduces the problem area and gives information on the CallMeSmart system as well as other points related to this thesis.
3 - Theory and state-of-the-art	Presents the current state of the art.
4 - Materials and Methods	Presents the hardware and methods used in this thesis.
5 - Results	Presents the results found.
6 - Discussion	Discusses the findings from chapter 5.
7 - Conclusion	Concludes on the findings of the thesis and presents future works.

Chapter 2

Background

2.1 CallMeSmart

CallMeSmart is a context-aware system for intramural communication in hospitals developed at NST, a department under The University Hospital of North Norway (UNN), in cooperation with St. Olavs Hospital. CallMeSmart focuses on making communication between physicians more efficient by reducing the number of interruptions through the use of context-aware services. CallMeSmart is still in development and is currently being tested at the Oncology department at UNN where around ten users from a testing userbase of around one hundred users are switcing between testing the system simultaneously. During this ongoing trial period the system is used primarily by nurses, but some doctors are using it as well. The feedback received from users so far has been mostly positive[71]. Future works are also planned in order to make CallMeSmart ubiquitous and self-learning[70].

CallMeSmart gathers contextual data from various sources including the users calendar, work schedule, and position. The system uses this information to adapt the users phone profile and availability. Using information gathered from this data, the system is aware of when users are busy and has the option of redirecting callers to another physician if the availability of the person originally called is set to busy, or callers can force the call to go through even if the callee is listed as busy, say in case of an emergency.

The primary aim of CallMeSmart is to reduce the number of interruptions that physicians receive. In addition, as physicians often carry more than one

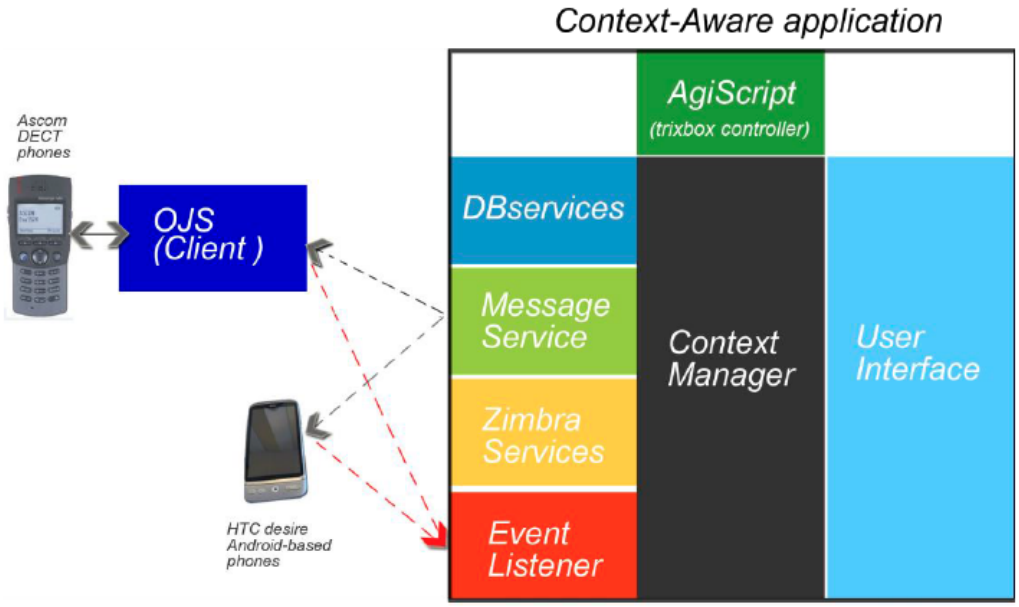


Figure 2.1: The overall software architecture of the CallMeSmart system. [71, Figure 4-14, p. 47].

device for communication[68], CallMeSmart offers added value by reducing the number of devices that physicians need to carry.

2.1.1 Architecture

Server architecture

The software architecture of the CallMeSmart system can be seen in figure 2.1. In addition, the framework architecture of the server can be seen in figure 2.2. The role of the different servers are presented in the following sections. The descriptions provided here are based on the data presented in [71], which covers the architecture of the system in more detail.

Ascom Unite and Ascom IP-DECT

The Ascom Unite system is a middleware platform that connects systems and allows for two-way communication between users and different systems. It consists of several modules which each handle different functionality. Ascom

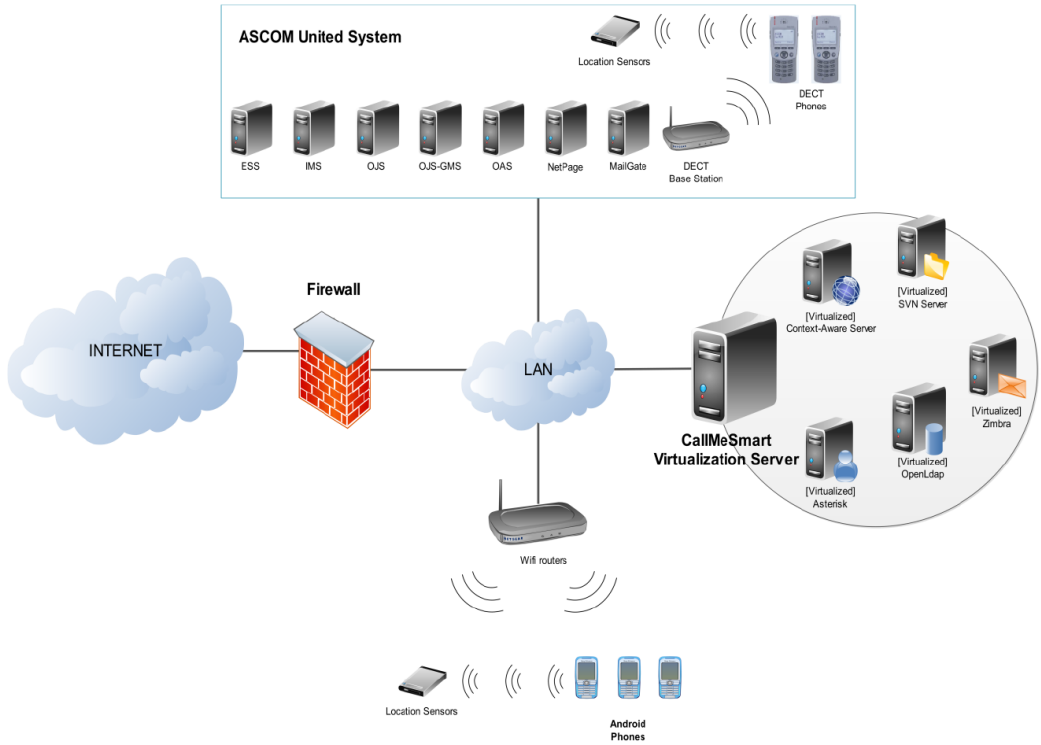


Figure 2.2: The overall framework architecture of the CallMeSmart system[71, Figure 3-6, p. 27].

Unite also interfaces with third-party systems. In CallMeSmart it handles communication with the Ascom Digital Enhanced Cordless Telecommunication (DECT) phones being used as part of the system.

Ascom IP-DECT is a communication system that integrates with call, messaging, and alarm services. It is used in combination with Ascom Unite in order to handle communication with Ascom devices.

CallMeSmart Virtualization Server

The CallMeSmart Virtualization Server (CMS-VS) is a Micosoft Server 2012 which is running VMWare Workstation 8 in order to virtualize all the machines and systems needed by the CallMeSmart system. These systems include the Context-Aware Server which handles the context-aware services of the system, an OpenLdap server which handles authentication of users, an SVN server that handles source control, and an Asterisk server acting as the systems Private Branch Exchange (PBX). Each system is described in more detail in the following sections.

Context-Aware Server

The Context-Aware Server consists of a set of software solutions that provide context-aware services by intercepting, analysing, and managing calls and messages. These solutions are presented in the following paragraphs. Initially the Context-Aware Server was running on a virtualized Windows 2008 Server R2, but as part of this thesis the server was migrated to run on a virtualized Debian 7 Wheezy server.

Context-Aware Application The Context-Aware Application is a custom made Java application which handles all context related services offered by the CallMeSmart system. It handles calls through the use of Asterisk Gateway Interface (AGI), which allows it to apply the context-aware services of the system to the Asterisk dialplan; it connects the Openfire-based messaging of the Android phones with the Open Java Server (OJS) managed messaging of the DECT phones; it communicates with the Zimbra server in order to get appointments for users, and it handles all interactions with the Context-Aware Database. It also send contact lists and other data to the Android phones using a Java API called JAX-WS.

(a) Calls per hour and per presence. (b) Call percentages per callee presence.

Figure 2.3: CallMeSmart call statistics.

As can be seen in figure 2.1, The Context-Aware Application's interaction with the Android phones differs from its interaction with the DECT phones. This is because the Context-Aware Application's communications with the DECT phones go through the client running on the Ascom OJS, while the Android phones communicate directly with the Context-Aware Application.

OpenFire The OpenFire server is a real-time collaboration server used by the Context-Aware application and the Android softphones for instant messaging services. Unlike the Ascom DECT phones the Android phones do not make use of the OJS for instant messaging. Messaging services are offered using the XMPP protocol.

Administration panel The administration panel is an application that allows administrators to manage groups, users, and the configuration of the global CallMeSmart system, including the Ldap server[84] and the context-aware database. It is a self-made Java EE program running on Glassfish. It also gives administrators access to call statistics like that shown in figure 2.3.

Context-Aware database The context-aware database consists of a set of databases running on PostgreSql 9.2 and is used to store all data used by the CallMeSmart system, including configurations, logs of services provided, account settings, and contextual data.

SVN server

An Subversion (SVN) server running on Ubuntu Serve 12.04 used to store the source code of every self-made application in the CallMeSmart project. This includes the Android Softphone, the OJS Client, the Context-Aware Application, the administration panel, and the location Application.

OpenLdap

The OpenLdap server implements the Lightweight Directory Access Protocol (LDAP) protocol for accessing and maintaining distributed directory information services over a hierarchical structure. It implements a self-made LDAP schema in order to provide a centralized authentication system for the CallMeSmart system (Asterisk and OpenFire). The CallMeSmart system is compatible with any LDAP server as long as the self-made schema is installed.

Zimbra

The Zimbra server runs an instance of the Zimbra Collaboration Suite, which is an enterprise open source email server which offers services through a Simple Object Access Protocol (SOAP) interface. It communicates with the Context-Aware Application through SOAP requests over a TCP-IP connection which makes it possible to extract user data stored on the server. The Zimbra server is used in order for the CallMeSmart system to extract appointments for the users. It runs on a emulated 64 bit Linux Ubuntu Server.

Asterisk

The Asterisk server is running an instance of Asterisk 1.8 and acts as the PBX of the CallMeSmart system. Asterisk is a software implementation of a PBX licensed under the General Public License (GPL). As mentioned earlier, The Asterisk server intercepts calls made on the network and then forwards them to the Context-Aware Application using AGI, which can then apply the context-aware services of the system to the call.

Ascom OJS Client

the Ascom OJS is a part of the Ascom IP messaging platform. It is a programming server, which makes it possible to implement customised features not covered by the standard Ascom UNITE system, and is directly interfaced with the Integrated Message Server (IMS). The OJS makes it possible to have Java applications communicate with the Ascom messaging system, which also allows it to establish communication with external systems such as CallMeSmart.

Ascom Integrated Message Server

The Ascom IMS is a middleware that helps connect the IP-DECT base stations to the other modules running on the Ascom Unite system. Its role in the CallMeSmart system is to manage and direct data to and from the DECT phones used by the system. It supports a set of services that are used by the system including message distribution, a cental phonebook, as well as an IMS messaging tool.

Phone software architecture

The software architecture of the CallMeSmart softPhone, showing how the various processes of the system are separated from each other, can be seen in figure 2.4.

2.1.2 Communication

CallMeSmart supports several forms of communication between users in the form of calling, messaging, and the ability to send out and answer alarms. Differing from many other alarm systems, which will alert all users when sending out an alarm, CallMeSmart uses contextual information in order to only sends out alarms to relevant users.

Calls are performed using Session Initiation Protocol (SIP) and acts much like it does with a normal phone. The differences come from the context-aware services of the system which decides what action to take depending on the availability setting of the user receiving the call. In addition to being

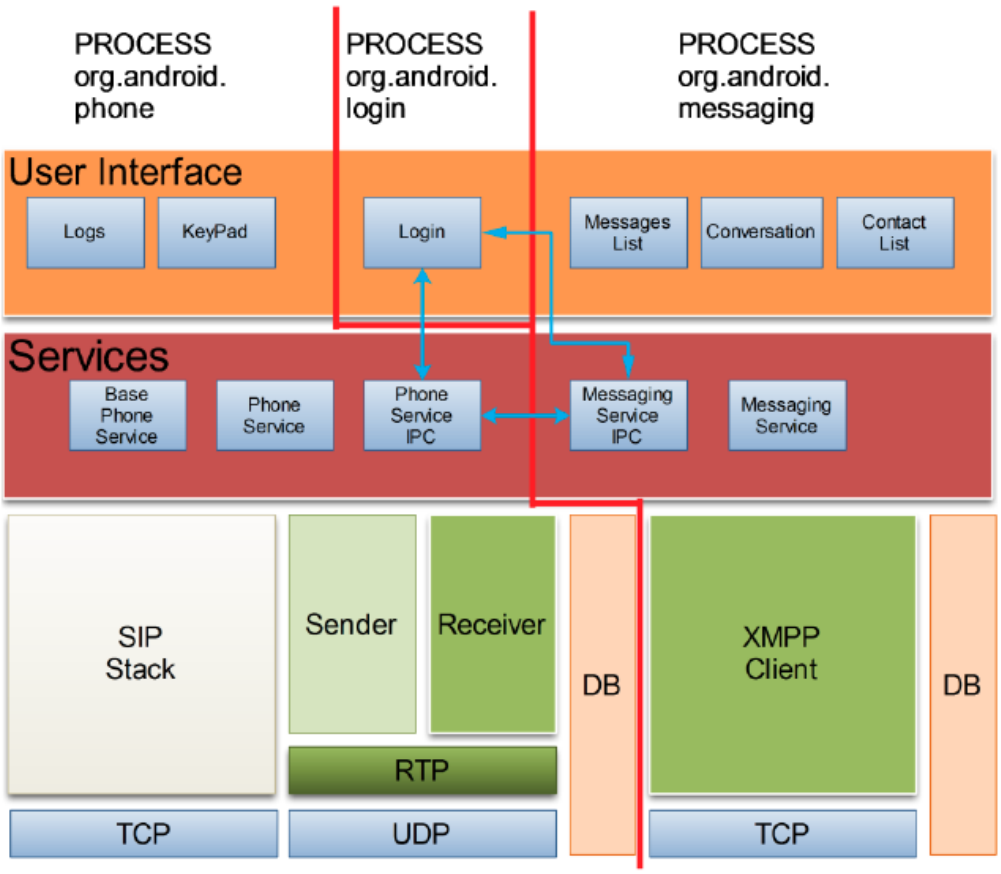


Figure 2.4: The software architecture of the CallMeSmart softPhone[71, Figure 4-15, p. 47].

able to call users in the CallMeSmart system, the softphones can also make calls to regular Ascom phones connected to the same system.

Messaging works similar to regular phone texting, but also gives the user sending messages feedback when the callee has read the message. This is helpful in that users no longer have to wonder whether the receiver has seen the message or not.

While calls are handled through a server using a software implementation of a PBX in the form of Asterisk[1], messages are handled through an OpenFire[2] server. The Context-Aware Application and Android smartphones use this server in order to share messages using the XML format. CallMeSmart handles all communication through TCP-IP sockets[71, p. 59]. The data flow of the system can be seen in figure 2.5 though it should be noted that the figure does not show the complete data flow. As mentioned in section 2.1.1 there is a missing data path which connects the Context-Aware Application directly to the phones through the Wi-Fi routers. This connection is amongst other things used to retrieve contact lists when users log on to the system and update contact lists on the server side when adding entries on the phone.

2.1.3 Handling availability

As mentioned earlier users of the CallMeSmart system can either set their availability manually or use the automatic option which uses location based information as well as information gathered from user's calendars and schedules in order to pick the option it deems most appropriate for the situation.

An overview of how a phone's default state varies depending on what mode it is currently in can be seen in table 2.1. If a user desires they can also choose what state should be associated with the different modes, as they can customize the associated values on an individual basis. When selecting an availability mode manually the phone stays in that mode until changed either by the system or the user, as user have the option of setting a timer which sets availability back to available a given time after the availability mode has been changed manually. When set to adjust availability automatically, the system uses contextual data about the user in order to adjust to the situation.

[1] `http://www.asterisk.org/`
[2] `http://www.igniterealtime.org/projects/openfire/`

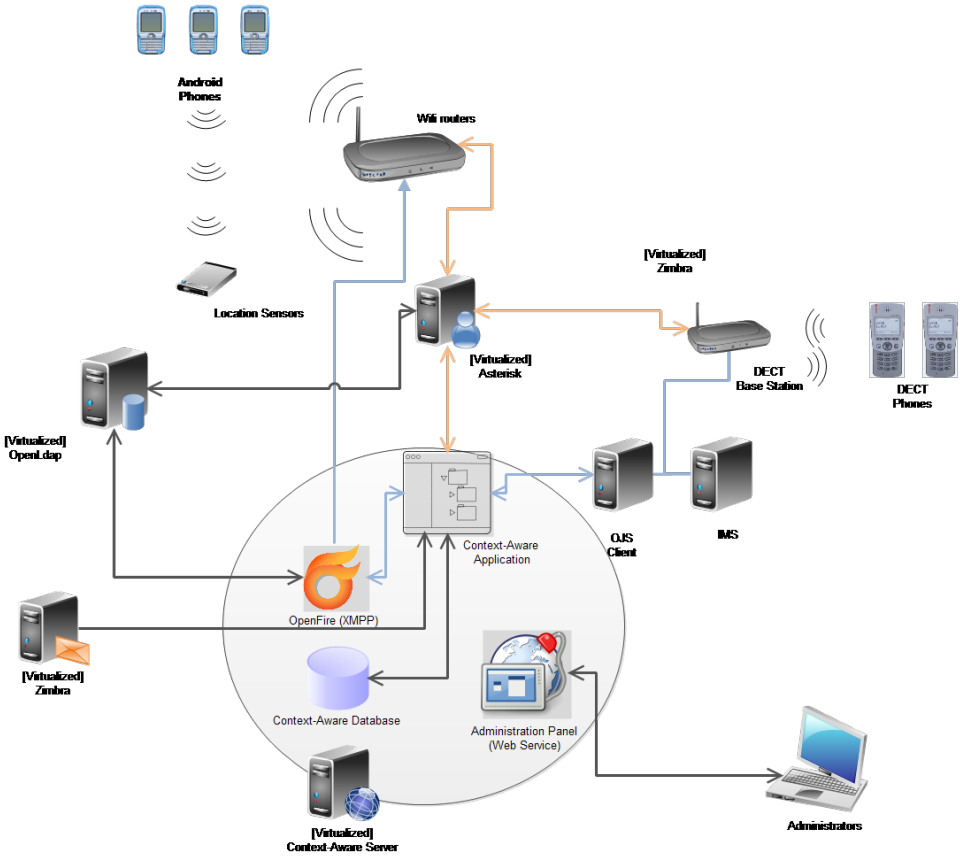


Figure 2.5: The global data flow of the CallMeSmart system. Yellow lines show calls, black lines show data, and blue lines show messages[71, Figure 4-13, p. 46].

Table 2.1: Default relation between phone mode and state[71, Figure 4-8, p. 42].

Mode	Critical area OR meeting	Global state
Available		Available
	X	Available
Busy		Available
	X	Available
Pager		Pager
	X	Pager
Auto		Available
	X	Busy

2.1.4 Security and authentication

As CallMeSmart is currently a closed system in that it only communicates with devices on the secured hospital network the authentication and security in place do not need to include support for communication with external networks. As was earlier mentioned in section 2.1.1, authentication is handled through the use of a server implementing an LDAP protocol using OpenLdap[3]. This authentication mechanism is also compatible with Active Directory (AD), the directory service some hospitals, like UNN, use in order to authorize hospital users and computers. In addition, Wi-Fi traffic is encrypted using shared key encryption, and any connection attempts from devices that do not have their MAC addresses registered to the server's list of approved devices are ignored.

Security requirements

Currently CallMeSmart meets the security requirements needed in order to be used at a hospital as long as the system remains on the closed network of the hospital. Additionally, GSM networks are considered secure enough for phone calls, meaning that for the goal of this thesis there is no need to implement additional security that what is already in place. However, if CallMeSmart was to start offering services like Voice over IP (VoIP) or messaging over either GSM or other types of external networks, e.g. regular IP connections over Wi-Fi using the internet, there are additional requirements that needs to be met. To offer these services over external networks CallMeSmart would need to need to meet the requirements of a so-called level four security system.

Security and authentication are an important part of systems with access to sensitive information. In 2008, the Ministry of Government Administration, Reform and Church Affairs, formerly known as the the Ministry of Government Administration and Reform, released a framework for authentication and non-repudiation for public services in Norway[49]. This framework presents four security levels for authentication and non-repudiation based on a set of security parameters which are listed below. The authentication level which a service is required to adhere to is based on what information the service gives the user access too. Table 2.2 shows what parameters are needed to satisfy the different security levels. For a level four system solutions which satisfy the requirements include authentication systems from Buypass,

[3] `http://www.openldap.org/`

Commfides, as well as bankID.

- Authentication factor specifications: Describes the number of authentication factors and their properties, e.g. dynamic or static. Examples of static factors include passwords and biometric data, while dynamic refers to things like BankID and Public Key Infrastructure (PKI) based solutions.
- Distribution: Describes how the connection between the authentication factors and the user identities are made. Does the user need to show up and identify themselves, or can they receive the authentication factor by mail.
- security requirements towards authentication factors during storage: Describes how the authentication factors should be stored and protected. As an example a password list stored on paper can be copied, but if presented as scratch cards the rightful owner will know if the passwords have been compromised as cards would need to be scratched.
- non-repudiation requirements: The degree to which it should be possible, at a later date, to document the authenticity that a user performed an action.
- Public approval requirements: If there are public specifications for this kind of solution, and states whether the solution has been declared by a public scheme.

Table 2.2: security levels for authentication and non-repudiation.

Level	Authentication requirements	Distribution to users		Storage	Approval	Non-repudiation
		Physical	Legal persons			
1	None	None	None	None	None	None
2	One-factor.	Mailed to registered address.	Mailed to registered address. The name of the physical person whom can sign for the legal person should be first on the shipment. Alternatively it can be sent to the registered address of the signer.	Both static and dynamic factors can be copied.	None	Routines and logs should be in place so there is a reasonable guarantee that a user is behind an action or piece of information.
3	Two-factor, whereas one is dynamic.	Same requirements as for level 2, but with an additional requirement that the distribution procedure should have an integrated security measure which should minimise the chance that the wrong person can make use of the solution. Personal attendance is not required.	Same requirements as for level 2, but with an additional requirement that the distribution procedure should have an integrated security measure which should minimise the chance that the wrong person can make use of the solution. Personal attendance is not required.	The dynamic factor can be copyable, the static cannot.	None	Routines and logs should be in place so there is a reasonable guarantee that a user is behind an action or piece of information.
4	Two-factor, whereas one is dynamic.	The requirements for registration and distribution are equal to the specifications for PKI, Person-High [50]. Personal attendance with legitimization at least once.	For legal persons the physical person who signs for the legal either show up in person, or give authorization to another who can attend in their place. Credentials shall be presented for both the physical and legal person, and both shall be checked against the Brønnøysund Register. Requirements equal to those set forth in the specifications for PKI, Enterprise [50].	Neither the static or the dynamic factor can be copyable.	The solution should be in accordance with the public specifications.	A communication party should be able to verify that another party is behind an action or piece of information. The communication party should not be able to produce or tamber with such evidence at a later date.

Chapter 3

Theory and state-of-the-art

3.1 Related technologies

This chapter covers two different areas that relate to CallMeSmart and the solution implement for this thesis. These areas are existing communications solutions within the hospital and healthcare sector and seamless handoff technologies.

section 3.1.1 displays some established communications systems within the hospital and healthcare sector while section 3.1.2 highlights some existing technologies that allow for seamless handoff between different communication networks.

3.1.1 Communication systems for hospitals

As earlier mentioned, communications systems are an important part of the day-to-day operations going on at hospitals and in the healthcare sector. Unsurprisingly, this means that there is a large number of communications systems and solutions available. When considering that more than half of all medical information systems introduced fail due to staff and user resistance[13], it is also clear that the people who use these systems do not accept half-baked solutions.

This section presents some well known solutions within the hospital and healthcare section that share similarities with the CallMeSmart system. Some

of these systems offer a combination off hardware and software solutions, others offer either hardware or software. The systems presented were found through a combination of searching the web using keywords like *hospital communications solution, ascom integration hospital communications solution*, looking at existing literature on context-aware services in hospitals, and talking to people at NST about existing systems.

Ascom

Ascom is one of the worlds leading providers of mission-critical communication systems with a primary focus on the healthcare sector. In the global wireless communications drive test they held a 38.2% market share in 2010[6].

While Ascom offers a large number of different communications devices for hospitals, the focus of this section will be on the product that relates the most to CallMeSmart. This product comes in the form of the recently released Ascom Myco[1]. Seen in figure 3.1b, it is a smartphone targeted towards healthcare workers and caregivers. A big advantage of the Ascom Myco is the fact that it is an Ascom product and as such is compatible with other Ascom systems. As a result of this it can be a lot easier to integrate into existing systems if a hospital is already using an Ascom system. The Ascom Myco is based on the Android operating system and uses Security Enhanced Linux (SELinux) in order to add mission-critical extension at an operating system level. Being an Android based phone, the Ascom Myco is also compatible with a wide range of Android applications, though it has been designed to suppress 3rd party applications in order to make sure that that the Ascom core applications always run reliably.

Being a specialized phone also gives the Ascom an edge over other software based solutions (like CallMeSmart) which simply take over an existing smartphone in that the phone can be built with specific features that are normally not found in regular phones. Some examples of this are the dedicated barcode scanner built into the phone and the display on the top of the phone which allows for quick at-a-glance notifications.

The Ascom Myco also features multi-carrier support, meaning that it supports both GSM call functionality as well as Voice over Wi-Fi (VoWiFi). The Ascom Myco's SIP client is also optimized for VoWiFi and offers seamless roaming, allowing users to move about without having to worry about the

[1] `http://www.ascommyco.com/en/`, Accessed 20.05.2015

phone losing connectivity. The Ascom Myco, like CallMeSmart, also uses some context-aware information in order to improve workflow. One example of how the Ascom Myco uses context is how alarms are handled. Instead of being broadcast to everyone in a ward, alarms initially only go out to the nurses assigned to the patient from whom the alarm originated, similarly to CallMeSmart and the Extension Engage system mentioned in section 3.1.1.

Vocera

Vocera is a provider for communication systems in hospitals and the health care sector. Their solutions offer seamless integration with other existing hospital systems, and offers some context-aware services, like alarm prioritization in order to improve the efficiency of users, or group-based calling, where a user can call a single number and the system will match the person to the most suitable person based on what they initially needed.

In addition to offering integration with esisting hospital infrastructure Vocera also offers solutions that can run on Android and iOS devices, allowing its users to use the devices they are the most comfortable with. They also offer the Vocera badge, a wearable device which allows for hands-free communication, which users to focus on what they are doing. It is shown in figure 3.1a.

Avaya

Avaya is a global provider of business collaboration and communications solutions. They offer solutions in the form of both hardware or software solutions, including integrated communication centers, conference phones, desktop phones, IP- and DECT phones.

Avaya one-X is an application lets users communicate with Avaya communication solutions through the use of smarthphones and tablets. It supports Bring your own device (BYOD), meaning that users have the option of using their own devices when communicating if desired, be it an Android, iOS, or Blackberry device. Avaya one-X also all allows users to hand over calls from Wi-Fi to cellular networks like GSM with the click of a button if users were tomove out of Wi-Fi coverage.

Avaya also offers a solution called Avaya Awareness, a context-aware solutions

(a) The Vocera Badge[2] (b) The Ascom Myco[3]

Figure 3.1: Some commercial communication tools for hospitals.

that aims to help users by providing them with information related to the situation they are currently in. Examples of this include providing users with relevant documents when joining a conference call, or showing relevant documents and relevant people to invite when receiving a call.

Extension Engage

Extension Engage is an alarm and response system developed specifically for use in hospitals and the health care sector. Much like CallMeSmart it aims to improve workflow by reducing the amount of interruptions that physicians receive.

Similarly to CallMeSmart, Extension Engage attempts to reduce alarm fatigue by using contextual information in order to make alarms go out to the most relevant personnel.

Extension Engage Mobile, which is the mobile component of the Extension Engage solution, lets users get access to relevant information like reports and contextual data from other systems connected to the Extension Engage solution in order to enhance workflow. Additionally, it gives users the ability to easily communicate with each other as well as respond to alarms on the fly.

[3]image used taken from `http://www.vocera.com/product/vocera-badge`. Accessed: 31.05.2015

[3]image used taken from `http://www.ascommyco.com/en/`. Accessed: 31.05.2015

Figure 3.2: Extension Engage mobile sample interface.[4]

The Extension Engage Mobile solution supports a high number of common device operation systems, including Android, Apple, Ascom, Avaya, Cisco, NEC, Spectralink, and Vocera. Figure 3.2 shows two different mobile devices running the Extension Engage Mobile platform. The left phone shows a user making a search and receiving a list of users along with their availability. The device on the right shows an incoming call along with some contextual information related to the call.

3.1.2 Seamless handoff technologies

The introduction of smartphones, tablets, as well as laptops becoming better and better in terms of battery life, power, weight, and size, has helped make communication while on the move much easier. Coupled with increased Wi-Fi and cell phone coverage, acts like checking and answering emails can be done with a few clicks from almost anywhere.

With all of these options available, naturally we want our devices to use the

[4]image used taken from `http://www.extensionhealthcare.com/extension-engage-mobile/`. Accessed: 31.05.2015

best available connection available. There is no need to use a 2G connection if there is a Wi-Fi network available which has both better bandwidth and signal strength. This process of having a device connect to more than one network as it moves around is often referred to as multihoming or seamless mobility. The idea of having devices seamlessly move between and make use of the best available network at the time has been explored extensively, from earlier implementations like in MosquitoNet[15, 27], Brattli[22] and Bharghavan[19], to more recent implementations like[23, 44, 56, 41]. In [22], Brattli discusses dynamic and seamless switching between wired and wireless networks by shielding system processes and applications from the interruptions that are normally caused by changing networks and goes on to mention how TCP/IP is not meant for mobile devices, as switching networks also requires that devices change their IP address[30], an action which can interrupt some operations.

Laptops are often connected to the wired network while in a cubicle, but when brought to conference rooms and meetings, they tend to rely on Wi-Fi for convenience. In this case, making a seamless switch from one network type to another is rarely that important. If a user is presenting something, the powerpoint file used is often already stored on the machine, or can be downloaded upon arrival. In most cases where a laptop is moving between different locations and access technologies there is no need to maintain a connection across different network types.

Smartphones normally default to using a nearby Wi-Fi network but can fall back to using networks like GSM or 4G Long-Term Evolution (LTE) if needed. This takes advantage of the strengths of both networks. Whereas Wi-Fi networks tend to have higher bandwith than cellular networks, the cellular networks provide a wider coverage area than Wi-Fi networks. This network switch, often referred to as a vertical handover, involves changing the technology used by a device in order to reach the supporting infrastructure it communicates with. Often, a device performing a vertical handoff will end up having its IP address changing as a result of changing the technology used in order to connect to the internet. For some services, a device changing its IP address mid service can cause interruptions. The difference between a vertical handoff and a horizontal handoff, where a device changes the access point it uses but the technology used remains the same, is illustrated in figure 3.3.

As smartphones are frequently used on the move, and are often used for streaming services like VoIP and video streaming. As a result, one would

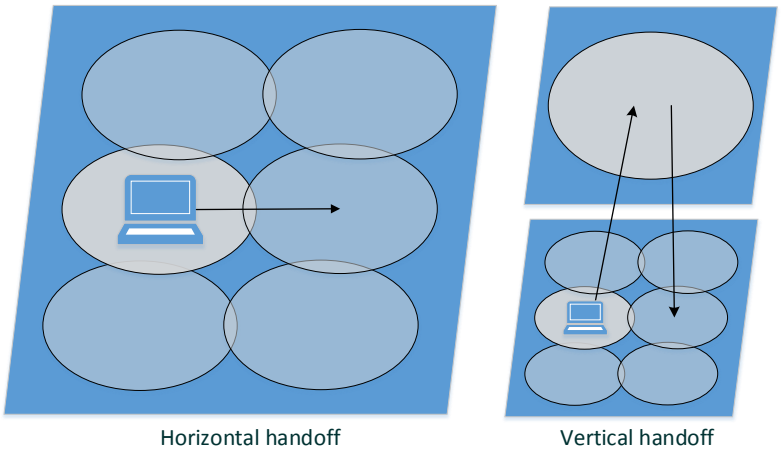


Figure 3.3: Horizontal and vertical handoff.

ideally like to be able to move between different network types without having to worry about having active services being interrupted. The following sections presents some solutions that focuses on solving this problem.

IEEE 802.21 and the Media Independent Handover standard

With mobile devices becoming increasingly popular the issue of maintaining data sessions across different networks became an increasingly larger problem. To aid against this, the IEEE created the IEEE 802.21 standard[75], whose goal was to provide a that could provide a media-independent framework in order to enable a seamless handover between different kinds of access networks. It shares some similarities to the IEEE 802.11u standard which allows for roaming between 802.11 networks and other networks, though the 802.11u standard does not support handover of ongoing IP sessions. The 802.21 standard provides information for use on handover between the 802.3, 802.11, 802.15, 802.16, 3GPP and 3GPP2 standards. The primary work of the 802.21 group is the Media Independent Handover framework[5].

The 802.21 standard does not attempt to standardise the handover execution mechanism itself[75], meaning that the Media Independent Handover standard can also be applicable to systems on different levels of the protocol stack, be it running Mobile IP at the IP layer, or SIP at the application layer. The Media Independent Handover standard defines a set of tools needed in order to exchange information, events, and commands to aid in handover initiation

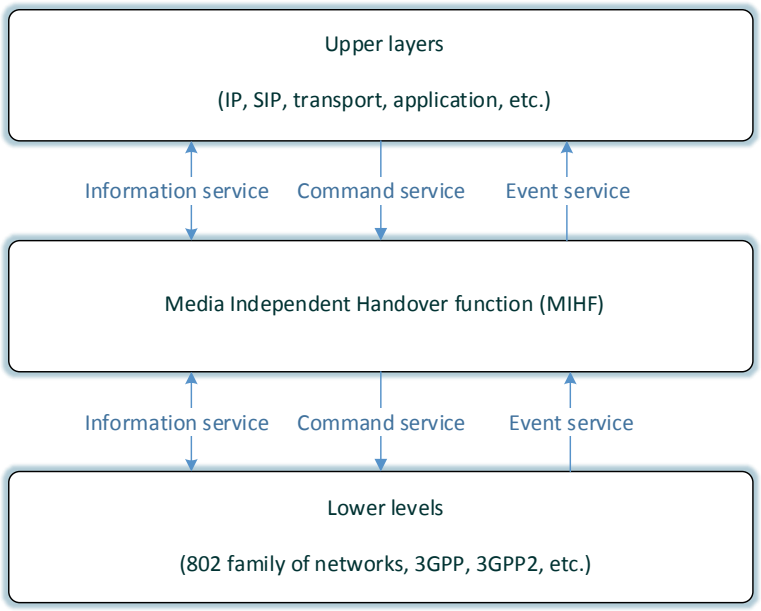


Figure 3.4: the architecture of Media Independent Handover.

and preparation.

Generic Access Network

Generic Access Network also known as Unlicensed Mobile Access (UMA), is a telecommunications system for improving data and voice applications by allowing network traffic to be sent over the internet instead of using an over-the-air technology like GSM. Generally, its most discussed feature is the systems ability to seamlessly transition between using cellular networks and Wi-Fi for traffic. Figure 3.5 shows how the architecture of how the Generic Access Network is organised. Using the Generic Access Protocol system devices have the ability to choose the currently best available method of passing traffic to the core mobile network, be it through an over-the-air service like GSM, or by using a secure IP connection to a Generic Access Network Controller over a Wi-Fi network.

The Generic Access Network system can be especially useful in places where cell phone coverage is unreliable or non-existent as providers can offer coverage through Wi-Fi, which is often cheaper and easier to set up than erecting

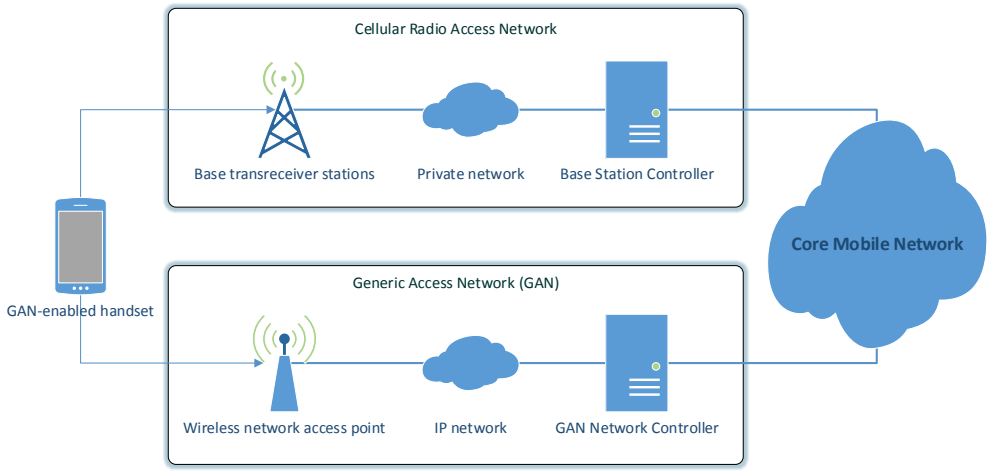


Figure 3.5: the architecture of the Generic Access Network system.

new cell towers.

Mobile IP

Developed by the Internet Engineering Task Force (IETF), Mobile IP is a communications protocol for allowing mobile devices to move between different networks while maintaining the same IP address through the use of indirection, offering users a seamless and continuous connection to the internet.

Devices, commonly referred to as Mobile nodes, are identified through an IP address aptly named the home address. This IP address is When connected to network other than the home network, known as a foreign network, the mobile node is associated with a so-called care-of address which is used to identify its current connection to the internet. At this point the home address is associated with what is referred to as a Home Agent (HA), a router on the home network that keeps track of the care-of and home address of mobile nodes. When a mobile node is connected to a foreign network the HA uses proxy Address Resolution Protocol (ARP) in order to receive datagrams from the correspondent host that the mobile node communicates with. These datagrams are then forwarded to the mobile node through IP tunnelling. When arriving at the foreign network, the Foreign Agent (FA) forwards the datagrams to the mobile node. The FA holds information about mobile nodes

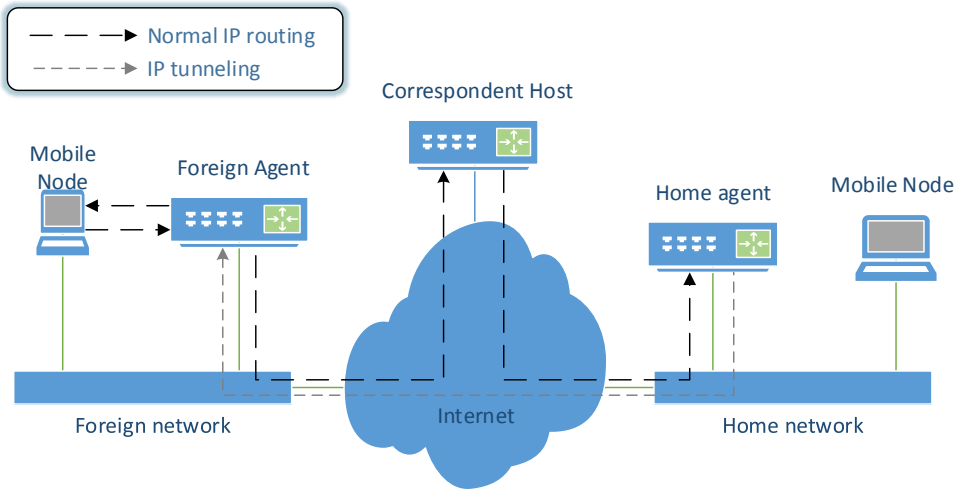


Figure 3.6: the structure of Mobile IP.

currently on the network and also advertises care-of addresses for mobile hosts. If a foreign network does not feature a FA, the mobile node has to get and advertise a care-f address by itself.

Figure 3.6 shows how communications are routed between the Mobile node, FA, Correspondent Host (CH), and HA when the Mobile Node is connected to a foreign network.

Whereas Mobile IP essentially treats all communication as if it is remote, others have taken the opposite approach. Contact Networking[25] treats all communications as if they are local with the goal of providing support for local connectivity equivalent to that provided by Mobile IP for remove connectivity.

Project Fi

Google recently unveiled Project Fi[5], a wireless subscription service which automatically and seamlessly switches between different mobile networks and

[5] `https://fi.google.com/about/`

Wi-Fi in order to provide the best available service at any given time. This is achieved through a cellular radio which supports all 4G LTE networks in the US, as well as many others. It allows for calls to be started on Wi-Fi, but will switch seamlessly to a cellular network if the connection is weakened or drops. Google does make any mention of how the system does this other than stating that it uses new technology. While not discussing any implementation details regarding the system, they mention that the system analyses network connections in order to always utilize the best network available as well as maintaining a list of open Wi-Fi hotspots that have been verified to be both fast and reliable.

Chapter 4

Materials and Methods

4.1 Materials used

The research conducted for this thesis was primarily done using hardware that was already used within the CallMeSmart system.

4.1.1 Phones

As the research did not require any additional functionality from the phones CallMeSmart is using, there was no need to change the phones currently being used in the CallMeSmart system. This meant that the system would continue to use the phones that it was currenly using. A complete list of hardware used is shown below.

- Samsung Galaxy SIII (S3) - Used as an outside-the-network phone.
- Samsung Galaxy SIV (S4), multiple - Used as CallMeSmart softphones and as GSM gateway phones.
- Nexus 5 - Used as a GSM gateway phone.
- Lenovo Thinkpad T440p - Laptop used to run the virtual machines.
- Broadcom Corp. Targus Bluetooth adapter BCM2035 - used to connect gateway phones to Asterisk.

- Broadcom Corp. Targus Bluetooth adapter BCM20702 - used to connect gateway phones to Asterisk.
- Plantronics BT 300 Bluetooth adapter - used to connect gateway phones to Asterisk.
- Jabra LINK 360 Bluetooth adapter - used to connect gateway phones to Asterisk.
- Subscriber Identity Module (SIM) cards - used for calls on the GSM network.

In order to allow the phones to communicate over the GSM network, the phones located outside of the Wi-Fi network had to be equipped with regular SIM cards.

4.2 Design

The initial idea for extending CallMeSmart to support extramural communication was to give the PBX a an external trunk which it could use in order to connect the IP based calls to users outside the hospital network. In order to simulate this a prototype setup was created. This setup used a set of mobile phones equipped with regular SIM cards, making them able to make calls over the GSM network. Asterisk features a module called chan_mobile which allows it to use a mobile phone as an Foreign eXchange Office (FXO) device over Bluetooth, which was used to connect the phones to Asterisk. The phones connected to Asterisk would then be used as gateways, bridging the internal IP based network to the GSM network. This setup is illustrated in figure 4.1. In addition, users should be able to select a GSM only mode where the phone will make calls directly to the intended callee rather than going through the system.

4.2.1 Maintaining contact lists

As phones are not tied to specific individuals the system needs a way of letting other users know that this user is now associated with this number. SIM cards stay with the phone so a user might have one number one day and another number the next day. Having someone else answer when you think

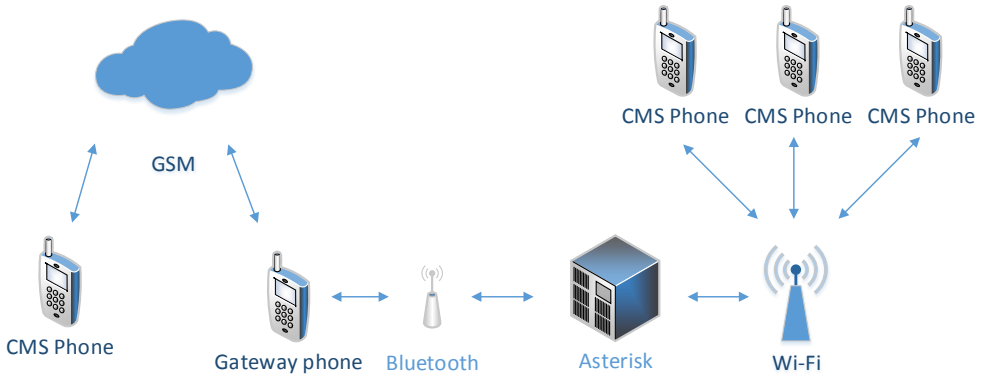


Figure 4.1: The architecture of the system using the GSM gateway.

you are calling a specific person is not a desired outcome. In order to avoid this the context-aware database contains a field for each user which holds their currently associated GSM phone number, as shown in figure 4.2. Note that this is not the same as their IP based number, the extension number, which remains the same across phones. when a user logs on to the system the server is informed of the users new GSM number and the context-aware database is updated accordingly as illustrated in figure 4.3a. If the user is using a DECT phone or a CallMeSmart softphone without a SIM card the GSM number is set to N/A in the database.

As contact list and other data are deleted locally on the phone every time a user logs off the system or the phone turns off, there is no need to check if a users contact lists have been updated when they log in as they will always download the most up-to-date version when logging on. As a result of this, the only scenario in which updating contact lists is applicable is in the case of users who are online when changes occur.

To ensure that users' contact lists are as up-to-date as possible, the server checks what contact lists contains a given number whenever a user logs in with a new GSM number. Naturally, this process is automated and doesn't require any input from the user carrying the phone. The process is shown in figure 4.3b and goes as follows:

1. A user logs on the system.

2. The server notes that the user has a new GSM number and updates

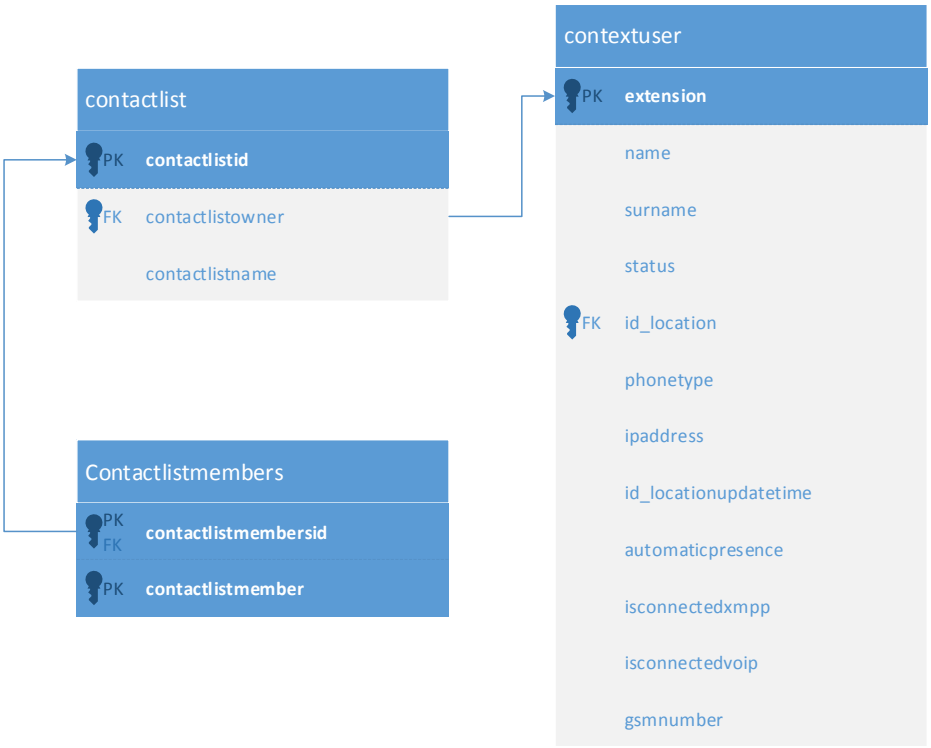


Figure 4.2: The contact list database schema.

the database.

3. The server looks up which contact lists contain the user which just logged on. If the owner of a list is online, the server notifies them that their contact list is outdated.

4. The CallMeSmart softphone, having received the notification, request that the server send it the updated contact info.

5. The server fetches the requested information from the database.

6. The server sends the requested data to the phone.

Currently the system is set up in such a way that if a user loses contact with the network for twenty seconds they are automatically logged out of the system and are required to sign in again when they can contact the network again. When discussing support for GSM calls so that users outside of the hospital can be reached it it clear that this setting would cause problems. Instead of simply removing this feature the system might require that users that stay outside the range of the hospital network for extended periods of time, say every few hours, reauthenticate themselves to show that they are still online. This process should of course be as non intrusive as possible as to not become an annoyance to the users. A suggested solution has been to allow users to scan their ID badges in order to generate a dynamic passcode which can be used in combination with their standard password in order to authenticate them when logging in and reauthenticating. If a user isn't in range of the hospital network this presents a problem in that the phone would need a secure way of contacting the server in order for the authentication to be approved.

The fact that user might be considered online even when they are not reachable through the hospital network presents a set of problems that need to be solved. Primarily, the following problems need to be addressed:

- The system needs a way of knowing whether users are still online if they stay outside the range of the hospital network for an extended period of time.

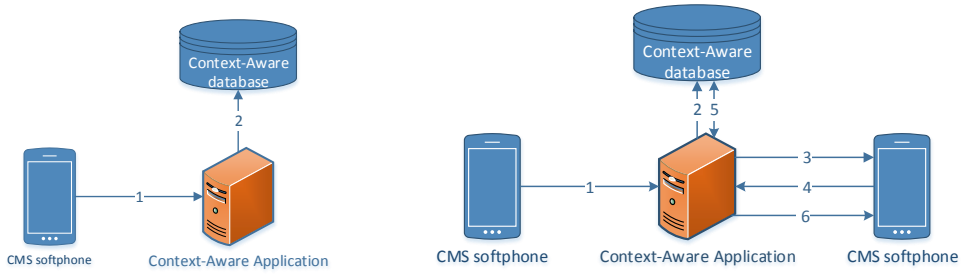
- When not able to connect to the server users can end up with outdated contact information as other users might change phones and GSM numbers.

The issue of the server knowing whether users are available or not could be solved through the use of reauthentication as mentioned earlier. If a user fails to reauthenticate within a given period of time they are considered logged out from the systems point of view. This would require that the system implements a way of contacting the system from outside the hospital network, be it through Virtual private network (VPN) tunneling or another solution. A secure VPN tunneling solution would also be able to solve the problem of updating contact information for users outside of the hospital network.

Another thing to note is that the default contact list which all users have access to contains every user in the system. This means that any time a user logs in with a new number, every online user will need to have the default contact list updated. While this isn't too big an issue at the current time as there are roughly a hundred users in total using the system with an average of ten users online at a time, it is important to consider how this solution would scale if the system was expanded to be used in the entire hospital rather than just a part of the oncology department. Updating the contact list for ten users is a lot less demanding than updating contact lists for potentially thousands of users. In order to handle a large amount of users one of the more obvious ways of making sure that the server isn't swamped with request would be to take a snapshot of all the online users when a number was updated and then sending out updates in waves. This method of handling updates would also allow the server to adapt how many simultaneous updates were going out depending on the status of the network. If the network is experiencing heavy traffic, the number of simultaneous requests could be reduced. Another approach might be to have the system analyse call patterns and prioritize updating the contact lists of users who have called the updated user before as well as their role. While the chance of a random contact in a list of potentially thousands of users being outdated might be small this is still an important point to keep in min. A simple solution could be to check call logs for calls where the user with the updated number was the callee and then prioritising updating the contact lists of the most recent callers.

4.2.2 GSM Gateway

In order to allow users of the CallMeSmart system to make calls to users that are in an area which is not covered by the CallMeSmart network, a GSM gateway is used, acting as an external trunk. This gateway consists of a regular Android smartphone which is connected to the Asterisk server through the



(a) Updating the database. (b) Updating the contact list.

Figure 4.3: Contact list updating.

use of a common-off-the-shelf Bluetooth adapter and the chan_mobile[1] module. Chan_mobile allows Asterisk to connect and use a Bluetooth enabled phone as an FXO device. This connection is made by defining an adapter and a mobile device in the Asterisk configuration files. An example of such a config can be seen in listing 4.1. The configuration shown in this listing defines two adapters, broadcom-20702 and broadcom-2035, as well as the two mobile devices nexus-5 and galaxy-s4. In addition to their MAC addresses, mobile devices also define what adapters they will connect to, the port number it uses, the adapter it connects to, its channel group, as well as their calling context. The context is used by Asterisk when routing incoming calls to the devices.

```
; Adapters

[adapter]
id=broadcom-20702
address=A0:B1:C2:D3:E4:F5

[adapter]
id=broadcom-2035
address=A6:B7:C8:D9:EA:FB

; Devices

[nexus-5]
address=AC:BD:CE:DF:E0:F1
port=3
```

[1] https://wiki.asterisk.org/wiki/display/AST/Using+chan_mobile

Table 4.1: The special characters used in dial plan patterns.

Character	Description
X	Matches any number 0-9.
Z	Matches any number 1-9.
N	Matches any number 2-9.
[1-9]	Matches any digit in the brackets. In this case 1 through 9. Additionally patterns like [145-7] can also be used, which would match with any digit that is either 1, 4, 5, 6, or 7.
[a-z]	Matches any lower case character. Can also create patterns like with digits.
[A-Z]	Matches any upper case character. Can also create patterns like with digits.
.	Matches one or more characters.
!	Matches zero or more characters.

```
adapter=broadcom-20702
context=gateway
group=1

[galaxy-s4]
address=A2:B3:C4:D5:E6:F7
port=3
adapter=broadcom-2035
context=gateway
group=1
```

Listing 4.1: A example chan_mobile config.

Asterisk handles calls through the use of a dialplan. The dialplan consists of one or more contexts, which are collections of extensions. Extensions are how Asterisk handles calls within contexts and can be either a literal or a pattern. Literals are static numbers like **113** or **90166447** but can also consist of the * and # symbols. Asterisk also allows the use of alphabetical characters in extensions, so extensions like **office5** are possible. Patterns are evaluations which allow for dynamic routing of calls. In addition to numbers and letters, is has a set of special characters which can be used to represent one or more characters. These characters are presented in table 4.1.

Listing 4.2 shows the syntax of extensions. The initial **exten** => part is used to inform Asterisk that the next thing it sees will be a command. The

first parameter is the extension, which is either a number or a pattern. When asterisk receives a call from someone trying to reach a number, it tries to find an extension which matches that of the called number. An extension can also be tailored to perform specific actions based on the number of the caller. The second parameter is the priority of the extension, which is the order in which extensions are performed. Asterisk requires that a context always contains one or more extension with the value **1** to determine where to start. Priorities are either an integer value or the value **n**, which automatically increments the priority of the extension. The final parameter is the command to execute when this extension is called. Examples of these commands are actions like answering the call, dialling a number to forward a call, hanging up the call, or putting the user on hold.

```
exten => extension, priority, command
```

Listing 4.2: The Asterisk extension syntax.

As mentioned earlier, Asterisk routes calls using literals and patterns. As such, it seems natural that there could be cases where more than one extension would match a given number. In order to solve this problem Asterisk has a set of sorting rules[2] that decides the order in which extensions are used. These rules are presented in the list below.

1. The dash character (**-**) is not considered when matching or sorting extensions and is ignored except when used to specify a range in either a digit or character set.

2. Non-pattern extensions, e.g. extensions that do not contain any of the **X**, **Z**, **N**, or a bracket range, are sorted in ASCII sort order before patterns.

3. Patterns are sorted by the most constrained character set per digit first. This means that a pattern with fewer possible matches will sort before another pattern that can be matched to a larger number of extensions. An example of this can be made using the **X** and **Z** characters. **X** can be matched to either digit 0-9, while **Z** matches any digit 1-9. As such, **X** Z sorts before **X**.

[2] `https://wiki.asterisk.org/wiki/display/AST/Pattern+Matching`

4. Character sets that contain the same number of characters are treated as strings containing their respective characters, and then sorted in ASCII sort order. As an example, **X** matches the ten digits **0-9** and **[a-j]** matches the ten characters **a-j**. As digits sort before characters, **X** sorts before **[a-j]**. This sort ordering is important if the character sets overlap as with **[0-4]** and **[4-8]**.

5. The period (**.**) wildcard sorts after character sets.

6. The exclamation mark (**!**) wildcard sorts after the period wildcard.

An example dialplan which consists of two contexts, *NST_ context* and *example_ context*, is shown in listing 4.3. NST_context utilizes only a single extension, **_X.**. Using the information in table 4.1 shows that this extension will be matched to any called number that begins with a digit and has at least one more character. Any calls matched to this extension are forwarded through the use of AGI. This is the default dialplan for CallMeSmart. All calls that matches this pattern are forwarded to the Context-Aware Application which lets the context-aware features of the system handle the call.

```
[NST_context]
exten => _X.,1,Agi(agi://192.168.32.131/context.agi)

[example_context]
exten => 2005,1,Answer
exten => 2005,2,SetMusicOnHold(default)
exten => 2005,3,WaitMusicOnHold(20)
exten => 2005,4,Hangup

[example_context_two]
exten => _9X.,1,Dial(Mobile/g1/\${EXTEN:1},45)
exten => _9X.,n,Hangup
```

Listing 4.3: An example dial plan.

example_ context shows an example context that might be used in order to stress test the Asterisk server. It matches to a single extension number, that being 2005 in this case. when receiving a call on this extension it sets the waiting music for that call to be a default sound clip and then plays twenty seconds of that sound clip before hanging up the call.

example_ context_ two shows how the system might use a phone as an external trunk. In this case the dialplan matches any calls starting with the number 9 and dials the called extension using chan_mobile. The *Mobile/g1/$EXTEN:1* parameter tells Asterisk to use the first free phone connected that belongs to group one. In the case of the configuration illustrated earlier in listing 4.1, this means that Asterisk would have two phones to choose from provided that both are connected. Using channel groups is useful as it allows the system to handle what devices are available itself, as each connected phone can only be used for one call at a time. This means that the system needs an addition phone connected for each simultaneous GSM call it is to support. As *chan_ mobile* does not support multiple simultaneous connections over a single adapter the system also needs one adapter for each phone that is to be connected.

4.2.3 Call handling

While both the caller and the callee are connected to the hospital network calls are routed as normal. A caller calls an extension and the server connects them to the callee at the other end as long as they are considered available. With the addition of GSM calls a few check need to be put in place in order to handle possible scenarios correctly. Ideally, these checks should be as invisible to the users as possible as they should be able to use the system as if nothing has happened. To better illustrate this, one can use a scenario similar to the one described in section 1.4 where a user is moving in between having Wi-Fi coverage and not. As users can now be reached while not connected to the hospital network, there are now three additional scenarios that can occur when a user makes a call. These scenarios are described below and the call path for each of them are also visualized in figure 4.4. Furthermore, the call handling logic is shown in the flowchart of figure 4.5.

- A user not connected to the network calls a user connected to the network.
- A user connected to the network calls a user not connected to the network.
- A user not connected to the network calls a user not connected to the network.

In the first scenario a user finds himself outside the range of the hospital walls and as a result doesn't manage to connect to the hospital network. As the user makes a call the CallMeSmart phone notes that it does not have a connection to the server, and calls one of the gateway phones as it cannot make a SIP call as it normally would. Since Android currently does not allow applications to send Dual-tone multi-frequency signaling (DTMF) tones over an ongoing call[3], the extension of the callee is sent as part of the initial call so that the callee extension is dialled when the gateway phone answers the incoming call. Upon noting that one of the connected phones is receiving a call the Asterisk server takes action accordingly. If the number calling is not a recognised number the call is declined, while a recognised number is answered. Upon the call being answered, the caller's phone sends the desired callee extension using DTMF. The server, upon seeing that the callee is online and connected to the server forwards the call using SIP.

In the second scenario the caller makes a SIP call to the server which then determines that the callee is not reachable through the hospital network. As the callee is still listed as online to the system the server attempts to reach the caller over the GSM network through the use of the external trunk.

The third scenario initially plays out similarly to the first scenario where only the caller is outside the range of the hospital network. The two scenarios branch off when the Contex-Aware Server receives the call and notices that the callee is not reachable through the hospital network. The server then dials the callee using one of the available gateway phones.

In addition to allowing users to call each other when they are outside the range of the network, GSM calls can also be used in order to continue call happening over Wi-Fi if one end of the call moves outside the range of the hospital network. If the server loses contact with either end during a call without receiving a hangup signal, the server establishes a new SIP call to the user still on the internal network and calls up the external user using one of the gateway phones. The idea behind this is to reduce the amount of work needed from the users in order to continue communicating.

The possibility of using the GSM network in order to communicate also opens up the ability for the system to have a potential fallback option in case a situation was to arise that would prevent the system from working as normal. In situations similar to the ones mentioned in section 1.3 ([2, 4]), the ability for users to switch their phones to a GSM-only mode would allow

[3] `https://code.google.com/p/android/issues/detail?id=1428`

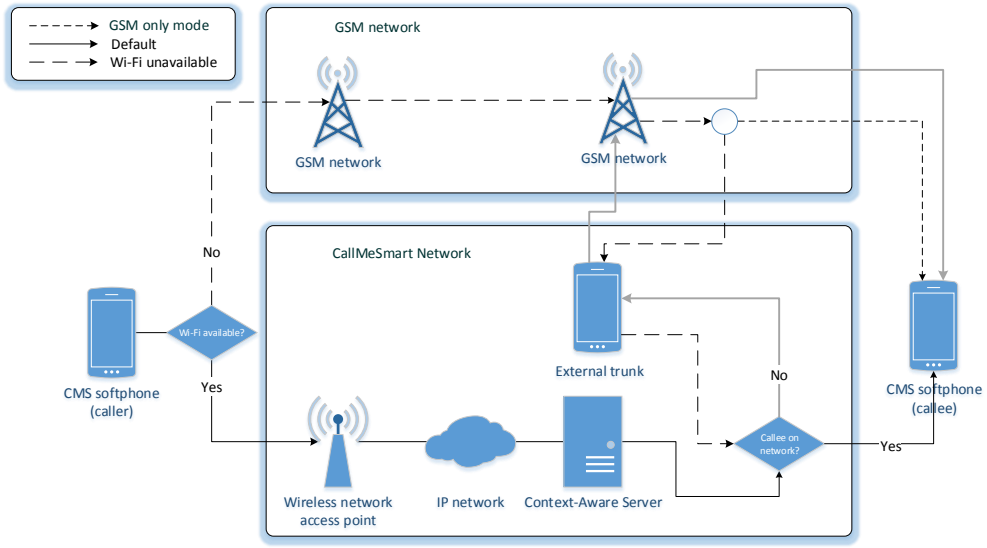


Figure 4.4: The CallMeSmart call path.

them to continue to communicate as usual without having to change what devices to use, reducing the impact of such problems.

4.2.4 Security

As this thesis is not looking at communication beyond the closed hospital network aside from GSM call functionality, additional security measures are not really needed. If communication in the form of messaging, IP based calls or alarms was to be extended to external networks this issue would need to be addresses however. At this time GSM is considered secure enough for the kind of communication envisioned for this thesis, that being voice calls, though it should be noted that it does have weaknesses that can be exploited[16, 53, 33].

It can be noted that while GSM has weaknesses, network providers in Norway are amongst the best when it comes to securing their GSM networks according to Security Research Labs' GSM Map[7], an online service which uses volunteer data in order to analyse the security of mobile networks around the world. While many providers still use the weaker[16, 53] A5/1 stream cipher to encrypt traffic on their GSM networks all the major network providers in Norway use the more secure but still vulnerable[33] A5/3 block

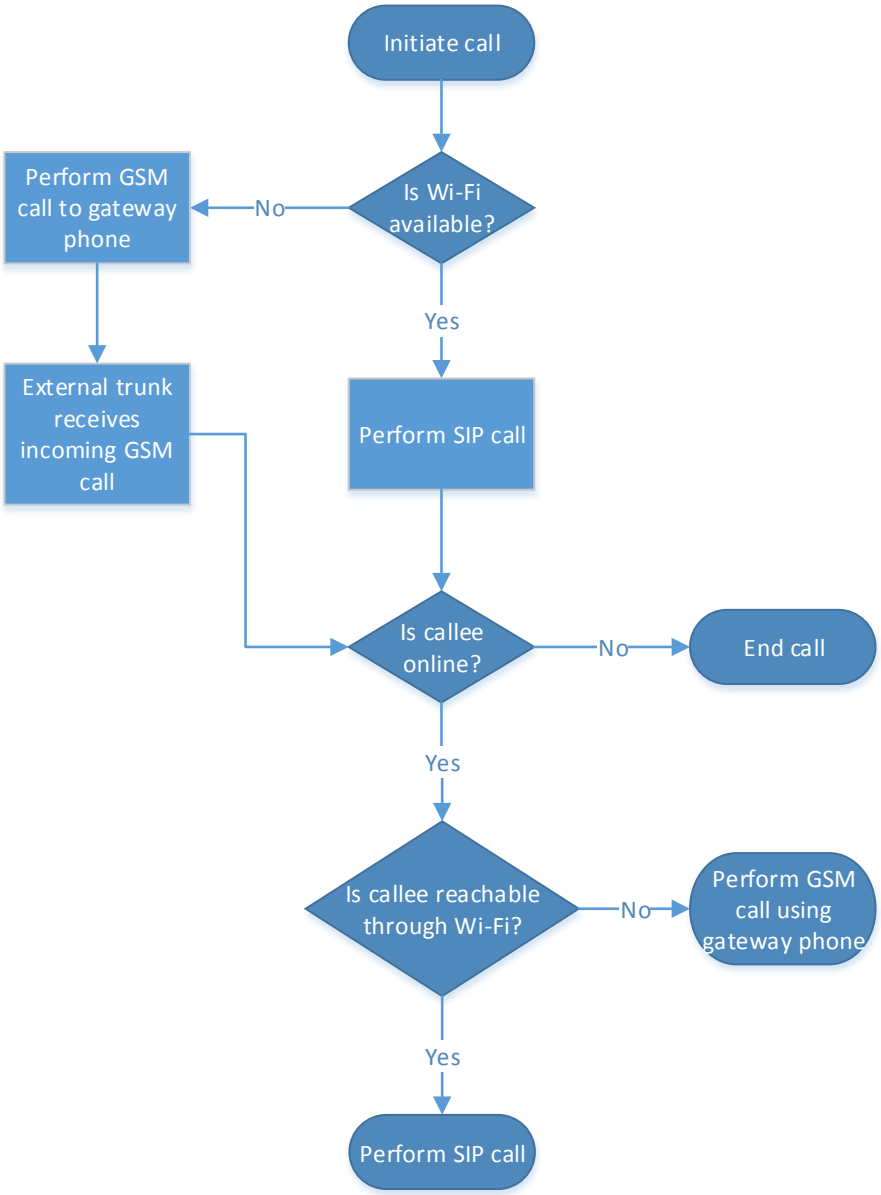


Figure 4.5: Call logic.

```
root@cmstest:~# ./newscript.sh
###################################################
#      Install/deploy script for CallMeSMart      #
###################################################
# Platform: Debian 7 Wheezy
# Version: 0.1
# Created by: Alain Giordanengo
###################################################
1) Configure Network  3) Install Asterisk   5) Install Glassfish
2) Install OpenLDAP   4) Install Openfire   6) Quit
Choose an option (Press [ENTER] to display them again):
```

Figure 4.6: The install script menu.

cipher for their 3G networks. In addition they also use A5/3 for their 2G networks as long as phones can support it.

4.2.5 Context-aware Server migration

The context-aware server which controlled the context-aware database, context-aware application, OpenFire client, and administration panel was initially running on a Windows 2008 Server. Work had been started towards moving the context-aware server to run on an instance of Debian 7, though this had not been completed as the context-aware database and the context-aware application had not yet been ported over. To help with simplify the process and future installations an install script had been started. This script allowed an administrator to easily configure and install the various parts of the context-aware server with a goal of reducing set up time and potential errors from incorrect settings. The inital options available in the script can be seen in figure 4.6. For this thesis, the install script was improved by implementing the remaining installations of the context-aware application and the context-aware database. With moving these server modules from the initial Windows Server, some of the software versions used were also updated. Most notably, the new Linux server uses Java development Kit (JDK) 8 instead of JDK 7 and the context-aware application now runs on Glassfish 4 instead of Glassfish 3. Updating to JDK 8 is good as Java 7 reached its End-of-life (EOL) at the start of April[1].

4.3 Testing

Due to the importance of the work they support, it is imperative that communication systems used in hospitals are thoroughly tested in order to ensure that they are as stable and reliable as possible.

The following sections present some aspects of the CallMeSmart system and details how they were tested. The results from these tests can be found in chapter 5.

4.3.1 Bluetooth adapters

4.3.2 Pairing

In order for the Asterisk server to connect to the phones that were to be used as an external trunk, the phones first had to be paired with the Bluetooth adapters connected to the computer. There are several ways of paring Bluetooth devices to adapters depending on the situation. One might create a Bluetooth agent to aid in the process, or if the adapter is broadcasting itself the pairing can be initialised from the phone. If paring a device that is being used in a virtual machine the paring is sometimes done through the host machine, as the host machine handles the adapter and passes data to and from the virtual machine and the adapter.

When pairing bluetooth devices the user is generally presented with a pairing request. Simple devices such as wireless speakers and headsets often don't present this request to the user, but it is standard when pairing devices such as laptops and smartphones, e.g. devices with feature a display. This request is often presented as a dialog showing a randomly generated number on each device. The user can check that they are connecting to the correct device by seeing that the number is the same on both devices. Another popular way of doing this is having the user enter a passcode when making the pairing attempt. The user then repeats the passcode on the other device. The purpose of this dialog is to allow the user to ensure that they are connecting to the correct device. For some adapters when using a virtual machine, like integrated adapters in laptops, this confirmation dialog will show up on the host machine even if the virtual machine is the one to initiate the pairing request. For the virtual machine it can look like the pairing succeeded without needing confirmation, or it might seem like the request failed though

it actually succeeded.

4.3.3 Adapters

Broadcom Targus BCM2035 Bluetooth adapter

Initial testing with setting up a gateway phone was done with The Broadcom Targus BCM2035 Bluetooth adapter, a common-off-the-shelf Bluetooth 1.2 adapter which the CallMeSmart group had lying around. It can be seen in figure 4.7a.

Intel Wireless Bluetooth 4.0 Adapter

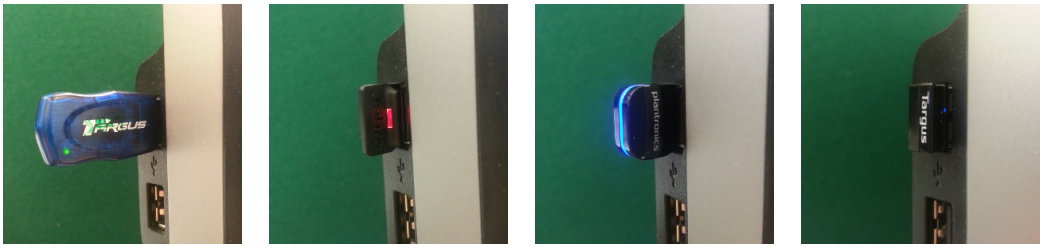
After the problems with the Targus adapter, the next attempt was to use the integrated Bluetooth adapter of the laptop running the virtual machines. The laptop used for this was a Lenovo Thinkpad T440p which features an Intel Wireless Bluetooth 4.0 Adapter. As the name suggests, this adapter is compatible with the Bluetooth 4.0 protocol, meaning that it offers more functionality than the Targus BCM2035 which only supported Bluetooth 1.2.

Unusable adapters

While the previously mentioned adapters were all able to be used to pair phones with Asterisk, this was not the case for all the adapters that were tested for this thesis. Many products like wireless headsets often come with their own dedicated bluetooth receivers which are difficult to use for other products, often due to specialized drivers. This was the case with two of the adapters that were tried, that being the Jabra LINK 360 and the Plantronics BT300 as illustrated in figures 4.7b and 4.7c respectively.

Broadcom Targus Bluetooth adapter BCM20702

After the problems encountered with the inital Targus adapter, the connection issues of the integrated adapter, as well as the unusable adapters, the last adapter tested was the Broadcom Targus Bluetooth adapter BCM20702, a Bluetooth 4.0 compliant adapter, shown in figure 4.7d.



(a) Targus (b) Jabra (c) Plantronics (d) Targus

Figure 4.7: Bluetooth adapters.

4.3.4 Call testing

In order to test the call quality and scalability of the system, call tests were performed using the freeware version of StarTrinity SIP Tester[4], a tool for monitoring and testing VoIP systems. For this purpose, StarTrinity was used in order to generate a number of rapidly successive calls directed towards the Asterisk server while StarTrinity monitored the calls in order to measure variables such as call response time, the amount of call jitter, as well as the amount of packet losses that occurred during calls. The free version of StarTrinity has limitations in that it only allows for fifty simultaneous call and a limit of one hundred and fifty generated calls per session.

During these tests the Asterisk server was set up to forward all calls to the Context-Aware Server over AGI, which would then handle the calls. there were two different configurations for how the Context-Aware Server was set up to answer these calls; one where it would emulate a SIP call, and one where it would use the external trunk. To emulate SIP calls, the server was set to answer calls, play twenty seconds of music, and then hang up the call. When testing using the external trunk the server would answer a call, dial out to a normal GSM phone using the external trunk, keep this call open for twenty seconds after the call was answered, and then hang up the call.

[4] `http://startrinity.com/VoIP/SipTester/SipTester.aspx`

Chapter 5

Results

5.1 Call testing

5.1.1 Adapters

Broadcom Targus BCM2035 Bluetooth adapter

The server had some problems recognising the Bluetooth adapter when plugged into the machine, though this was generally solved by ejecting and plugging in back in the machine repeatedly until it the system registered it. The Broadcom Targus Bluetooth adapter did not have any problems connecting a phone to Asterisk once the pairing between the phone and the adapter had been made. Asterisk was able to notice when the phone made and received calls, and could take control, e.g. answer a call and forward it to an IP based phone, or forward a call from an IP based phone to a GSM phone through the gateway phone. It suffered from a problem where the callee was unable to hear what the caller was saying as no audio seemed to be sent from the caller. The adapter also had problems where it would flood the console with warnings due to corrupted Synchronous connection-oriented (SCO) packets.

Intel Wireless Bluetooth 4.0 Adapter

The virtual machine did not have any problems recognising this adapter, but the Asterisk server had problems connecting to phones despite the fact

that the pairing process was successful. This problem was consistent across multiple phones, though it seemed to occur more often when attempting to connect to the Nexus 5. In addition, even when connected the Asterisk server had problems identifying what happened on the phone. While Asterisk could create an outgoing call or answer an incoming call on the phone, once a call was answered the server seemed to disconnect from the phone, no longer recognising what was happening on the phone. While this could be solved by selecting to reconnect to the adapter on the phone the same issue as with the previous adapter occurred in that the callee did not receive any audio from the caller. If the gateway phone was disconnected from the Asterisk server the callee was able to hear the caller, though at this point it was just a normal GSM call between the two phones.

Broadcom Targus Bluetooth adapter BCM20702

Unlike the inital Targus adapter the BCM20702 did not have any problems with corrupted SCO packets, but like the other adapters had a problem where the callee would not receive any audio from the caller. In one case the host machine running the virtual machine used for testing received a Plug and Play fatal error Blue Screen of Death (BSOD) as a result of plugging in the BCM20702 adapter. After rebooting the machine, the server would no longer recognise the BCM20702 adapter when plugged in. This was solved by uninstalling and reinstalling the adapter drivers on the host machine running the virtual machine hosting the Asterisk server, though this lead to neither the inital Targus adapter or the integrated Bluetooth adapter being recognised as long as the BCM20702 adapter was plugged into the machine. The adapter would also begin to change what MAC address it presented to the system when plugged in. This is assumed to be working as intended. The Bluetooth standard permits up to seven devices to connect to a so called master Bluetooth device, which was also the number of different addresses presented by the adapter. It would seem that reinstalling the adapter drivers caused the adapter to choose one of its available MAC addresses at random to be presented to the system.

While managing to pair with devices without any problems before the BSOD, it began to have issues after reinstalling the adapter drivers. While the devices were able to pair with the adapter, the Asterisk server was no longer able to connect to any devices using the adapter.

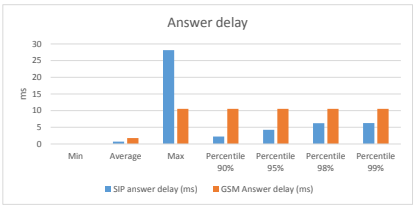
5.1.2 Call quality

As detailed in section 4.3.4, tests were run using the freeware licence of StarTrinity SIP Tester.

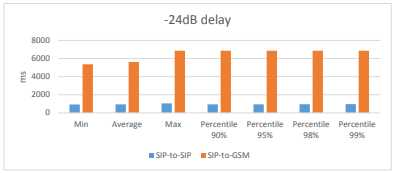
As each session of testing could only generate 150 calls, the same test was used several times in order to generate a bigger sample size to draw from. As only 50 simultaneous calls were able to be generated with the freeware version of StarTrinity, each session was performed in three parts in order to always make sure that the generated call were performed in close succession. In total, 1500 calls were generated when looking to see how multiple simultaneous calls affected the SIP call quality.

Due to the multitude of problems encountered with adapters not pairing correctly and Asterisk failing to connect to phones that had been paired with adapters, only one phone was able to be connected to the system when performing testing. As a result, no testing could be done in regards to seeing how Asterisk would handle having multiple external lines at the same time. This lead to using StarTrinity in order to generate a single call at a time, and manually answering it on the phone called by the gateway phone. In total, twenty calls where generated and measured when using the external trunk. Luckily, the problems with one end of the call not receiving any audio affected the callee, meaning that StarTrinity would still be able to measure call quality as it would be the calling party. Callee call quality is assumed to be equal to that experienced by the caller. While only one adapter was used at a time, all adapters that were able to be paired with a phone and connected to Asterisk were used for testing. There did not seem to be any notable difference between using the different adapters. In no cases, either SIP testing or GSM testing, did a call fail to be answered.

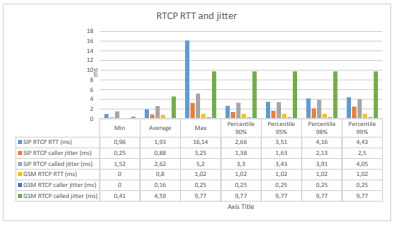
The results from these tests are presented in figure 5.1. Figure 5.1a shows the answer delay, which is the time from sending an *invite* request and receiving a response, which is not the same as the call being answered. Figure 5.1b shows the time from sending the *invite* request to receiving an audio signal which is greater than -24dB, effectively measuring the time from making the call until the call is answered. Figure 5.1c shows the Round-Trip Time (RTT) and jitter of Real-time Transport Control Protocol (RTCP) packets. Figure 5.1d shows the packet loss measured during calls.



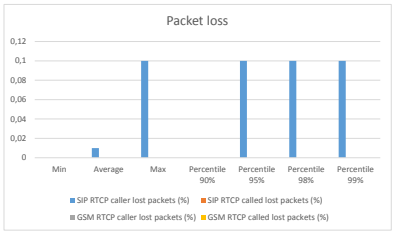
(a) Call answer delay.



(b) Time from sending invite request to receiving audio signal.



(c) RTCP RTT and jitter measurements.



(d) Caller packet losses.

Figure 5.1: StarTrinity test results.

5.2 Server migration

While there has been limited testing done so far as to ensure that all functionality is still in place and working after moving the Context-Aware Server from the Windows server to running on Debian 7, initial impressions are that things seem to be working fine so far. Throughout testing the other parts of the system there did not appear to be any errors that occurred as a result of moving the context-aware server to Debian.

Chapter 6

Discussion

6.1 Results

6.1.1 Adapters

Throughout testing, the most consistent problem to occur was issues with the Bluetooth adapters. While working fine one day, the adapters would suddenly no longer manage to bridge the Asterisk server and the phones the next day. Sometimes an adapter would pair with phones without any issues, other times not at all. This caused a lot of time to be used trying to figure out what was going on. Initial suspicions were that some of the adapters were not compatible with chan_mobile, though this was difficult to verify. While a unofficial list of compatible adapters and phones exists[1], it is far from complete and it would seem it is no longer being updated. In addition to the test performed and listed in chapter 5, the adapters and their pairings with devices were also tested by transferring files and playing simple audio clips over the Bluetooth connection. As these actions were performed without any issues, it would seem like the drivers were operating as intended. With adapters eventually able to connect devices to the Asterisk server it would seem that the problems experienced were not due to incompatibility with Asterisk.

Seeing how pairings were able to be performed successfully and connections to Asterisk being successful as well, the culprit of the aforementioned prob-

[1] `http://www.voip-info.org/wiki/view/chan_mobile`

lems is suspected to be a combination of the *chan_mobile* driver and its interaction with the adapters, as some of the adapters where more difficult to pair than others. This matches with the observations that the initial Broadcom Targus BCM2035 adapter had an easier time connecting devices to the Asterisk server than the integrated adapter and the Broadcom Targus BCM20702 adapter. while the two latter adapters often had trouble pairing and connecting to devices, the inital Targus BCM2035 adapter was instead troubled with corrupted SCO packets and difficulties detecting the adapter on the machine.

As mentioned in the previous chapter, while all adapters had their own problems in regards to paring and connecting to phones and the Asterisk server they all shared a more pressing issue, that being that the callee would not receive any audio from the caller side. As each of the adapters were able to connected to the Asterisk server and Asterisk could manage incoming and outgoing call, as well as other operations working as intended with the adapters, this problem was assumed not to be a result of drivers or *chan_mobile* adapter or phone incompatibility. Instead, it was suspected of being a incorrect configuration of the server, be it the Asterisk client itself or a Bluetooth setting. Despite repeated configuration changes this problem remained unsolved, and the actual cause was not found.

6.1.2 External trunk solution

The setup used shows that a setup which can switch between using IP based calls and GSM is possible, though the implemented solution is far from an optimal solution. The reality is that needing an additional Bluetooth adapter and mobile phone for each simultaneous GSM call is not a viable solution in the long run. Problems with conflicting adapters, where using one adapter might cause another to not be registered combined with the numerous problems encountered in regards to pairing and connecting adapters and phones it becomes obvious that another solution will be needed if this new functionality was to be implemented. Additionally, each gateway phone would ideally need to be connected to a charger, increasing either the number of USB devices connected to the server or the number of power sockets needed to ensure that they would not be unavailable due to lack of power. Figure 6.1 show the setup used while testing, featuring two adapter connected to the left side of the laptop, each connected to one of the phones lying in front. The final phone was used to either receive or send call.

While a small department like the oncology department where CallMeSmart is currently being tested might be able to get by using a solution similar to this, the number of adapters and phones needed for anything bigger than that quickly grows beyond what is reasonable. Using UNN as an example, which currently has more than six thousand employees, it should be obvious that the number of gateway phones needed would be difficult to properly manage.

A more realistic way of handling the external trunk might be to connect the Asterisk server to the existing PBX used by the hospital, thereby receiving a set of external lines through it. Another solution might be to set up a set of external lines for the Asterisk server. In other words, connecting the Asterisk server directly to the external telecommunications network. As the external trunk is related to the Asterisk server, changing the solution used only requires that the Asterisk server and the AGI be adjusted while the rest of the system remains unaffected.

Section 4.2.3 mentions that in the case that both the caller and the callee is located outside of the internal hospital network that the system will be using two lines of the external trunk. While this uses unnecessary resources, taking up two outgoing lines for a single call, as the caller could just contact the callee directly, it can be used in order to prevent issues related to inconsistent contact information. If a user is located outside the hospital they do not know if their current contact information is correct as a user might have had trouble with their device and swapped it out for another one. By calling the external trunk with the callee's SIP extension, the system can ensure that the call goes out to the correct user. Additionally, the system can maintain logs of calls as well as the current status of the users, even when they are outside the hospital. While this can all be useful, it only serves as a band-aid for the biggest problem related to letting users communicate when outside the hospital network; the system has no way of communicating and letting users know that their contact information is outdated, nor does it have any way of receiving updated information regarding the availability of the user in order to adapt their availability in the system.

Adding to the issues of using an external trunk is the fact that the callee cannot see who is contacting them. While their phone can note that the incoming call is coming from the external trunk of the CallMeSmart system, the callee is none the wiser about exactly who is calling them. Without being able to contact users outside of the network the system loses some of its ability to adapt to the users' availability. While still able to infer information from

Figure 6.1: The GSM gateway setup.

schedules and calendars, the location of users as well as manual updates to availability settings are not available to the system. Details like these are important to handle as they can quickly pile up and frustrate users. Given the amount of systems that fail due to staff resistance[13] it is important to ensure that users experience as few problems and frustrations as possible.

6.1.3 Quality of Service

It should be noted that the machine hosting the testing tool used to gather test results was the same machine used to host the virtual machines running the various parts of the CallMeSmart system. This could have an impact on the measured test values as network delay might be lower than it typically would be for devices normally communicating with the server. While probably not substantial, the test results might have been slightly higher if the testing tool had been hosted on another machine.

Figure 5.1a shows that the initial answer response is affected minimally by multiple simultaneously incoming calls.

Figure 5.1b verifies what most people would assume, that the time from placing a call until the call is answered is much faster for IP based calls than that of GSM calls. This it not surprising as GSM calls take a lot longer to initialize. While undesirable there isn't anything to do about this.

Looking at the results shown in figure 5.1c shows that call jitter was consistently low, with a worst case of just below ten milliseconds, which is generally considered well within the tolerance range before it becomes noticeable.

Figure 5.1d shows that packet loss is essentially non-existent. It should be noted that this would likely not be the case during actual use as users move around the hospital, moving between varying strengths of signal strength.

Looking at the results presented gives the impression that the quality of calls, both in terms of latency and audio quality, is of good quality. It also shows that having the system forward calls using the external trunk does not have any notable effect on the quality of call. While the initial waiting for the call to initialize is considerably higher for GSM calls there isn't really anything that can be done about this apart from moving to using a VoIP service through external networks as well.

6.1.4 System capabilities

The test performed show that the system does not have a problem with managing a high number of incoming calls over a short period of time. While the test performed did not have the proper numbers in order to properly simulate a large organisation, say a hospital in a major city, CallMeSmart has been stress tested in earlier iterations, with results showing that was able to support up to around two thousand users while running on a laptop.

6.1.5 Security

As previously mentioned several times, the security measures currently in place for the CallMeSmart system meet the requirements for the kind of solution implemented in this thesis. That said, the ability for devices to communicate with the system from external networks is very desirable, to the point where it is unlikely that GSM support would be added to the system without it. In this case the security measures in place would not be up to par, as authentication requirements would need to be improved.

6.1.6 Server migration

As mentioned in the previous chapter, there haven't been any direct testing done to ensure that the migrated server is working correctly, though the fact that the other parts of the CallMeSmart system that interacts with the Context-Aware Server did not encounter any problems during their interactions with the Context-Aware Server seems to indicate that the server is indeed working as intended, though there might still be undiscovered errors.

Chapter 7

Conclusion

7.1 Future works

The solution presented in this thesis presents an interesting addition to the CallMeSmart system. Giving users a way to use CallMeSmart in order to communicate beyond the reach of the hospital's wireless can be a valuable addition to the existing system. The market for communications solutions for hospitals and the health care is though, and systems need to constantly improve and evolve in order to stay competitive. If CallMeSmart is to offer GSM call functionality, the most related and important future work in addition to providing another external trunk solution will need to offer a way for devices to communicate with the CallMeSmart server while connected to external networks. Losing the ability to send and receive messages and alarms, as well as contact information becoming inconsistent is not acceptable for such a system.

If the system was to implement a way for devices to communicate with the server when outside the range of the hospital networks, this would also open up the ability for the system to offer VoIP through external networks. This could allow the system to maintain the same call if a user ventured outside the range of the hospital network and into another external network as discussed in section 3.1.2. This would eliminate the wait time associated with the GSM call initialization of the current system as the initial call could be continued instead of creating another call.

Similar to this, the current solution is to reestablish a call once the connection

between two users is broken. An improvement to this could be to have the system analyse the network traffic and signal strength in order to predict when connections would be lost. This would allow the system to initialize calls before the initial call fails and ideally reducing the time users need to wait in order for the call to reestablish.

As CallMeSmart is still being developed, there are several other plans for future improvements, including the proposal of making the system ubiquitous and self-learning[70].

7.2 Conclusion

Previously users of the CallMeSmart system were limited to keeping within the limits of the hospitals wireless network in order for the system to function properly. With this thesis a system for maintaining communications as users move out of the range of the wireless network has been implemented and demonstrated to work. In chapter 1 the following question was asked.

> *How can we create a solution for an extramural communication service for the existing CallMeSmart system through a transparent Wi-Fi to GSM switch?*

To this end it has been concluded that while creating a solution for switching between using Wi-Fi and GSM in order for users to communicate when outside the hospital can be beneficial, supplementing this functionality with the ability to also exchange information with the server from external networks is essentially required in order to still provide the same level of user experience as the current version of CallMeSmart provides.

Bibliography

[1] Java 7 end-of-life status. URL http://www.oracle.com/technetwork/java/eol-135779.html.

[2] Teknisk kaos lammer sykehus (technical chaos cripples hospital). http://www.pd.no/Innenriks/helse/article2156578.ece. URL http://www.pd.no/Innenriks/helse/article2156578.ece. Accessed: 2015-05-19.

[3] Telenor legger ned personsokertjenesten om to ar (telenor shuts down pager network in two years). http://www.telenor.com/no/media/pressemeldinger/2001/telenor-legger-ned-personsokertjenesten-om-to-ar/. URL http://www.telenor.com/no/media/pressemeldinger/2001/telenor-legger-ned-personsokertjenesten-om-to-ar/. Accessed: 2015-05-19.

[4] Kaos pa ahus (chaos at ahus). http://www.tv2.no/a/3520524. URL http://www.tv2.no/a/3520524. Accessed: 2015-05-19.

[5] Ieee draft standard for local and metropolitan area networks: Media independent handover services. *IEEE Unapproved Draft Std P802.21/D14, Sept 2008*, pages –, 2008.

[6] Ascom network testing wins market share leadership award. http://www.ascom.com/nt/en/index-nt/nt-news/news-nt/ascom_network_testing_wins_market_share_leadership_award-2.htm, 2011.

[7] Mobile network security report: Norway. https://gsmmap.org/assets/pdfs/gsmmap.org-country_report-Norway-2015-02.pdf, 2015. URL https://gsmmap.org/assets/pdfs/gsmmap.org-country_report-Norway-2015-02.pdf.

[8] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Wirel. Netw.*, 3(5):421–433, October 1997. ISSN 1022-0038. doi: 10.1023/A:1019194325861. URL `http://dx.doi.org/10.1023/A:1019194325861`.

[9] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[10] Norman Adams, Rich Gold, Bill N Schilit, Michael M Tso, and Roy Want. An infrared network for mobile computers. In *Proceedings USENIX Symposium on Mobile & Location-indendent Computing*, volume 10, 1993.

[11] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[12] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. On the value of temporal information in information retrieval. *SIGIR Forum*, 41(2): 35–41, December 2007. ISSN 0163-5840. doi: 10.1145/1328964.1328968. URL `http://doi.acm.org/10.1145/1328964.1328968`.

[13] J.G. Anderson and C. Aydin. *Evaluating the Organizational Impact of Health Care Information Systems*. Health Informatics. Springer, 2006. ISBN 9780387303291. URL `http://books.google.no/books?id=pJvyFoLyBNcC`.

[14] Daniel Avrahami, DARREN Gergle, Scott E Hudson, and SARA Kiesler. Improving the match between callers and receivers: A study on the effect of contextual information on cell phone interruptions. *Behaviour & Information Technology*, 26(3):247–259, 2007.

[15] Mary Baker, Xinhua Zhao, Stuart Cheshire, and Jonathan Stone. Supporting mobility in mosquitonet. In *USENIX Annual Technical Conference*, pages 127–120, 1996.

[16] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of gsm encrypted communication. In *Advances in Cryptology-CRYPTO 2003*, pages 600–616. Springer, 2003.

[17] Louise Barkhuus and Anind K Dey. Location-based services for mobile telephony: a study of users' privacy concerns. In *INTERACT*, volume 3, pages 702–712. Citeseer, 2003.

[18] Victoria Bellotti and Abigail Sellen. Design for privacy in ubiquitous computing environments. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW'93*, pages 77–92. Springer, 1993.

[19] Vaduvur Bharghavan. Challenges and solutions to adaptive computing and seamless mobility over heterogeneous wireless networks. *Wireless Personal Communications*, 4(2):217–256, 1997.

[20] Jesper J. Bisgaard and Dk Aalborg East. How is context and context-awareness defined and applied? a survey of context-awareness.

[21] Nathan J Blum and Tracy A Lieu. Interrupted care: The e ffects of paging on pediatric resident activities. *American journal of diseases of children*, 146(7):806–808, 1992.

[22] Dag Brattli. The software network. 1996.

[23] Milind M Buddhikot, Girish Chandranmenon, Seungjae Han, Yui-Wah Lee, Scott Miller, and Luca Salgarelli. Design and implementation of a wlan/cdma2000 interworking architecture. *Communications Magazine, IEEE*, 41(11):90–100, 2003.

[24] Markus Bylund and Fredrik Espinoza. Testing and demonstrating context-aware services with quake iii arena. *Communications of the ACM*, 45(1):46–48, 2002.

[25] Casey Carter, Robin Kravets, and Jean Tourrilhes. Contact networking: a localized mobility system. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 145–158. ACM, 2003.

[26] Guanling Chen, David Kotz, et al. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[27] Stuart Cheshire and Mary Baker. A wireless network in mosquitonet. *Micro, IEEE*, 16(1):44–52, 1996.

[28] Bachir Chihani, Emmanuel Bertin, Fabrice Jeanne, and Noel Crespi. Context-aware systems: a case study. In *Digital Information and Communication Technology and Its Applications*, pages 718–732. Springer, 2011.

[29] Enrico Coiera and Vanessa Tombs. Communication behaviours in a hospital setting: an observational study. *Bmj*, 316(7132):673–676, 1998.

[30] Douglas E Comer. Principles, protocols, and architecture, volume 1 of internetworking with tcp/ip, 1991.

[31] Steve B Cousins and Michael G Kahn. The visual display of temporal information. *Artificial intelligence in medicine*, 3(6):341–357, 1991.

[32] Maria Cvach. Monitor alarm fatigue: an integrative review. *Biomedical Instrumentation & Technology*, 46(4):268–277, 2012.

[33] Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time attack on the a5/3 cryptosystem used in third generation gsm telephony. *IACR Cryptology ePrint Archive*, 2010:13, 2010.

[34] Stuart A Eisenstadt, Michael M Wagner, William R Hogan, Marvin C Pankaskie, FC Tsui, and W Wilbright. Mobile workers in healthcare and their information needs: are 2-way pagers the answer? In *Proceedings of the AMIA Symposium*, page 135. American Medical Informatics Association, 1998.

[35] Jesus Favela, Ana I Martinez-Garcia, et al. Context-aware mobile communication in hospitals. *Computer*, 36(9):38–46, 2003.

[36] Kelly Creighton Graham and Maria Cvach. Monitor alarm fatigue: standardizing use of physiological monitors and decreasing nuisance alarms. *American Journal of Critical Care*, 19(1):28–34, 2010.

[37] Andy Harter and Andy Hopper. A distributed location system for the active office. *Network, IEEE*, 8(1):62–70, 1994.

[38] William Hersh, Mark Helfand, James Wallace, Dale Kraemer, Patricia Patterson, Susan Shapiro, and Merwyn Greenlick. A systematic review of the efficacy of telemedicine for making diagnostic and management decisions. *Journal of Telemedicine and Telecare*, 8(4):197–209, 2002.

[39] Jason I Hong and James A Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189. ACM, 2004.

[40] Eric Horvitz, Paul Koch, Carl M Kadie, and Andy Jacobs. Coordinate: Probabilistic forecasting of presence and availability. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 224–233. Morgan Kaufmann Publishers Inc., 2002.

[41] Shun-Hsiang Hu, Po-Hsun Cheng, Ren-Hao Wu, Yu-Pao Lin, Hsiao-Chi Hsieh, Bor-Shing Lin, Chu Yu, and Sao-Jie Chen. A seamless wireless network switching tunnel for ubiquitous healthcare environment. In *Consumer Electronics (GCCE), 2012 IEEE 1st Global Conference on*, pages 387–391. IEEE, 2012.

[42] Xiaodong Jiang and James A Landay. Modeling privacy control in context-aware systems. *Pervasive Computing, IEEE*, 1(3):59–63, 2002.

[43] Mitchell H Katz and Steven A Schroeder. The sounds of the hospital. paging patterns in three teaching hospitals. *The New England journal of medicine*, 319(24):1585–1589, 1988.

[44] Anders Lidén. Seamless mobility between current and future ip networks. 2004.

[45] H. Lieberman and T. Selker. Out of context: Computer systems that adapt to, and learn from, context. *IBM Syst. J.*, 39(3-4):617–632, July 2000. ISSN 0018-8670. doi: 10.1147/sj.393.0617. URL `http://dx.doi.org/10.1147/sj.393.0617`.

[46] Sue Long, Dietmar Aust, Gregory Abowd, and Chris Atkeson. Cyberguide: Prototyping context-aware mobile applications. In *Conference Companion on Human Factors in Computing Systems*, CHI '96, pages 293–294, New York, NY, USA, 1996. ACM, ACM. ISBN 0-89791-832-0. doi: 10.1145/257089.257328. URL `http://doi.acm.org/10.1145/257089.257328`.

[47] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[48] Daniel Massaguer, Bijit Hore, Mamadou H. Diallo, Sharad Mehrotra, and Nalini Venkatasubramanian. Middleware for pervasive spaces: Balancing privacy and utility. In *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '09, pages 13:1–13:20, New York, NY, USA, 2009. Springer-Verlag New York, Inc. URL `http://dl.acm.org/citation.cfm?id=1656980.1656998`.

[49] *Rammeverk for autentisering og uavviselighet i elektronisk kommunikasjon med og i offentlig sektor (framework for authentication and non-repudiation in electronic communication with and in the public sector).* Ministry of Government Administration, Reform and Church Affairs, 1.0 edition, 4 2008. URL `https://www.regjeringen.no/globalassets/upload/fad/vedlegg/ikt-politikk/eid_rammeverk_trykk.pdf`.

[50] *Requirements specification for PKI in the public sector.* Ministry of Local Government and Modernisation, 2.0 edition, 6 2010. URL `https://www.regjeringen.no/en/dokumenter/requirements-specification-for-pki-in-th/id611085/`.

[51] ANN MINNICK, KATHY PISCHKE-WINN, and MARY BETH STERK. Introducing a two-way wireless communication system. *Nursing management*, 25(7):42–49, 1994.

[52] Stefano Mizzaro, Elena Nazzi, and Luca Vassena. Retrieval of context-aware applications on mobile devices: How to evaluate? In *Proceedings of the Second International Symposium on Information Interaction in Context*, IIiX '08, pages 65–71, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-310-5. doi: 10.1145/1414694.1414710. URL `http://doi.acm.org/10.1145/1414694.1414710`.

[53] Karsten Nohl. Attacking phone privacy. *Black Hat USA*, 2010.

[54] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 265–268. ACM, 2009.

[55] Julie Parker and Enrico Coiera. Improving clinical communication a view from psychology. *Journal of the American Medical Informatics Association*, 7(5):453–461, 2000.

[56] Ahmad Rahmati, Clay Shepard, Chad Tossell, Angela Nicoara, Lin Zhong, Phil Kortum, and Jatinder Singh. Seamless flow migration on

smartphones without network support. *arXiv preprint arXiv:1012.3071*, 2010.

[57] Keith J Ruskin. Communication devices in the operating room. *Current Opinion in Anesthesiology*, 19(6):655–659, 2006.

[58] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

[59] Bill N Schilit and Marvin M Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, 1994.

[60] Bill N Schilit, David M Hilbert, and Jonathan Trevor. Context-aware communication. *Wireless Communications, IEEE*, 9(5):46–54, 2002.

[61] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *Handheld and ubiquitous computing*, pages 89–101. Springer, 1999.

[62] Jeremiah Scholl, Per Hasvold, Eva Henriksen, and Gunnar Ellingsen. Managing communication availability and interruptions: a study of mobile communication in an oncology department. In *Pervasive Computing*, pages 234–250. Springer, 2007.

[63] Sue Sendelbach and Marjorie Funk. Alarm fatigue: a patient safety concern. *AACN advanced critical care*, 24(4):378–386, 2013.

[64] Statistisk sentralbyraa. Statistisk aarbok 2012 (statisitical yearbook 2012), 2012. URL `https://www.ssb.no/befolkning/artikler-og-publikasjoner/_attachment/91790?_ts=13c6c86a160`. p. 359.

[65] Kamran Sheikh, Maarten Wegdam, and Marten van Sinderen. Quality-of-context and its use for protecting privacy in context aware systems. *Journal of Software*, 3(3):83–93, 2008.

[66] Julie Siebens. Extended measures of well-being: Living conditions in the united states: 2011, 9 2013. URL `https://www.census.gov/prod/2013pubs/p70-136.pdf`.

[67] Asim Smailagic and Daniel P Siewiorek. Matching interface design with user tasks. modalities of interaction with cmu wearable computers. *Personal Communications, IEEE*, 3(1):14–25, 1996.

[68] Terje Solvoll and Jeremiah Scholl. Strategies to reduce interruptions from mobile communication systems in surgical wards. *Journal of Telemedicine and Telecare*, 14(7):389–392, 2008.

[69] Terje Solvoll, Jeremiah Scholl, and Gunnar Hartvigsen. Physicians interrupted by mobile devices in hospitals: understanding the interaction between devices, roles, and duties. *Journal of medical Internet research*, 15(3), 2013.

[70] Terje Solvoll, Monika Johansen, Gunnar Hartvigsen, and Alain Giordanengo. Callmesmart becoming ubiquitous and self-learning. In *eTELEMED 2015, The Seventh International Conference on eHealth, Telemedicine, and Social Medicine*, 2015.

[71] Terje Geir Solvoll. From being interrupted by mobile devices to callmesmart: A context-sensitive communication system for mobile communication in hospitals. 2013.

[72] Mike Spreitzer and Marvin Theimer. *Providing location information in a ubiquitous computing environment (panel session)*, volume 27. ACM, 1994.

[73] Patrice A Spurck, Mary L Mohr, Anna M Seroka, and Martha Stoner. The impact of a wireless telecommunication system on time efficiency. *Journal of Nursing Administration*, 25(6):21–26, 1995.

[74] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[75] Kenichi Taniuchi, Yoshihiro Ohba, Victor Fajardo, Subir Das, Miriam Tauil, Y-H Cheng, Ashutosh Dutta, Donald Baker, Maya Yajnik, and David Famolari. Ieee 802.21: Media independent handover: Features, applicability, and realization. *Communications Magazine, IEEE*, 47(1): 112–120, 2009.

[76] Mark Van Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Adaptive hypermedia and adaptive web-based systems*, pages 235–244. Springer, 2004.

[77] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[78] Roy Want, Bill N Schilit, Norman I Adams, Rich Gold, Karin Petersen, David Goldberg, John R Ellis, and Mark Weiser. An overview of the parctab ubiquitous computing experiment. *Personal Communications, IEEE*, 2(6):28–43, 1995.

[79] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[80] Mark Weiser, Rich Gold, and John Seely Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM systems journal*, 38(4):693–696, 1999.

[81] Johanna I Westbrook, Amanda Woods, Marilyn I Rob, William TM Dunsmuir, and Richard O Day. Association of interruptions with an increased risk and severity of medication administration errors. *Archives of Internal medicine*, 170(8):683–690, 2010.

[82] M. F. Worboys. A unified model for spatial and temporal information. *The Computer Journal*, 37(1):26–34, 1994. doi: 10.1093/comjnl/37.1.26. URL `http://comjnl.oxfordjournals.org/content/37/1/26.abstract`.

[83] Robert Wu, Peter Rossos, Sherman Quan, Scott Reeves, Vivian Lo, Brian Wong, Mark Cheung, and Dante Morra. An evaluation of the use of smartphones to communicate between clinicians: a mixed-methods study. *Journal of medical Internet research*, 13(3), 2011.

[84] Kurt Zeilenga. Lightweight directory access protocol (ldap): Technical specification road map. 2006.