UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology
Department of Computer Science

Nexans Tracking System

An indoor positioning system using Bluetooth Low Energy

—

Ragnhild Kosmo Holm

INF-3981 Master's Thesis in Computer Science - June 2015

Abstract

Nexans Rognan is a cable manufacturer located in Nordland County. Each new cable order have a set of requirements provided by the customer. The cable specifications change in terms of characteristics of fibre optical elements, type and quality of cable armouring and dimension and amounts of electrical conductors. Due to the variety in different cables produced, they also need a wide array of different cable drums available at the plant. The drums are stored in sporadic locations around the plant and the lack of organization of the cable drums creates extra work for production workers. They estimate an yearly cost of 2-3 FTE's is lost searching for drums. This have created a need for an *indoor tracking system*.

The main objective of this thesis is implementing a system which should follow a cable drum from when it enters the factory for the first time, and until it is disposed or shipped out to the customer. It uses Bluetooth Low Energy (BLE), NFC, Android devices and Microsoft Azure for creating fully functional system for cable drum organization. The system heavily emphasize the use of user roles to personalize the web page for each user for creating a more seamless work flow.

The system consist of different modules, performing different tasks. Three different mobile applications, a web site, a middleware and a SQL database have been created. Two of the mobile applications provides functionality for tracking the cable drums inside the factory hall. The NFC application provides functionality for identifying the drum, and for updating the information about it. The middleware does most of the work within the system. It decides whether or not the drum has moved, checks user privileges and implements methods for adding, updating and deleting from the database. The web site provides a simple user interface, and creates simple methods for a user to update the backend with correct information.

The system have been designed and implemented from requirements set by Nexans Rognan. Multiple smaller, fault prone and time-consuming tasks have been adopted into the system. Users at Nexans have been included in the testing, and they have provided feedback which have been implemented in the system. Testing have shown that the system is mature enough to be used as it is now, and the technology chosen does suit the environment at Nexans.

Acknowledgements

I would like to thank Nexans Rognan for the unique opportunity to design my own thesis. Your positive attitude have been very motivating and made the cooperation with you enjoyable. I would especially like to thank Steffen Aarøe for being my contact person at Nexans, for showing me around the plant, helping me with testing and answering my questions.

Next, I would like to thank Randi Karlsen and Anders Andersen for keeping me on the right track throughout this semester. Your comments and guidance have been much appreciated.

Furthermore, I would like to thank my fellow classmates for the awesome five years we have had together. I wish you all the best for the future!

Special thanks goes to my family for all your support and help through the years. Finally, I would like my thank my boyfriend, Christopher Haugen, for being supportive, helpful and patient through the work with this thesis.

Contents

List of Figures

List of Listings

1

Introduction

This thesis is done in cooperation with Nexans Rognan. Nexans produces different types of cables, ranging from fibre cables, copper cables and Remotely Operated Vehicles (ROV) cables. Their customers are mostly located in the industry, petroleum, construction and infrastructure markets. Nexans Rognan mostly have customers within the petroleum industry.

Nexans Rognan produce mostly specially made cables for different customers. Due to the huge variety in cables produced, they need many different cable drums available at the plant. The cable drums used vary in terms of material (steel or wooden), weight limit, material and diameter.

The current situation at Nexans Rognan is that they do not organize the cable drums in a manner that makes it simple to keep track of the status of the drums as well as locating drums within the plant. They have a system, SAP, where orders are placed, and a list of drums at the plant is registered. As the production have grown, the way Nexans Rognan organize cable drums have not been amended in line with the other developments. It have become increasingly more time consuming for the production workers to locate the drums and to find suitable storage location. The amount of time spent on locating drums are estimated to be around 2-3 FTE's, and a more efficient system for organizing cable drums is now needed.

Since GPS was introduced in the 1970's, more and more services that rely on positions of different objects have emerged. GPS have become a de facto standard for outdoor positioning system. There are no such standard in indoor positioning systems. Over the last two decades increasingly more research have been done on indoor tracking systems, but neither of the technologies presented have gotten the same results as GPS have for outside tracking.

Indoor tracking systems are implemented using radio waves, acoustic signals, information from mobile devices or magnetic fields. Many of these are inaccurate or expensive, as we will present in later chapters. Bluetooth Low Energy (BLE) have made it possible to create an affordable indoor tracking solution. This is a radio frequency-based (RF) technique using Bluetooth and signal strength measurements to position the object inside for example a house, factory or hospital.

By using BLE, the tracking system is simple to implement, the BLE beacons are reasonably priced, ordinary mobile devices can be used to locate the objects and the range of the beacons are flexible, from only a few meters up to approximately 50 meters.

1.1 Pre-projects

This project is done in three parts, one feasibility study, one pre-project and this thesis.

First study - feasibility study The first study researched different technologies, for example Near Field Communication (NFC), Bluetooth Low Energy (BLE) and Radio frequency Identification (RFID). It presented the advantages and disadvantages with each technology, and concluded with BLE as the most promising technology to use at Nexans Rognan. This feasibility study also looked at some other, ready-to-use, tracking systems, and researched how these would fit into the environment at Nexans Rognan. These technologies and systems are presented as related work in this thesis.

Second study - pre-project The second study started testing on the BLE and NFC technology. This testing included how the behaviour of BLE beacons were in the harsh environment at Nexans Rognan and what range that can be expected. A few different beacon types, from different providers, was tested. The two most promising beacons were *StickNFind (SNF) Pro* and *Estimote*, with *SNF Pro* slightly better because of its size and price. The testing done in this pre-project did show that the BLE technology is mature enough for use in an indoor tracking system at Nexans Rognan.

Because of limitations with Apple iOS and Windows Phone, as presented in section 3.6, Android was chosen as the platform to implement the tracking system application on. This project included an unfinished *BLE Reference Point Scanner*. It was not used as a tracking tool in this project, but instead as a tool for testing different specifications of the beacons, such as range and advertising interval.

1.2 Project statement

Two pre-projects, presented in section 1.1, are done prior to this thesis, and this thesis will be built upon the results from these.

> The system is created from requirements provided by the customer, Nexans Rognan. The goal of this thesis is to create a functional tracking system using BLE beacons, Android devices and Azure as a cloud provider. The system implemented emphasizes the use of access rights to customize the web site for each user.

1.3 Scope and limitations

The motivation for creating this system was to give Nexans Rognan, both users and the company itself, better overview and control over the production at the plant. This thesis will design and implement a prototype system with the functionality needed to support the goal presented in the project statement.

The existing production system, SAP, is not integrated with this system. Integrating *Nexans Tracking System* with SAP is extensive work, and is out of scope of this thesis.

Due to financial restrictions a full scale system have not been implemented and tested at Nexans. An operational system will include up to a few thousand beacons. It will also and a large amount of *reference points* (smart phones) to create a grid system for plotting the beacons around the plant. A smaller amount of beacons and smart devices have been bought and tested, and therefore this system is a downscaled version of the full version. Beacon testing have not been completed. This is mostly regarding testing of battery life. Each beacon is expected to last approximately two to five years, and testing have only been done for one year.

1.4 Contribution

The successful use of Bluetooth Low Energy beacons shows that it is possible to implement an indoor tracking system using this technology. Testing of the beacons shows that, even though they are exposed to much interference from machines at the plant, they have approximately 20 meters range.

Different modules have been implemented, together creating a fully operational indoor tracking system for use at Nexans Rognan. The system brings together many smaller, time-consuming and fault prone tasks at Nexans into one system for better overview and control over production. By providing methods for more accurate planning of the production of a cable drum, this system will contribute to a more efficient work flow at Nexans Rognan.

The middleware implemented, contains most of the functionality in this system and the majority of design choices are done here. Different functionalities have been implemented, where an important part is emphasizing user roles. Users of different roles will have customized access to different parts of the system. The *Administrator* in the system have the rights to add users to different role groups, the purchasing departments is able to add new drums to the database, while some workers only be able to see information about the cable drums. Three mobile applications have been implemented, utilizing the Bluetooth Low Energy and NFC technology.

1.5 Structure

The rest of this thesis is structured as follows:

Chapter 2 presents the context of this thesis. It will present Nexans Rognan, the challenges and limitations they have regarding tracking of cable drums at the plant as well as presenting the motivation for creating this system.

Chapter 3 presents relevant technical background information for this thesis. This includes for example a comparison of different cloud providers and different Bluetooth Low Energy (BLE) beacon providers.

Chapter 4 presents related work to this system. This will include information about tracking systems using other technologies and other uses of BLE.

Chapter 5 describes the design of the system implemented.

Chapter 6 describes the implementation details of the system.

Chapter 7 presents challenges that have and can occur using this system. It will also present an evaluation of the system implemented. This includes results from beacon testing, security issues and details on other design choices than the ones used in the system.

Chapter 8 will summarize the thesis. Future work will also be presented in this chapter.

2

Context

This chapter will give an introduction to Nexans Rognan. It presents the challenges at Nexans that can benefit from the system implemented in this thesis.

Previous projects are presented in section 1.1, and this chapter is based on the conclusions from them.

2.1 Nexans Rognan

Nexans is one of the world's leading cable manufacturers, and they employ over 25 000 people world wide. Their headquarters are located in Paris, France, but they have factories all over the world. They mostly produce cables for infrastructure, construction and petroleum.

Four of the Nexans factories resides in Norway, under the name *Nexans Norway*. These factories are located in Halden, Karmøy, Namsos and Rognan. They also have an office location in Helsfyr, Oslo. Together about 1100 people is employed by Nexans Norway. The factories in Norway is the leading supplier of telecommunication and power cables in Norway. They also produce large amount of offshore cables [1].

1. http://www.nexans.no/eservice/Norway-no_NO/navigate_217577_2371_40_6045/Overview.html

This thesis is done in collaboration with Nexans Rognan. The Rognan factory is located in Saltdal municipality in Nordland County. They mostly produce specially made cables for the petroleum industry and fibre optic cables. They also produce ROV (Remotely Operated Vehicles) cables, which is used between the mother ship and a submarine. What distinguish the Rognan factory from the other plants in Nexans Norway, is that they mostly produce cables with special requirements, while the other factories produce standard cables. More about this is presented in section 2.1.2.

2.1.1 SAP

SAP is the production system used for planning and executing production of a cable order. The planning includes, among other things, how much and what kind of raw materials to use, which machines are required, in what order machines should be used and how many work hours are needed to complete the order within the given time.

The system also includes information about how many cable drums of each type is available at the plant. Even though they know how many of each cable drum type there is, this information is not completely trustworthy. SAP is rarely completely up to date. Drums can be broken, in use by others, or containing raw material or products.

SAP is currently frozen, waiting for a new version to be implemented. This, the limited time span of this thesis, and the dependency on other developers, makes it impossible to integrate *Nexans Tracking System* with SAP. When a new version of SAP is implemented it is possible to add an extension to SAP, and this extension can include the system created in this thesis.

2.1.2 Production

The Rognan plant has a wide array of customers. When a customer places an order, they have given a set of specifications that Nexans have to follow. These specification vary, and Nexans can not produce the cable before an order is received. Examples of specifications that can change is the characteristics of fibre optical elements, type and quality of cable armouring and dimension and amounts of electrical conductors.

When a customer places an order, the specifications have to be reviewed. Depending on these specifications they will know which machines are needed, which tests that should be performed and how many work hours are needed. This planning is done in the existing production system, SAP.

Due to the production being non-repetitive which cables that are needed vary from order to order. Cable drums are either made from steel or wood and have different properties. Wooden drums are cheaper to buy, and can therefore be disposed. They are often used for cables that have light reinforcement/protection and they are not used in water nor for voltage tests. Wooden drums are often the preferred drum to ship to customers.

Steel drums can withstand more and are more expensive than wooden drums. They are not the preferred type to ship out to customers because of the cost. They are used for voltage tests and are soaked in water. Steel drums, of type LS11, LT10 and LT20, are used for testing in this thesis.

SAP includes a list of all cable drums at the plant. However, much of the information needed is not included in SAP. This information includes whether the drum is in use, if it has material spun on it, if it is being performed maintenance on or if it's broken. There are often halts in production due to the lack of overview.

2.1.3 Tracking cable drums - current situation at Nexans

The current situation at Nexans is that they lack control over the cable drums. This includes not knowing where the drum is, if it is in use or is damaged or broken. Production, as it is done now, have three major challenges that can be made more efficient by implementing a tracking system for cable drums.

- The SAP production system lacks the functionality to add remarks to the cable drum. The production workers have spreadsheets to take notes in if they notice some faults on the drum. This spreadsheet is shown in figure 2.1. Other workers use post-it notes to write down any remarks. The drums in use can be old, since they do not add the production date, and they can also be crooked or rusty due to lack of maintenance. Nexans Rognan wishes to enforce regular maintenance every two years, but this is not done at the current moment.

- The second challenge is that they do not always know whether or not the drum is in use by other parts of production. Those who accept new orders only sees the amount of drums at the plant, but does not know the exact number of available drums. Lack of drums available creates a halt in production, and they might not be able to finish the production within the agreed time.

- The last, and major, challenge is that they lack overview over the locations of the drums. The plant covers a large area, both inside and outside, and drums are mostly placed where there is room. Nexans Rognan have expanded their production, including new and larger machines, and this have created an even more shortage of storage for cable drums. They do not have designated areas for storing drums that are currently not in use. The drums are placed in clusters around the plant, as can be seen in figure 2.2a, 2.2b and 2.2c. Larger drums, as the one seen in figure 2.3a, are placed in different locations where there is available space.

Not knowing the location of the drums at all time creates a challenge for the workers that should transport the drums from one location to another. They rely on their own and other peoples memory when searching for drums that are scheduled to be used. Nexans Rognan have estimated that approximately 2-3 FTEs [2] of 700 000 NOK each year searching for drums.

2. Full-time equivalent. A unit that defines approximate cost of one production worker

REGISTER OG SNELLE / TROMMEL NUMMER

Behov for ny merking er når, merking mangler, lesbarheten er redusert og når merker av papir er brukt. **JA/NEI**

Generell status beskrives i klasser 1-4, hvor: 1

erok	1	er OK.
	2	er mindre feil
	3	er grove feil og kan ikke brukes før feil er utbedret.(Skal til Rep)
	4	er vrak
Renhetstilstand klassefiseres 1-4, hvor	1	Er OK
Kombinasjon eks 2,4 kan brukes.	2	skitten utvendig
	3	skitten utvendig og innvendig
	4	Tilgriset med vaselin /reogel/olje
Rustgrad klassefiseres 1-3	1	er ok
	2	Rust på flater som ikke er i berøring med kabel
	3	Rust på flater som er i berøring med kabel

Type	Merket med nr.	Behov for ny nr merking	Generell status	Renhetstilstand	Rustgrad	Farge		Dato	Sign		Dato	Sign	Kommentar
LS 11	1		1					14.jan.06	BM				
LS 11	2		1					23.jan.06	RH				
LS 11	4		1					27-okt-04	RH				
LS 11	6		1					27-okt-04	RH				
LS 11	7		1					22-des-04	HOH				
LS 11	8		1					27-okt-04	HoH				
LS 11	11		1					22-des-04	Skolee.				
LS 11	12		1					22-des-04	Skolee.				

Figure 2.1: Spreadsheet showing the status of the cable drums

(a)

(b)

(c)

Figure 2.2: Three subfigures showing clusters of cable drums

2.1.4 Desired situation

Through meetings with Nexans, both production workers and managers, a set of specifications were formed. The system should include the following features and information:

- It should be easy to update and handle by the users.
- Full overview of the drums at Nexans should be available. This should include the location of the cable drums.
- The users should be able to search for a particular drum or drum type.
- The last maintenance of each drum should be shown. This will include when the last maintenance was, who did it, what was done and when the next maintenance is scheduled. They wish to enforce regular maintenance every two years.
- They also wish to add a link between the existing system SAP, and the tracking system. Due to the extent of this task, it will not be implemented in this thesis.

(a)

(b)

(c)

Figure 2.3: Three subfigures showing different singles that are used in testing the solution at Nexans. All three have a height of approximately 2-4 meters

3

Technical Background

This chapter will present relevant technologies, concepts, frameworks used and different development platforms.

3.1 Internet of Things

In 2010 there were approximately 12.5 billion devices connected to the Internet, and Cisco IBSG estimates that in 2015 there will be twice this amount, 25 billion devices [1]. IoT's represents the ability to gather, analyse and distribute data. This data is turned into information and knowledge. IoT is traced back to 1999 and MIT [2] when they started working within the field of RFID (Radio Frequency Identification) and sensing technologies. From here the technology has evolved into a *network of networks* [1], which collects and analyses data from many different sources.

By adding more and more devices, the term *ubiquitous computing* emerged, where computing appears everywhere, using any device in any format [3]. The computing unit within the term *Internet of Things* is called a *thing*. A *thing* can be a heart monitor implant, an auto mobile or a cable drum. The *things* have sensors which gives information about the status of the object [4]. In Nexans' case this will be location of the cable drums, and in other cases this can be battery life or temperature.

Sensor energy is one of the more relevant challenges for this thesis. In general, for IoT to real its full potential, the sensors have to be self-sustaining. By this, it means that they have to produce energy from light, airflow or vibrations to keep itself going. As we know, most sensors needs batteries, rechargeable or not rechargeable, to function, so this technology is still in the making for most sensors [1].

3.2 Cloud services

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [5].

Within Cloud computing, instead of storing and accessing data and programs on a local computer, the data is retrieved from *over the Internet*. Cloud computing refers to both the software that deliver the service, and the hardware in the data centres, which the software runs on. When a cloud is open to the public, and is a *pay-as-you-go* service, it is called a *public cloud* [6]. A *private cloud* refers to internal data centres of a business. A private cloud is mostly used by large businesses that can benefit from cloud computing, and which have the resources to develop their own cloud.

There are several advantages for companies to use clouds. An infinite amount of resources are available on demand. Companies can start small and increase the the amount of resources needed as the business increases its production. This will decrease the need for large investments before knowing how much resources they need [6].

One of the most important obstacles of using a cloud provider is the security aspect. Why should the business trust a cloud provider with their data? There are both internal and external security threats when using a cloud. The external are in the hands of the cloud provider, and this includes the physical security. This also includes external firewall policies [6]. The cloud user is responsible for application-level security.

One of the more common approaches for cloud storage services is to encrypt the data *at-rest*[1] using a 256-bit AES encryption. Google Drive encrypt the data using AES 128, while Microsoft OneDrive does not encrypt the data [7]. When

1. The cloud providers storage servers

discussing *in-transit* security, many cloud providers uses SSL [2]/TLS[3] to establish a secure connection between client and server. Recent research have shown that the system is still vulnerable even though SSL/TLS is used [8].

The cloud providers does provide different levels of resources. If the user is provided with a lower level of abstraction, the user is responsible for more of the security challenges. The users does not need to handle these security challenges themselves, because this can be outsourced to a third party service [6].

A challenge for many businesses is data transfer bottlenecks. If the software running is data-intensive, the costs of transfers can become very expensive. In *Nexans Tracking system*, the data sent between the cloud provider and Nexans is not data-intensive, and only smaller amount of text is sent and received from applications.

By renting resources from a cloud provider it is possible to cut cost. This includes the need of hardware, software electricity and bandwidth. There is also limited need for in-house IT personnel. By putting data in the cloud the work regarding security, maintenance, backup and support is simplified for the user. By using a cloud it is simple to scale up a system when in need of more resources. With using a local server, this process would have been more time consuming since hardware would have to be ordered and set up, and software installed. This is handled by the cloud storage provider, which in this thesis is Windows Azure.

3.2.1 PaaS vs. Saas vs. Iaas

When implementing a new system or adding functionality to an already existing system the businesses have to analyse the requirements they have for. These requirements can regard how much resources, in manpower and money, they have available. It can also regard what security level they wish to pursue, and how much control they want over the applications and data they store. By having everything locally, both hardware and software, they businesses have to think ahead and see if this is something that they will have to expand. If this is the case, they might decide to add more hardware from the start. By using a cloud provider they can start smaller, and expand the system when their needs increase.

In this thesis the choice was between establishing a server locally at Nexans or using a cloud provider. Nexans does currently have a local instance of a SAP

2. Secure Socket Layer
3. Transport Layer Security

system, however this system is currently frozen and will not be expanded in the near future. This thesis does not include an integration with SAP. However, if this system were to be implemented in SAP, the system could not have been put in Azure, and would have to be run locally. The reason why choosing a cloud provider in this theses, is because Nexans does not need to invest in new hardware or software to run the prototype system created, it will also simplify future maintenance. This section will present what services that are available for Nexans if they wish to proceed with keeping the system in the cloud.

Software as a Service, or *SaaS*, represents the largest cloud marked. It uses the web to deliver services through an interface that is accessed on the client's side. These services are provided by a third-party [9]. Some of the characteristics of a *SaaS* service, is that many of these services can be accessed through the web browser, often just needing some additional plugins to function. For a business using *SaaS* they can outsource the managing of applications, middleware, storage and servers [9]. The user does not need to think about hosting, updating, managing or developing applications or storing data [10].

Platform as a Service, or *PaaS*, is what Microsoft is betting on when developing Azure as a long time project [11]. Google also presents their cloud solution as a fully-managed *PaaS* [12]. One of the main reasons to use *PaaS* is to increase the developers' productivity. It is a framework where developers can build their own applications, while also making it more cost-efficient and simpler to publish applications [9]. In *PaaS* the system provider makes most of the decisions when determining which operating system to use, which programming language is supported and the API used [10].

Amazon Web Services (AWS), Azure and Google Compute Engine can also go under the last service, the *Infrastructure as a Service*, or *IaaS*. These providers provide access to raw computing infrastructure and the middleware software [13]. This service provides self-service models for accessing and managing remote data centres [9]. The resources can be storage, computation power or networking. The advantage with *IaaS* as opposed to having the hardware locally is that they can pay for what they use (and not for what they have), limited start-up cost and less risk of data loss because of globally replicated data. They do not have to have IT staff to manage the hardware.

3.2.2 Windows Azure

> Windows Azure Storage (WAS) is a cloud storage system that provides customers with the ability to store seemingly limitless amounts of data for any duration of time. WAS customers have access to their data from anywhere at any time and only pay for

what they use [14].

WAS is a cloud storage that store the data in *Blobs, Tables and Queues*. The *Queues* are the message delivery, *Tables* are structured storage and *Blobs* are the user files. The common usage pattern is that the data enters and leaves the system as *Blobs*, *Queues* processes the *Blobs* and the service state and results are stored in *Tables* or *Blobs* [14].

Before implementing Azure they received feedback from many potential customers, and this resulted in a set of specifications.

- **Strong consistency** - Many of the customers wanted strong consistency. Many enterprise customers are moving their data to the cloud, and they want to be able to perform conditional reads, writes an deletes for optimistic concurrency on the strong consistent data.

- **Global and scalable storage** - WAS implements a global namespace that allows data to be stored and retrieved consistently from any location in the world.

- **Disaster recovery** - Locations are always in risk of disasters, like earthquakes, floods or hurricanes. Therefore WAS stores the data across multiple data centres around the world. In case one datacenter is faulty, the other data centres will be available.

- **Cost of storage** - To lower the cost of storing data in WAS, multiple customers share the same storage infrastructure.

To provide a single global namespace they leverage on DNS as a part of the storage namespace. The namespace is divided into three parts, the *AccountName*, the *PartitionName* and an *ObjectName*. The *AccountName* is a customer selected account name. In this thesis it is *nexanstrackingsystem*. This name is used to locate the primary storage cluster and data center. It is possible for an application to use multiple *AccountNames* to store the data in different locations. The *PartitionName* locates the data once the request have reached the storage center. The *PartitionName* holds many objects, and the *ObjectName* identifies the different objects inside the partition [14].

WAS provides string consistency, global partitioned namespace and disaster recovery. Multiple users will run on the same hardware, and this reduces the costs per user.

Windows Azure started up as a *PaaS* in the form as Microsoft Windows Azure, but later have evolved into also supporting *IaaS* features. This also included support for Linux servers, and not just Windows operating systems. For Windows-centric developments, Azure offers better options for this. Visual Studio, .NET and Windows programming languages are fully supported [15], and users can create and deploy applications straight from Visual Studio. They also provide SDKs for many other languages, like PHP, Java and Python.

3.2.3 Amazon Web Services

> Amazon Web Services (AWS) provides on-demand delivery of IT resources via the Internet with pay-as-you-go pricing [16].

Amazon is one of the most long-living companies using a decentralized IT infrastructure. In 2005 they have spent a decade building the IT infrastructure that supports one of the world's largest online retain stores. From their retail platform, AWS emerged in 2006 [4] [5]. AWS was developed so that other organizations could rent and utilize their cloud resources. Data is written and read using for example SOAP and REST, and is available through web browsers.

AWS consists of a variety of web services, from which the user can chose what or which resources that suits the organization best. Below is a short presentation of the two different resources most relevant for this thesis.

- *Amazon Elastic Compute Cloud (Amazon EC2)* is a web service that provides compute capacity in the cloud. The amount of resources for the user is not fixed, and can be scaled up or down as the need for resources change [6].

- *Amazon Simple Storage Service (Amazon S3)* is a simple web service that can be used to store and retrieve data in the cloud [7].

In addition to these two, there are many services that different users can utilize, like building their own online store upon Amazons infrastructure or retrieving historical pricing for different items of interest [16]. Other examples are *SimpleDB* and *Simple Queue Service*. The *SimpleDB* service provides a database service that allows users to store structured data. This service provides the facilities of a relational database as well as a web interface for this system. The *Simple Queue Service* is a reliable queueing service between instances in EC2 [17].

4. http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-newsArticle&ID=1216597&highlight=
5. http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-corporateTimeline
6. http://aws.amazon.com/ec2/
7. http://aws.amazon.com/s3/

AWS is a flexible platform, making it possible for the organization to decide what programming platform or model they wish to use to solve their problem. The user will also chose what resources are needed, from compute power to storage. When a user puts their data in the Amazon EC2, they pay relatively low prices for storage. When they run the application, they pay for the amount of compute nodes they need, and is done at an hourly rate. While having access to the rented nodes, the user will have root access to the system. This control implies that the user can configure the node as the user desires and can then chose what system software is needed [17].

3.2.4 Google Cloud

This section will present some of the details from Google Cloud that is relevant for Nexans when deciding which cloud provider to use, if they decide to have the data in the cloud.

> Google Cloud Platform is a set of modular cloud-based services that allow you to create anything from simple websites to complex applications [8]

Google Cloud Service consist of modules, as seen in figure 3.1. The platform is based on Python and it provides the user with the possibilities of storing data, hosting applications and networking [18]. In addition to Python, they support other programming languages like Java, PHP and Go. *Google Cloud Service* consists of three parts within the compute section of the cloud, the *Compute Engine*, the *App Engine* and the *Container Engine*.

- *App Engine* is Google's *PaaS*, and makes it simple for users to develop and publish applications [9]. It is possible for users to try *Google App Engine* for free. There are some restrictions to this free trial, for example it does only include 1 GB of data storage [10].

8. https://cloud.google.com/
9. https://cloud.google.com/products/
10. https://cloud.google.com/appengine/docs/whatisgoogleappengine

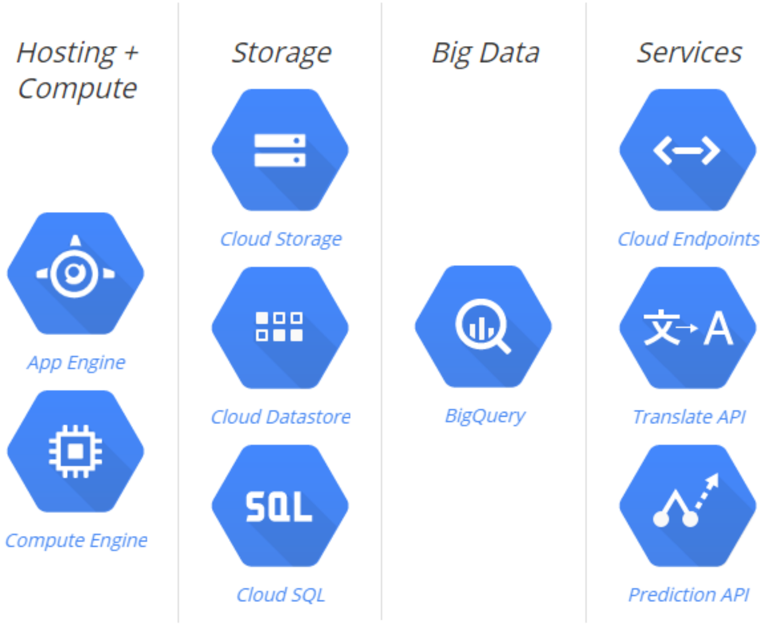

Figure 3.1: The different modules available in the Google Cloud

The *App Engine* offers three distinct sets of features for the user. First is *PaaS*, which gives the user the opportunity to create applications that support transactions and scalable and available applications. Second is that they provide *SaaS*, which is finished applications that can be used by the user straight away. Thirdly they provide the user with the availability to integrate systems and applications from other service platforms, into the *App Engine* [18]. With *Google App Engine* it is easy to deploy new applications. They support persistent storage, scaling and load balancing and integration with other *Google Cloud Services* and APIs.

- The *Compute Engine* is their *IaaS*. They run large-scale workloads on virtual machines. These machines are hosted on Google's infrastructure, and the user can chose what virtual machine, from micro-VMs to large VMs, that would fit their needs [11]. By using Google's fibre network the users can gain the performance and consistency required. Depending on the users need, Google's *Compute Engine* supports global load-balancing, strong and consistent bandwidth across machines, encryption of data, scalable and reliable machines and their VMs support several types of distros [**?**]. As the other two cloud providers presented in this thesis, the user will pay for what he uses and nothing more.

11. https://cloud.google.com/compute/

Within the module *Storage*, Google also provides three more sections, the *Cloud Datastore*, the *Cloud SQL* and the *Cloud Storage* [12].

- Cloud Datastore is a schemaless database for storing non-relational data [13]. It automatically handles sharding[14] and replications, and this creates an highly available and consistent database. It also provides ACID[15] transactions, and does this using optimistic concurrency control. By using this module the user does not need to worry about the underlying structure. It allows the user to search through the data and sort the data as needed [16].

- Cloud SQL provides a relational MySQL database. The data stored in the database is replicated at many locations, which means the data is stored durable and it is both available and secure. Google also handles database management and patch management to ensure the performance of the database [17] [18].

- Cloud Storage is a durable and highly available object storage service [19]. Within this section Google provides different services, where two of them are presented below.

  – The *Standard Storage* provides highest level of performance, durability and availability. Data that is stored here needs low latency access. It is enabled at the *bucket level* [20]. Each bucket contains objects that can be accessed by their own methods. *Standard Storage* is also the default storage, so that if the user does not chose any other storage option, *Standard Storage* is chosen.

  – The *Durable Reduced Availability (DRA) Storage* stores data at lower cost. This lower cost comes with the disadvantage of less available data. DRA is also enabled at the *bucket level* [21].

In addition to this Google provides a service so that the user can gain insight on his application, and how it performs and the availability. They also have

12. https://cloud.google.com/products/
13. https://cloud.google.com/products/
14. Database partitioning that is used to separate a large database into several smaller and faster *shards*
15. Atomic, consistent, isolated and durable
16. https://cloud.google.com/datastore/
17. https://cloud.google.com/sql/
18. https://cloud.google.com/products/
19. https://cloud.google.com/products/
20. https://cloud.google.com/storage/docs/standard-storage
21. https://cloud.google.com/storage/docs/durable-reduced-availability

algorithms to analyse data to predict the behaviour of an application.

3.2.5 Comparison of cloud providers

These three cloud providers does offer much of the same features and services, but this section will provide some of the differences between them. For this thesis which cloud provider chosen would not have a significant impact on the results. Because of platforms used and knowledge of the Windows platform at Nexans Rognan, Azure was chosen. In addition, Nexans runs Windows servers, and it would be simple to port the data to one of their servers if needed. This choice is not a fixed choice, and Nexans can easily change cloud provider, or decide to have the data locally later on.

A *normal PaaS* service have included databases, message queues and caching, but Azure, Google and Amazon have also included services belonging to the *IaaS* core services like virtual machines. Therefore it is difficult to place each provider into only one of the categories.

Google File System (GFS) is the system closest to Azure. One of the major differences between the systems is that GFS allows relaxed consistency across replicas and does not guarantee that the two replicas are bitwise the same. WAS does provide the guarantee that two (or more) replicas are bitwise the same [14].

	Virtual CPUs or cores	RAM	Cost per hour (US$)
Amazon m1.medium	1	3.75GB	12 cents
Amazon c3.large	2	3.75GB	15 cents
Amazon m3.2xlarge	8	30.00GB	90 cents
Google n1-standard1	1	3.75GB	10.4 cents
Google n1-highcpu-2	2	1.80GB	13.1 cents
Google n1-standard-8	8	30.00GB	82.9 cents
Windows Azure Small VM	1	1.75GB	6 cents
Windows Azure Medium VM	2	3.50GB	12 cents
Windows Azure Extra Large VM	8	14.00GB	48 cents

Figure 3.2: A price comparison of different cloud providers, and different specifications[19]

For many businesses it is important to keep some of the data close to home, and not in the cloud. For achieving this, the businesses is in need of a *hybrid* solution [20]. Microsoft have tried to build their cloud solution around this *hybrid* concept, and hopes to attract businesses that requires this. Amazon does not, at the current moment, support a hybrid solution, and most of the Amazon Web Services (AWS) resides in Amazon's public cloud [20].

Figure 3.2 does show an example of prices from the three different cloud providers. This is a very general comparison, but it is too many variables to consider, so a conclusion cannot be made deciding which is pricier than the other. When a solution becomes complex, with many different services running, it will be difficult to estimate the total cost of the solution. All three providers does have a pricing calculator, so that the user can compare them to each other, and the result will depend on which services the user selects [22] [23] [24].

When looking at how the user should administrate the data and code in the cloud, there are some major differences. Which of the solutions that are *best*, will depend on the users' needs. Azure does provide an online management and portal service, where the user easily can organize and manage the code and database. This solution will be the best choice for many people, but some would feel some lack of control over the operations. Google does not have the same, simple administration features. One example is running Hadoop [21] on the *Google Compute engine* requires the user to download a Hadoop package, along with some other packages, to deploy the cluster. Amazon does provide some assistance with deploying applications on the AWS platform, but in addition to this the user can be certified within different parts of AWS. For example they provide tutorials and exams to be a certified *SysOps Administrator* or *Developer* [25].

3.2.6 The Fog

Fog computing resembles cloud computing in many ways. It provides storage, data and computational power. The services it provides is located between the end dives, for example smart phones and tablets, and the traditional cloud computing data centres. What distinguishes it from cloud computing is the distance from the users. Data can be processed mostly in locally, in smart devices, instead of the computations being done in the cloud[22]. Because of this we can say that a *fog* is a cloud close to the ground [23]. It is placed at the network edge, close to the user, and it has a dense geo-distribution. This

22. http://calculator.s3.amazonaws.com/index.html
23. https://cloud.google.com/products/calculator/
24. https://azure.microsoft.com/en-us/pricing/calculator/
25. http://aws.amazon.com/certification/

reduces the latency for the user and improves QoS[26] [23][24].

3.3 Positioning techniques

This section will present some techniques that are most commonly used in indoor positioning systems. In radio-frequency (RF) based systems, there are six categories of approaches that can be used, and they are *Time of Flight (ToF)*, *triangulation*, *trilateration*, *fingerprinting*, *cell-based* and *filter-based*.

- *Triangulation* uses angles to determine the position of an object. The position is based on the *Angle of Arrival (AoA)* of wireless signals. To estimate the position of an object atleast three reference points are needed, and the angles between these points are known [25]. From mathematical equations the position of the object (cable drum) can be estimated.

- *Trilateration* calculates the position by measuring the distance from atleast three reference points [25]. These reference points have a fixed distance between them. The distance between the *reference points* and the cable drums are estimated based on the wireless signal parameters.

- When using a *fingerprint*-method the area is divided into a grid. It consists of two phases, a learning (offline) phase, and an online phase. In the offline phase the *fingerprints* are created. These prints are unique for each node in the grid, and often consists of RSSI measurements. During the online phase, the moving objects (cable drums) receive the RSSI measurements from the reference points. From the database the moving objects is able to retrieve the *fingerprints* and these are compared to the new RSSI measurements. The moving objects positions is registered after analysing the difference in RSSI measurements [26]. One advantage with using fingerprinting is the accuracy that can be achieved, but the disadvantage is that the offline phase is very time-consuming and difficult to maintain.

- The *cell-based* positioning method is quite similar to the *fingerprint* method. It has an offline/learning phase where (in this case) the smart phones are placed at strategic locations at the factory, and they are registered in a database. In the online phase these smart devices read any signals from the surrounding beacons and stores the MAC-address of the

26. Quality of Service, or QoS, is the overall quality of service seen by the users of the computer network

beacons. The set of stored beacon addresses are compared to previous entries, and the best match is selected.

- Within the *filter technology* there are two filters that are commonly used, the Kalman filter [27] and Monte Carlo filter (particle filters) [28]. The *Monte Carlo* filter uses an algorithm to estimate the position of an object as it moves. By having a map over the environment and the algorithm it uses a particle filter to represent the most likely positions, where each particle is one possible position [29]. This method is often used in robots when they move around inside an area. For the robot to move around a space it needs to know the surroundings and its orientation [30].

- *Time of Flight (ToF)* works by estimating the distance between two objects by counting the time it takes for a signal to go from one object to the other. This method requires the object's clocks to be synchronized with high precision. Because a small difference between the two clocks means a significant distance between the two objects, this method is often used when the distance between objects are hundred meters or more [27].

Research have mostly focussed on the use of RSSI measurements, triangulation and triateration. RSSI measures are used in the *BLE User Scanner* where it represents a number which tells the user if he is getting closer or further away from the beacon. This thesis also uses the *cell-based* method to create a grid system within the plant and to plot the cable drums within this grid.

3.4 Bluetooth

> *Bluetooth* technology is a wireless communications technology that is simple, secure and everywhere [28].

The first section will give an intro to the *Classic Bluetooth* technology, while the second section will present *Bluetooth Low Energy (BLE)* and its specifications

27. http://www.rp-photonics.com/time_of_flight_measurements.html
28. http://www.bluetooth.com/Pages/Basics.aspx

3.4.1 Classic Bluetooth

Bluetooth was created by Ericsson in 1994. The technology operates on frequencies between 2400 and 2485 MHz. It uses FHSS[29], which have a *hopping carrier* that will prevent interference between devices. The Bluetooth transmission will only remain on one channel for a certain amount of time before *jumping* to another channel. If there is interference between two or more devices, the message from one device will retransmit later when the device have switched to another channel [31]. Bluetooth, FM radio and television all send data over wireless. One of the specifications that distinguish Bluetooth from other technologies that uses radio waves, is that the range of Bluetooth is only within a devices Personal Area Network (PAN).

Millions of COTS[30] devices support Bluetooth, ranging from computers, smart phones, medical equipment and television controls. It was conceived with the intention of removing the use for cables between devices [31].

Range and positioning

According to *Bluetooth Special Interest Group (BSIG)*, Bluetooth is not about getting the exact location, but more about proximity[32]. And therefore the more correct term would to call it a proximity solution, and not a positioning solution.

Bluetooth was designed to be a short range technology, with a range of only a few meters. However, today's Bluetooth have achieved a range of up to 100 meters [33]. Depending on the range the different devices has, they are divided into three classes, as shown in figure 3.3. This advertised range is done where the signals from devices are not hindered by other objects. There are many different factors that can give the devices a much lower effective range. This can be obstacles, like cable drums, machines and walls, interference from other devices and machines, and signal reflection.

29. Frequency Hopping Spread Spectrum
30. Commercial Off-The-Shelf
31. http://www.bluetooth.com/Pages/Fast-Facts.aspx

Class	Output power	Maximum Range
1	20 dBm	100 meters
2	4 dBm	10 meters
3	0 dBm	1 meters

Figure 3.3: A comparison of different Bluetooth classes, with their estimated range and output power[34]

Discovery and pairing of devices

For two devices to be able to send and receive data between each other a bond, also called *pairing*, have to be set up first. There are two different roles that the devices can take, and those are *master* and *slave*. Each master can have 7 simultaneous connections to different slaves. To initiate this bond between the master and its slaves the master will emit a *inquiry message*. The slaves within proximity, and which are *discoverable*, will respond to the *inquiry* process with its name, and a connection can be established [34][35].

3.4.2 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) was developed by *Bluetooth Special Interest Group*, and was presented in 2011 as Bluetooth 4.0. The intention of BLE was short range communication with low power consumption. It built upon the *Classic Bluetooth*, but includes some new features, which is presented below [32].

Figure 3.4a shows the protocol stack of BLE. The figure 3.4b shows the structure of the different fields of each layer. It consists of two main parts, the *controller* and the *host*. Within the *controller* lies the *link layer* and the *physical layer*. The *host* includes upper layer functionality and and it runs on an application processor [36].

Link layer The link layer of BLE is a state machine. The five states of this state machine is *advertising*, *scanning*, *initiating*, *passive* and *connective*. The connective state includes the roles master and slave, and the scanning part includes the active and passive substates [36].

Physical layer BLE works in the 2.4GHz ISM band, it has 40 channels and three of these channels are advertising channels. The remaining channels

32. http://www.bluetooth.com/Pages/low-energy-tech-info.aspx

are data channels. The packets between the channels are either advertising or connection events. The advertising channels are used to discover other devices, establish a connection between devices and broadcast transmission. The data channels are used for bidirectional communication between paired devices [36].

L2CAP, ATT and GATT The L2CAP, or logical link control application protocol, is a protocol based upon the classic Bluetooth L2CAP. Its main goal is to multiplex data of the three higher layer protocols [36].

The ATT is what defines the communication between the server and client. GATT defines the framework that uses ATT to discover other devices. GATT also controls the exchange of device characteristics from one device to another [36].

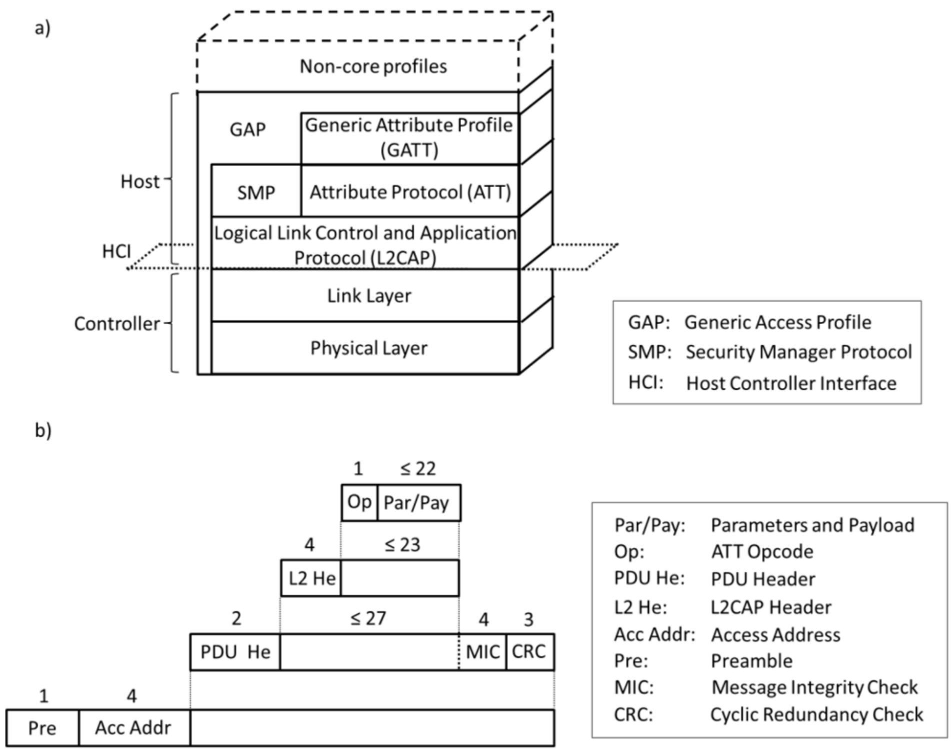

Figure 3.4: The Bluetooth Low Energy protocol stack[36]

Even though the *controller* inherits from the classic Bluetooth Controller, they are incompatible. This means that an application which only implements *classic Bluetooth* support, cannot communicate with a device that only supports *BLE*, and vice versa. A device that supports both the classic controller and BLE controller is called *dual-mode* devices [36][37].

BLE Modes

BLE has four modes: technology, central, peripheral, broadcaster and observer. The *observer* mode compliments the *broadcaster* mode, and the *central* role is similar to the *master* role in *Classic Bluetooth*. The *central* role is more advanced and designed for devices like smart phones, and it can manage more connections from multiple *peripheral* devices [36]. Previously, two devices had to be paired to exchange information, in the new mode the devices can be set to *broadcasting* mode, and data can be sent through the advertisement channels. For this thesis the *broadcasting* mode of the BLE beacons is used. The smart phones used as *reference points* will pick up the signal and data (MAC-address) from the different devices and this is added in the database.

Range BLE Beacons

The range of BLE devices are, as *Classic Bluetooth*, dependent on the broadcasting power, obstacles and interference from other devices and machines. Most BLE beacons operate with a 0 dBm transmitting power, and this have an expected range of 50 meters. Different beacon providers advertises with different range and expected battery consumption. These numbers are showed in figure 3.7, and we can see that most providers expect a range of about 50 meters.

BLE security

Communication over BLE is secure and protected, and the Bluetooth Core Specification (BCS) does provide multiple features to cover the encryption, privacy of user's data and data integrity [33].

To ensure secure communication between two Bluetooth devices, pairing is used. When two devices pair up, their identity set up the encryption keys and the devices are ready to exchange information. They key generation is performed by the host, and Bluetooth uses AES-CCM [34] cryptography. In BCS version 4.2 security got enhanced by using the Elliptical Curve Hellman-Diffie (ECDH) algorithm for key exchange between devices. BLE also includes a privacy feature, which reduces the ability to track a single BLE devices over

33. https://developer.bluetooth.org/TechnologyOverview/Pages/LE-Security.aspx
34. https://asecuritysite.com/encryption/ccmaes

longer period of time. This feature will hinder outsiders to track one particular cable drum, and possible steal it when it is located in a less protected area.

3.4.3 BLE Beacons

From testing and research done in the pre-projects leading up to this theses, two beacon providers stood out as the most promising. Those are the *Estimote* and *StickNFind Pro* beacons.

StickNFind Beacons

StickNFind (SNF) have several different types of beacons for sale, as well as some that have not been released yet. In this project, which includes the two pre-projects and this thesis, two beacons from SNF have been tested. These are the SNF Consumer edition and the SNF Pro [35] [36]. StickNFind have also included a SDK[37] to help developers in creating new applications.

These two beacon types differ somewhat in specifications, and they both have pros and cons.

StickNFind Consumer Edition The biggest advantage with the SNF Consumer edition is that the battery can easily be changed while stuck on a cable drum. When the beacon runs out of battery Nexans does not need to invest in a new beacon and configure the beacon, they only need to change the battery.

The disadvantages does however prevent this beacon from being used at the plant. Firstly, the expected battery life is not sufficient. The expected battery life is approximately one year, with a daily use of 30 minutes. With up to 2000 beacons at Nexans, this short battery life would create significant maintenance on the beacons.

Another disadvantage is the lack of waterproof case. Many of the drums are soaked in water for testing, and it would not be possible to remove the beacons before every soak to prevent it from being ruined.

StickNFind Pro The StickNFind pro have an expected battery life of approximately three years. Since Nexans wishes to have regular maintenance on the

35. https://www.sticknfind.com/sticknfind.aspx
36. https://www.sticknfind.com/enterprise.aspx
37. Software Developing Kit

drums every two years, it would be possible to include beacon management as a part of the regular management.

This beacon, as opposed to SNF Consumer Edition, cannot be changed battery on without compromising the seal that keeps it waterproof. This means that they have to change the whole beacon every two-three years.

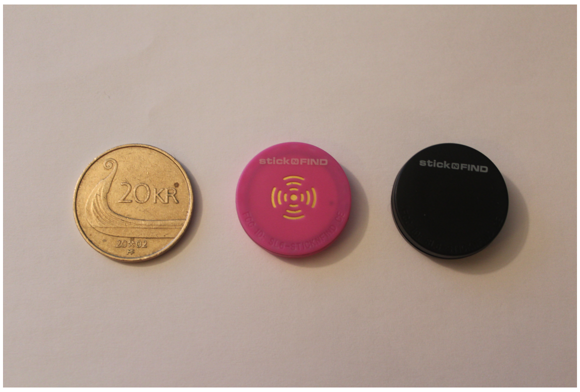

Figure 3.5: Figure showing the original StickNFind (pink) and the StickNFind Pro (black) compared to a 20 NOK coin

Estimote Beacons

Estimote has the second beacon type tested in this thesis[38]. They provide waterproof beacons with an average battery life of three years. Estimote have also created an SDK to help developers create applications. The beacons are described as a *mote*, and as figure 3.6 shows, it is a large beacon compared to the SNF beacons.

38. http://estimote.com/

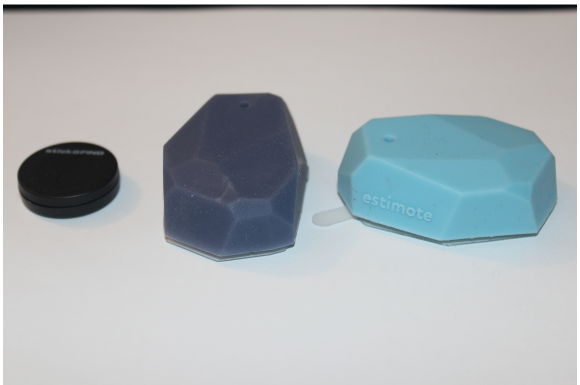

Figure 3.6: Figure showing two Estimote beacons compared to the StickNFind Pro beacon

Comparison of BLE beacons

In addition to the two beacons presented here, the pre-projects also tested two other beacon types. This was the *Easibeacons* and *RedBear* beacons. These two did not meet the waterproof requirement and were not as robust as *SNF Pro* and *Estimote*. Therefore they have not been installed and tested at Nexans.

Both *Estimote* and *SNF Pro* provide waterproof beacons with long battery life. The main difference in the beacons are the size. The *SNF Pro* beacons are considerably smaller than the *Estimote* beacons. This feature will prevent the SNF Pro beacons from being as easily destroyed by trucks and other machines. Both *Estimote* and *SNF Pro* is installed at Nexans, and both are still working now, one year after they installation. A figure showing the differences between the beacons are presented in figure 3.7.

	StickNFind	StickNfind Pro	Estimote	Estimote Stickers	Easibeacon	RedBear
Advertised batterylife	1 Year*	3 Years	5 Years**	1 year	1.5 Years	1 Year***
Waterproof	No	Yes	Yes	Yes	No	No
Advertised max range	30 meters	45 meters	40-50 meters	40-50 meters	70 meters	50 meters
Battery Type	CR2016	Unknown	CR2450	Unknown****	CR2032	2xAAA
iBeacon approved	Yes	Yes	Yes	Yes	Yes	Yes
Battery changeable	Yes	No	No	No	Yes	Yes

*30 minutes a day use
**default is 3 years
***4500 hours, operating 12 hours per day
****To be announced later on

Figure 3.7: A comparison of some of the beacons available, as well as one beacon (Estimote Sticker) that have not been released yet

3.5 NFC

Near Field Communication (NFC) is a technology based on RFID, and is used for contactless communication between two devices. The motivation for creating NFC was to make it easier to transfer content, connect devices and make transactions. As opposed to other existing communication technologies, the distance between the devices can not exceed 10 centimetres [38].

Most devices support NFC today. Apple does have a restriction on their NFC solution, but this is explained more in section 3.6.

NFC involves a *target* and an *initiator*. The initiator generates the RF signal, initiates contact and controls the data exchanged. The *target* will answer this request. It also differs between two communication modes, *active* and *passive*, where *active* is where both the target and initiator creates their own electrical fields [39].

NFC operates in three modes; Card Emulation, Peer-to-Peer and Reader/Writer. In Reader/Writer mode, the NFC device can read and write data to a tag. This mode is used in the NFC application implemented in this thesis. In Card Emulation, the phone acts as a physical object. Examples are house keys and bus cards. In the last mode, Peer-to-Peer, data can be transferred between two NFC devices [38].

When trying to create a technology that stores personal data, like payment card information, how secure the technology is often the most discussed and researched topic. Limiting the range can reduce the security challenge as the complexity of eavesdropping increases. Since NFC per default is backwards compatible with RFID, the NFC is not encrypted. Encryption may be implemented as an extra feature for certain applications. An example is the AES encryption, which is implemented in many applications to prevent eavesdropping, *man-in-the-middle* [39] attacks and data manipulation, corruption and insertion.

The NFC application have to be stored on either the SIM card, the smart card or on the smart phone itself. By storing the application only on the SIM card, this will be the secure element for the authentication protocol. This will also create simpler portability between devices [39].

An example where NFC is used is a payment solution created in collaboration

39. Where the two devices believes that they are communication directly with the other, but in reality a third-party device is listening on the data sent and received

with Telenor and DNB [40]. The payment terminals contains support for NFC, and the user's smart phone act as a VISA card. The smart phone and SIM card has to support NFC, the user needs a VISA mobile card and the smart phones needs an application, *Valyou*, installed.

In this thesis NFC is used to identify the cable drum. This is to discard the piece of paper stuck on the cable drum which gives the user an indication of the status of the drum. Due to most of the cable drums tested is made of steel, special NFC tags have been used. These are designed to be placed on metal objects.

3.6 Mobile Platform

There are three platforms mobile that are able to connect to BLE beacons, and these are Android, iOS and Windows Phone. Previous works have tested two of these platforms, Windows Phone and Android, and Android is the preferred platform. This section will present a summary of the tests and research results from the pre-projects.

Apple iOS Apple iOS was, alongside Android, early with supporting Bluetooth 4.0. In addition to supporting Bluetooth 4.0 they created a framework which they call iBeacon. The iBeacon framework extends the location services in iOS. Apple devices newer than the iPhone 4s, iPad 3. Generation, iPad mini and iPod Touch 5. Generation supports Bluetooth 4.0. By using an iBeacon, the iOS unit can notify the user when he leaves or enters an area. In addition to this, it can estimate the distance from the user and an location, for example a particular shelf in a store, and this is estimated using RSSI measurements [41]. Not all beacon providers are approved as iBeacon providers. Apple have a set of requirements for the providers, which they have to pass, to become a licensee. These requirements will provide the user with a consistent and reliable experience. Both StickNFind and Estimote are iBeacon approved.

Previously Apple devices did not support NFC (Near Field Communication), but the new iPhones (iPhone 6 and iPhone 6 Plus) that was released in September 2014, does support this technology. In addition to NFC support, they also introduced *Apple Pay*, which is a contact-less payment solution. NFC on the new iPhones are currently on *lockdown*. This means that they cannot be used for anything more than *Apple pay*. In this thesis it would not have been possible to read the NFC tags placed on the cable drums because of the *lockdown*.

40. http://www.online.no/trender/nfc-betal-med-mobilen-i-butikken.jsp
41. https://support.apple.com/no-no/HT202880

Windows Phone Microsoft have not been as early as Apple iOS and Android in supporting BLE, but this support was implemented in Windows Phone 8.1. In previous works a *proof of concept* using Windows Phone and BLE researched to see if an implementation on this platform was possible. Research showed that there were some challenges with using Windows Phone. An application running on Windows Phone is only able to see and list beacons that have previously been paired with the device. As Nexans will have more than one thousand beacons at the plant, and due to the pre-pairing requirements, an application running on Windows Phone is not a viable solution.

Android Android was early in supporting BLE, and it was a built-in platform support for Bluetooth 4.0 in Android 4.3. It provides APIs that can be used by applications to discover devices and query for services.

3.6.1 Comparison

Apple and Android have both been early in supporting Bluetooth 4.0 on their devices. Due to restrictions on Nexans, Apple iOS is not an option in this thesis. For systems that does not have this restriction iOS might be just as good a choice as Android.

As presented, Windows Phone have the disadvantage of the phone having to *pair* with each beacon to be able to see them in an application. This excludes this platform from using it at Nexans.

Since neither iOS nor Windows Phone is possible, further research on which platform to use have not been done. Further requirements from Nexans is that the smart device should, if possible, have a certification of IP67. There are two certificates that are common for robust smart phones, and they are IP54 and IP67. The first number indicate how dust proof the phone is, and the second number is how waterproof it is. The higher number, the more sustainable the phone is. A phone with a certification of IP54 does not guarantee that it is dust proof, and it only supports some splatter of water, for example rain. A phone with the IP67 certificate is dust proof, and it should not take damage of falling into freshwater, and it can be left in water for approximately 30 minutes. Three examples of smart phones that does have the IP67 certificate is the Huawei Ascend Y530, Galaxy Cover 3 and Motorola MOTO G. For this thesis the Huawei Ascend Y530 and Samsung Galaxy S3 is used for testing.

3.7 Summary

This chapter have presented technologies used to create the system implemented in this thesis. It shows advantages and disadvantages with the different technologies, and presents the reasons behind choosing the technology that is used. Each parts of the system are good standalone technologies, but combined they provide specifications that can support the technical requirements from Nexans.

The hardware used, which includes Android smart phones supporting Bluetooth 4.0 and NFC, NFC tags for use on the steel drums and BLE beacons for use in a harsh environment at Nexans are easy to acquire by Nexans. They are also easy to set up, install and use for the workers at Nexans.

By using a cloud provider, and not a local server, the need for hardware and software while testing the system is not needed. By using Azure, the IT personnel does not need to acquire new knowledge about a new platform. The user interface in Azure is also simple to use, and have the required functionality for administrating the system.

4

Related work

This chapter is divided into two sections. The first section presents work done within the field of Bluetooth Low Energy. The other section presents other indoor tracking systems, and the technologies used in these.

4.1 Systems using Bluetooth Low Energy

The BLE technology is a technology not only meant for indoor tracking, and this section will present other user cases for BLE. Many different services use BLE as a tool for collecting data, where some of these are presented in figure 4.1. The health care and sports industry, presented in this section, are two of the largest markets for BLE today.

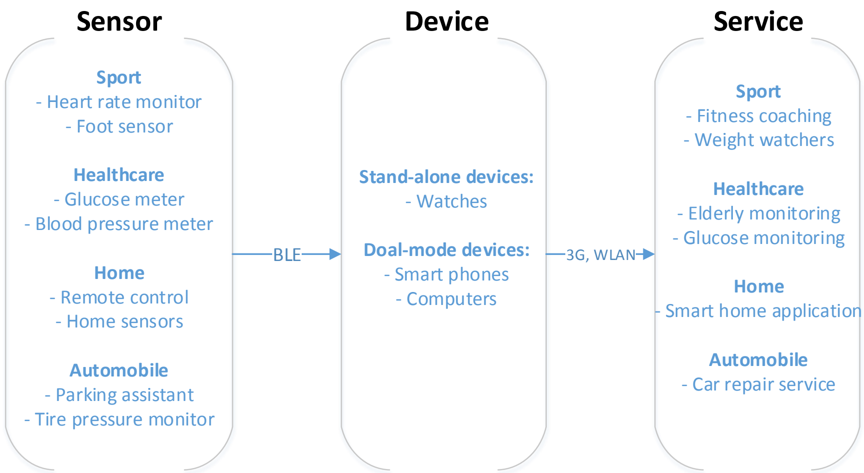

Figure 4.1: Some different uses of the BLE technology

4.1.1 BLE within the health sector

Many services within the health care industry have started using the BLE technology as a tool to improve their work flow. A major challenge in the health care industry is security. The medical data is often confidential, and the data should not be accessed by outsiders. One advantage with BLE is that it adopts the security features of classic Bluetooth. In addition it is possible to add features to prevent outsiders to read and alter the data. This is done by limiting the ability to track a transmitting device by using a random device address that is changed frequently [40].

Blood Glucose Monitoring

More and more people are developing diabetes, and diabetes causes different health care issues like blindness, arterial disease and amputation of limbs [40]. To prevent these health care issues good monitoring and management of the blood glucose (BG) levels are needed. The BG levels needs to be kept at a reasonable level, and this is measured by a BG meter.

By using BLE buildt into a BG meter these measurements will be made simpler to use for the user and the health care providers. The user will measure the BG levels as he normally does, and the BG meter will send the results to the users cell phone and/or the users medical physician. By storing this data the user and physician can see trends and levels over time, and can take any precautions to prevent a spike in BG levels [40].

4.1.2 BLE within sports equipment

There are a vast number of exercise equipment that uses BLE. Due to the long lasting battery, the user does not need to charge the device as often as when using the classic Bluetooth technology. Older models of sports devices that were to communicate digitally as well as having long battery life have been using mostly ANT+ [1] or WIND[2]. The issue with using these protocols have been that almost no devices, like tablets or smart phones, have ANT+ chips built in, and therefore a connection between a smart phone and a heart rate monitor have been impossible without an extra adapter [3].

By using BLE different devices can be paired and are able to work together. The information stored from foot pods, heart rate monitors and location services can be stored and analysed together for a more personalized exercise log.

4.2 Tracking systems

Tracking, both indoor and outdoor, systems have become increasingly more popular. The process of tracking systems is called geolocation, position location, radiolocation or location sensing.

4.2.1 Different tracking technologies

This section will present different tracking solutions available. A comparison between them and BLE will be done, as well as present how they would work indoors in the environment at Nexans Rognan.

In figure 4.2 a comparison of different technologies are listed with regards to their ability to send voice, data, audio, video or state.

1. The AND+ protocol supports interoperability between devices like heart rate monitors, bike computers and power meters
2. WIND is Polar's equivalent to the ANT+ protocol
3. http://www.dcrainmaker.com/2012/07/the-current-state-of-bluetooth-smartlow.html

	Voice	Data	Audio	Video	State
Bluetooth ACL / HS	x	Y	Y	x	x
Bluetooth SCO/eSCO	Y	x	x	x	x
Bluetooth low energy	x	x	x	x	Y
Wi-Fi	(VoIP)	Y	Y	Y	x
Wi-Fi Direct	Y	Y	Y	x	x
ZigBee	x	x	x	x	Y
ANT	x	x	x	x	Y

Figure 4.2: A comparison of different technologies and their uses[41]

GPS

The Global Positioning System (GPS) is one of the best known tracking tools today. It is an *outdoor* tracking system, which can be used to track moving objects, such as cars, planes or people [42]. The targeted objective, whether it is a car or person, have to be carrying a GPS receiver for receiving radio waves. These waves are received from multiple satellites and the position is determined based on the data received using triangulation etc..

Due to the need of receiving signals from satellites, GPS does have limitations in certain environments. One of the limitations are *GPS signal reception*. For the GPS to function properly it needs an undisturbed signal reception from a minimum of four satellites. Known objects that obscure the signals are water, soils and walls [43]. The second limitation are *signal integrity and accuracy*. The measurements from the different satellites are unique receiver solutions, but some of these measurements can be wrong. When a satellite position is wrong, the calculation of the position will also be wrong. The GPS receiver will measure the time for a signal to travel from the satellite to the receiver. These measurements are corrupted by different errors, and the effect of the errors are calculated and removed from the result. The errors can be satellite errors, signal propagation errors, which is that the earth's atmosphere changes the speed of the signal, and receiver errors [43][44]. Since Nexans Rognan mostly keep their cable drums inside a factory building, with walls and machines that will disturb a GPS signal, this technology is impossible to use and an indoor tracking solution have to be implemented using other technologies.

Indoor tracking systems is used for many different things, such as tracking items in a warehouse, detecting different personnel or equipment or tracking elderly people. The primary progress of indoor tracking have been done the last 10-15 years, and both the industry and academia are involved in developing new ways tracking items and people.

Both GPS and BLE can be used to determine the position a device, but there are some major differences between them. Since GPS uses satellites to determine the location, it will work better in an outside area, but loses accuracy if it is exposed to interference from other objects. BLE does not have the same range as GPS, but it can be used as a proximity tracking system for example if the user enters the area of a particular store, and the user will get exclusive offers on his mobile device. Another difference is that GPS uses much power compared to BLE, and the user will often turn of GPS when it is not in use. BLE on the other hand goes into sleep mode when it is not in use, and therefore does not consume as much power [45].

ZigBee

From the ZigBee Alliance, which is an alliance of commercial companies of for example Philips, Texas Instruments and Silicon Labs [4], ZigBee was designed. It is a low-power technology which, most often, runs in the 2.4 GHz band and can be powered by coin-cell-type batteries. The range can extend up to approximately 200 meters, but the typical range is 50 meters. For many systems, the main disadvantage with ZigBee is the low bandwidth of only 250 kbps. For the system at Nexans this would not have been an issue because of the small amount of data transferred.

It provides a low-cost system with reliable data transfer, short range operation and appropriate levels of security [46]. A ZigBee network does require a *full function* device to operate as a network coordinator, but most of the devices can have reduced functionality, which means that they for example only emit the MAC-address of the device. Figure 4.3 shows how an example of how a ZigBee network can be built up. This figure is a *star* where all data goes through the coordinator. Other designs can be that the fully functional devices can receive and send data as well.

- The *Network Coordinator* sets up the network, often operates in the *receive state*, transmits network beacons and manages and stores information about the nodes.

- The *Nodes* in the network are designed for low power usage and will sleep for periods when it is unused and it will transfer data when needed.

4. http://www.zigbee.org/zigbeealliance/our-members/

To provide the security needed ZigBee uses MAC layer security [47]. The MAC layer uses AES[5] as the core cryptographic algorithm.

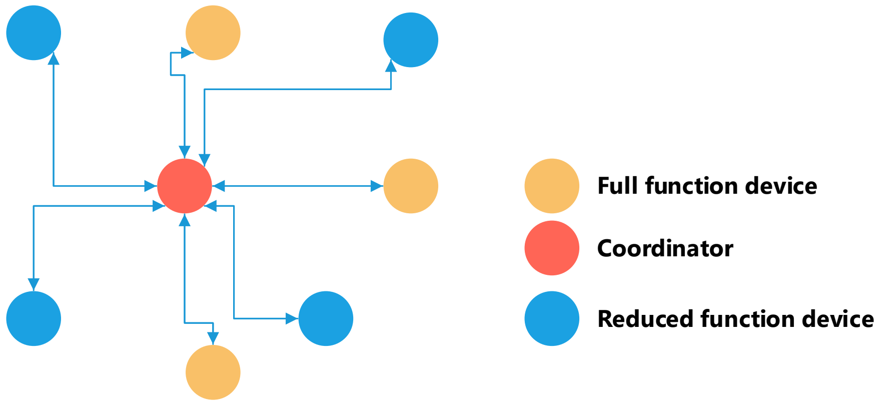

Figure 4.3: A figure showing how the ZigBee network is built

ZigBee could have been used to track cable drums within Nexans, but was not chosen for a couple of reasons. Bluetooth is a more commonly known protocol, where a lot of support, guides and research can be found when developing a Bluetooth application. Bluetooth is supported in most of the COTS [6] units today, and this gives better opportunities to continue to develop the system and change units as newer, more powerful, and less power consuming devices enters the market.

Wi-Fi

Wi-Fi have been a proven solution for connecting multiple computers. As opposed to many outdoor solutions, a Wi-Fi solution does not rely on time measurements to position the object. Wi-Fi will calculate the approximate position using the received signal strength from each object. The computers are connected through wireless local area networks. The disadvantages of a Wi-Fi solution is that it is relatively expensive to use. It also have high power consumption, and it does require frequent (depending on the battery capacity) recharges [48].

There are many differences between BLE and Wi-Fi. Range is one of them. Depending on the Wi-Fi protocol used, Wi-Fi can offer up to 650 meters range,

5. Advanced Encryption Standard
6. Commercial Off-The-Shelf

while BLE advertise the range to be approximately 50 meters. At Nexans the range of the BLE beacons have shown to be around 20 meters because of the amount of obstacles and interference from other machines.

Data transfer rates is another difference. Some of the latest specifications of BLE is that it can offer up to 25 mbps transfer rate, while Wi-Fi can transfer at a rate of 250 mbps. For other systems, that have to transfer large amounts of data, this would be a significant difference. But for the system implemented in this thesis, it does not have a significant impact.

Bluetooth does not offer much security, and is just limited to key matching [45]. This means that information that should be kept secret is not meant to be transferred via Bluetooth. But Bluetooth does support AES-128 encryption of the data. Wi-Fi on the other hand is secured by WPA2 [7] [45, 49].

Radio Frequency Identification (RFID)

RFID is a passive technology that only prompts a transaction whenever the object passes near a sensor [50]. It will consist of a RFID-tag, a transceiver, which decodes and interprets the data, and a scanning antenna that communicates with and gives power to the RFID tag.

RFID systems can either be *passive* or *active*. The *active* tags requires a power source, whilst the *passive* does not. The *passive* tags are small, and contains an antenna and a semi-conductor chip attached to the antenna. The RFID *reader* will power up the tag to read information from it. Further, RFID is divided into near-field and far-field RFID. Far-field would be the type used at Nexans, where tags capture electromagnetic waves from the RFID *reader*. The typical reader can emit signals to readers 3 meters away [51].

A system using RDIF for indoor tracking is *SpotON* [52]. The concept behind this system is having multiple base stations, each base station provides signal strength measurements to the central station. The central station calculates the exact position of the objects using triangulation [52].

One of the main differences between BLE and RFID is that the *passive* RFID-tag does not need to contain any batteries, since it is powered by the antenna nearby. By using RFID-tags they can last for years, whilst BLE beacons run out of battery after a couple of years. One of the major disadvantages with RFID is the limited range, with only a 3 meter typical range without disturbance from objects. More powerful readers could extend the range, but compared to

7. Wi-Fi Protected Access

the expected range of 50 meters of BLE, RFID would not work as well inside Nexans.

Infrared (IR)

There are two kinds of IR motion systems, and those are active (AIR) and passive (PIR). *Passive* means that they will measure the change in energy of the surrounding area, and in this way detect objects moving. The *passive* method will not work at the Nexans plant since we wish to detect *what* objects (cable drums) that have moved, and not just that *something* have moved [8] [9].

The *active* IR have two implementations, the *proximity sensors* and *motion sensors*. The *proximity sensors* can be used to sense when a person's hands are close to a hand dryer in a public bathroom or to automatically open a garbage dispenser when a person gets closer to it. *Motion sensors* will emit an IR-beam, which is received by an IR receiver. The disadvantage with IR is that the beam can easily be hidden by other objects, so that the position will not be registered [10].

The main difference between IR and BLE is how the signal is distributed. IR is a beam, which have to be in line of sight of the receiver. With an environment like the one at Nexans, where there are large obstacles and the drums are often stacked in clusters like figure 2.2, a solution where the receiver and transmitter have to me in line would be hard to maintain. When using BLE, the signal can reach the receiver even though the beacon and receiver is not in line of sight of each other.

LTE

LTE Direct provides device-to-device communications over cellular connections. It can have a range of up to 500 meters, supports thousands of devices, and is designed to consume little power. It uses radio signals, and they can set to *private* that only certain devices or people are able to receive the signals, or they can be *public*, so that everyone can receive them [11] [12]. By using LTE Direct the developers and users do not need extra beacon hardware, as in Bluetooth,

8. http://www.facilitiesnet.com/lighting/article/Occupancy-Sensors-Passive-Infrared-Ultrasonic-and-Dual-Technology-Facility-Management-Lighting-Feature–9608
9. http://www.digikey.com/en/articles/techzone/2012/jun/sensing-motion-with-passive-infrared-pir-sensors
10. http://www.guardall.com/product.asp?PageID=245
11. http://arc.applause.com/cards/lte-direct-future-beacons-explained/
12. https://www.qualcomm.com/invention/research/projects/lte-direct

but the cellular device becomes the beacon.

It is created as a competitor to BLE, but it supports longer range. It has somewhat other use cases. Examples are for a user to discover a particular store, to receive offers from a store or to notify the user if a friend is nearby. Since the LTE supporting smart phone have to be used as the beacon, this technology would not work in the use case at Nexans Rognan, since a cable drum will have to have a LTE device attached to it.

4.2.2 Example systems

This section will present some working tracking systems, starting with the first well-known system, *Active Badge*.

The first indoor tracking system created was *Active Badge* [53]. In the early 1990's, when this system was created, the way people connected and located other people was via a *pager*. The pager system works by sending a signal to a receiver, and the receiver will produce sound to alert the pagers owner that someone wants to contact him, and the pager had a display showing who the caller was. One drawback with the paging system was that the caller did not know whether or not the callee received the message or not. Around this time the way people located others were by using key cards which registered when a user entered or left a room or building.

When *Active Badge* was applied the users wore a badge, and this badge emitted an unique code every 15 seconds using infrared signals [53]. The signals are collected and sent to a *master station* so that the positions registered can be read by others. The *Active badge system* is quite similar to the BLE tracking system created in this thesis, but it does have a couple of drawbacks. It has only a range of approximately six meters, and the signals cannot pass through walls. This would mean that the grid system at Nexans would have to have very small masks. The system is quite costly in terms of installation and maintenance. It also performs poorly in bright environments, for example in direct sunlight [54]. The last drawback is that when there are many badges present within the range of a *station*, the signals collide and some will be lost [53].

RADAR [54] is another tracking system using *radio frequency* (RF) instead of IR. It is the Wi-Fi signal-strength based indoor positioning system. This system works by recording signals strength information at multiple base stations. These signals are interpreted using different methods, and the location is stored for later use. It works using a map, which is a lookup table holding collections of *packet signal strengths* and where in the building these where measured. The location is calculated using these reference points. *RADAR* also takes into account any environmental challenges in the different areas inside the building [54]. *RADAR* is able to calculate the object or persons location with an accuracy of a few meters. A system using RF instead of IR is less prone to disturbances from different objects, such as walls or machines.

LANDMARC is a system based on active RFID [50]. The purpose of the research done was to investigate whether or not RFID could be used in an indoor tracking system. In *LANDMARC reference tags*, or reference points, are placed at given locations. Signal strength measurements from the *moving objects (eg. cable drums)* are compared to the reference points, and the location is calculated. This will limit the cost of adding more expensive RFID readers, and instead use more of the less expensive RFID tags. This system require signal strength information from each tag to the readers. These signals strengths varies depending on environmental challenges in each area, and to increase the accuracy of the readings and measurements, more RDIF readers are needed. So when implementing *LANDMARC* in an environment, for example Nexans Rognan, the desired accuracy have to be weighed up against the costs of expensive RFID readers.

4.3 Summary

As the first section, about other user cases using BLE, presents, the BLE technology is a versatile technology that can be used to solve many issues in different industries. In this thesis, BLE is only used as a tool for tracking cable drums. If requirements change from Nexans more data can be collected from the beacons. Examples are temperature changes on the cable drums. They can also use the technology to track and analyse data from other elements at the plant, like tire pressure monitors on the trucks and temperature sensors at different locations.

The second section presents other technologies for tracking, as well as some systems using these technologies. A comparison of the different technologies, with the Nexans environment in mind, have been done. BLE is one of two technologies that could have been used at Nexans, and the other is ZigBee. BLE have been chosen for its easy accessible BLE beacons and easy set-up.

5

Design

5.1 Architectural Overview

The *Nexans Tracking System* is a loosely coupled modular system, consisting of three mobile applications, a RESTful middleware and a SQL database. Loosely coupled systems has some important benefits for Nexans, including modularity which allows for easily upgrading the modules that the system consists of. An overall design of the system is shown in figure 5.1 and a more thorough description is presented below

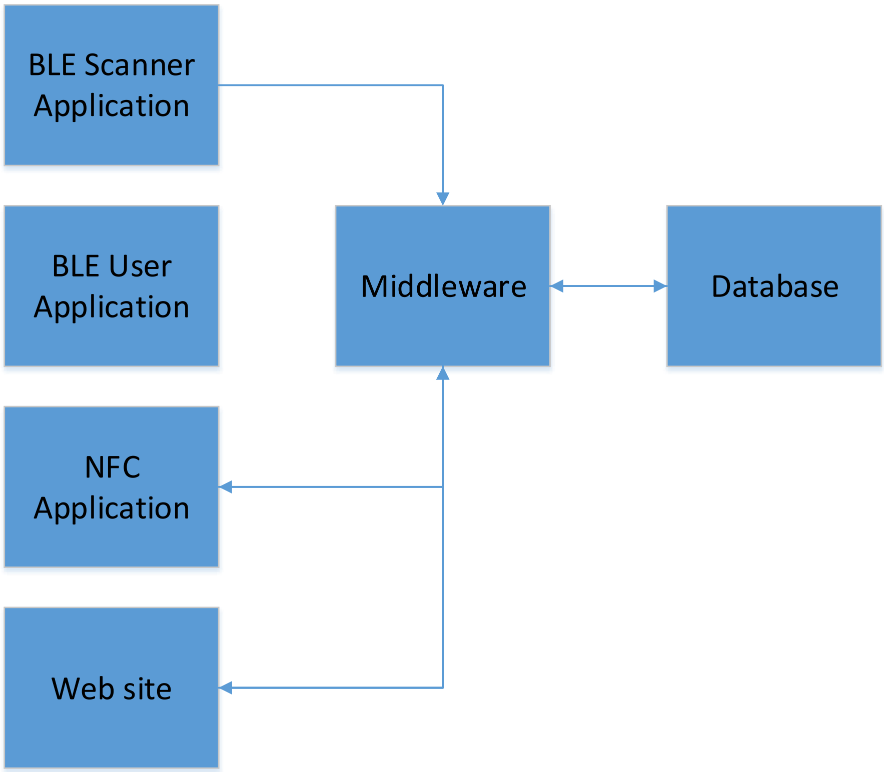

Figure 5.1: The overall design of the tracking system

One of the subgoals of this thesis was to implement a system for tracking cable drums inside the factory. This will limit the time spent searching for drums every day, and increase efficiency by knowing where every cable drum in the factory is located. By knowing the location of each cable drum, the down-time of each machine can be decreased. This increases the production rate, which will again mean that Nexans can accept more orders or are able to make shorter deadlines for customers.

The subgoal of tracking cable drums could have been achieved with only implementing the cell-based positioning system, including the *BLE Scanner Application* described below and a simple database containing only the names of the *reference points* and the cable drum number. This application was created in the pre-project, but was only used for testing how beacons behaved inside the factory.

However, that system would lack in areas such as user-friendliness, areas that are important for the users at Nexans. This creates a need for another goal: to create a user-friendly interface that Nexans can use in their daily production. This functionality addresses other issues regarding the cable drums. For example representing specification details or remarks to the drums.

The goal of this thesis is to be able to follow the cable drum from when it enters the factory the first time and until it is disposed. The system should contain functionality for tracking and maintaining all aspects of the drums life at Nexans, from production details, to maintenance and shipping. In other words: a fully functioning tracking system should be designed and implemented.

Each part of the system are created to expand the functionality of the system and together to create an easy-to-use, fully functional system for the administration and workers to use on a daily basis. A benefit of the system is the possibility to retire older, more inefficient operations, such as adding remarks to the drum or adding an order number to the drum ticket.

Mobile applications

- The first application is the *BLE Reference Point Scanner*. This application is installed on the *reference points* (smart phones) in the cell-based positioning system. It periodically scans the area for beacons, and reports any cable drums found to the middleware through a RESTful api. Most of the functionality of this application was implemented in the pre-project. The pre-project did not implement a tracking system for the cable drums, and the application was only used for testing the specifications[1] of the beacons.

- The second application, the *BLE User Application*, is used by the workers at the plant. After finding the approximate location of a cable drum using the web interface, the user can go to that area He can use the *BLE User Application* application to navigate within this area, and to locate the exact position of the drum. This application is not connected to any of the other modules of this system and does not require an internet connection.

- The third mobile application is the *NFC Drum Information Application*. There are cable drums located all around the plant, which either are empty or containing some raw material or product. The users does not always have control over each cable drum at the plant, and this application allows the user with a easier way of identifying the cable drum and the product that is coiled on the drum. When a user wishes to gain more information about the drum, both specifications and status, he can use

1. Range, advertising interval, robustness, etc.

the *NFC application* installed on his phone. By touching the NFC tag stuck on the drum, the user will be redirected to the cable drum's web site. He can then read and edit information about the drum (granted that he has the correct access rights).

Web site The web site is created to collect information from the database and present the information in an user-friendly way. More details about the different functionalities of the site is presented in the middleware section of this chapter.

Middleware The middleware will handle any incoming messages from the web site or mobile applications. It contains logic for deciding whether the user has access to view the page, or if a cable drum have moved and the position should be updated.

Database The database contains multiple tables with information about the cable drums, localization, user credentials and the reference points (smart phones) used. More details about the contents of the tables can bee viewed in Appendix B.

5.2 Mobile applications

The mobile applications are divided into three applications. The two BLE applications is built around the same code-base with minor modifications each. They are built upon the BLE scanner implemented in the pre-project.

5.2.1 BLE Reference Point Application

This application is used to create a grid inside the Nexans factory. An example of the grid is shown in figure 5.2. This application is created to implement support one of the subgoals of this thesis; to create a indoor tracking system for cable drums.

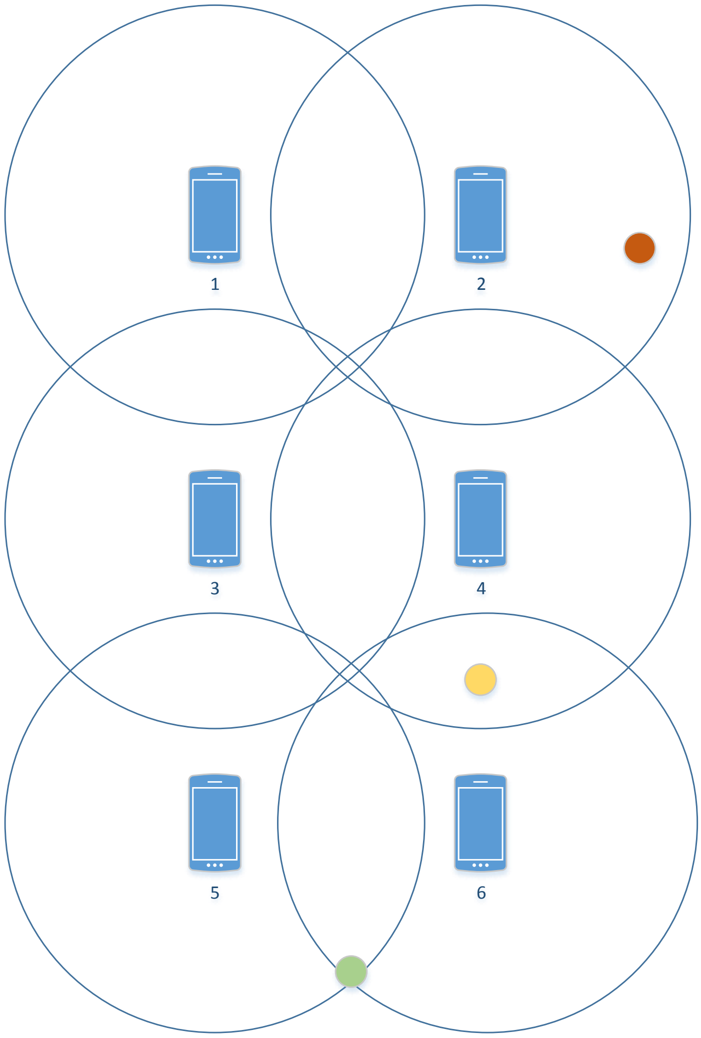

Figure 5.2: Example of cable drums within the grid

This application will, every two minutes, scan for nearby beacons. The beacons are placed on each cable drum, with a dependency between the MAC-address of the beacon and the drum id provided by Nexans. The two minute delay is chosen based on the behaviour of the drums, they are located at the same spot for extended periods of time. This means that continuous updates are not needed, nor is a more accurate time variable required by Nexans. By increasing the delay, the power requirements and data requirements lowered. The longer the delay is, the fewer database queries the middleware will have to perform. This delay can easily be changed in the future if needed.

The application generates a list, as seen in figure 5.3, and it is populated with the MAC-addresses of the nearby devices. The screen of the mobile device is never dimmed or turned of, to prevent the application of exiting and to make it easier to debug. Each of the entries in the list is sent to the middleware, which handles whether or not the position of the drum have changed.

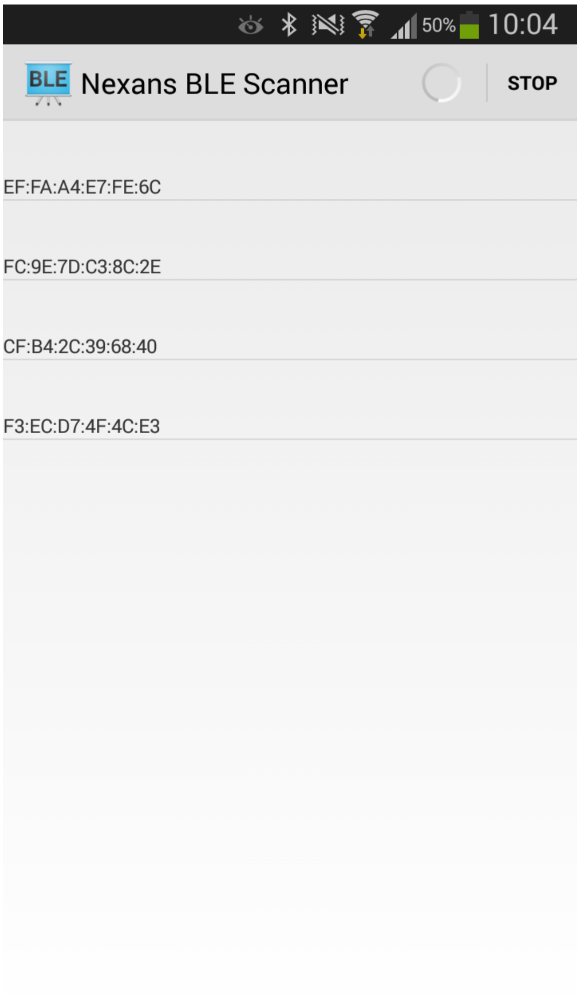

Figure 5.3: The BLE scanner used for scanning the area for beacons

5.2.2 BLE User Application

This application is run on the users phone and it has no connection to the middleware, the other applications nor the database. It is created to give an more accurate measurement of the distance between the user and the cable drum. The *BLE Reference Point scanner* gives a rough estimate of the position of the cable drum, while this application will create a more accurate position since the user will receive the RSSI measurements from the beacon.

From information found on the web page (populated by the *BLE Reference Point Application*), the user will know what area in the factory the cable drum is located. This can for example be in one of the storage tents, near the *WaterSoak* pool or near the laboratory. The *BLE User Application* is started on command from the user and scans the surrounding area for one minute and populate the list of drums found. If needed, for example if the users moves around, can the scanning be restarted.

When started, the *BLE User Scanner* will populate a list with the MAC-addresses of the beacons that are nearby. From the web page, shown in table one of figure 5.8, the user will know which MAC-address he is looking for. The user will locate the beacon address in the application and may move according to the RSSI measurements located underneath the MAC-address of the beacon in the list. This RSSI measurement will tell the user if he is close or far away from the beacon. If the user have an measurement of -100 dBm he is further away than if he had had a measurement of -50 dBm.

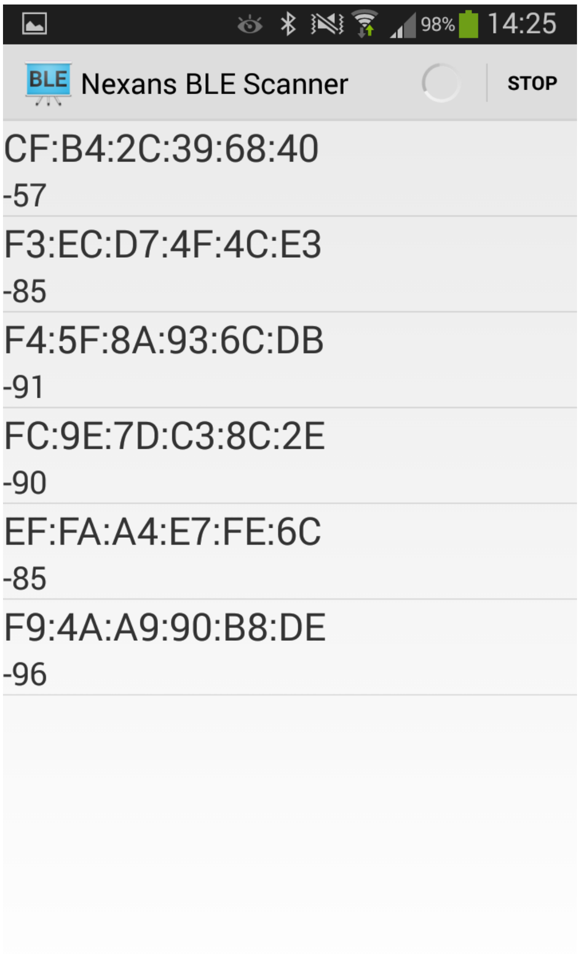

Figure 5.4: Screenshot of the *BLE User Scanner*, showing the beacon MAC-address and measured RSSI

5.2.3 NFC Drum Information Application

At Nexans, each drum currently has a paper ticket as shown in figure 2.1, stuck on the drum. This has information about the current status of the drum and the users have to update this when the drum enters a new state. This status includes for example how much rust the drum has and how clean the drum is. This paper ticket can easily be lost or ruined when the drum is exposed to different environment challenges. The ticket also has to be removed before the drum soaked in water. This is an inefficient and fault-prone method of keeping track of the cable drum, and it is difficult to keep track of which drums that should be taken into maintenance.

The NFC application created in this thesis should replace this method. By using NFC tags and the web site, it is simpler to update and keep track of the status of the drum. The NFC tags used in this thesis are water proof, and are specially made to be placed on metal. The NFC tag will contain the *drum id* of the cable drum, and by reading the sticker using the *NFC Drum Information Application* the user will be redirected to a web page containing information about the drum. If the user has the sufficient access rights, he will be able to edit the data on the web site.

5.3 Middleware

The middleware is what contains most of the logic in the system. It is divided into several parts.

User Accounts The *user accounts* consist of two parts, the *account* itself and the *user roles*. First the user have to register with name, email and password as shown in figure 5.5. A new user are then created in the system, but it has no user roles assigned to it. This means that they will not be able to see any sensitive information on the site, and the site will present the same data as when the user was not logged in.

When a user is assigned an *user role* he can see the data assigned to that role. The administrator have to assign the required roles to the user based on the work that the user is assigned.

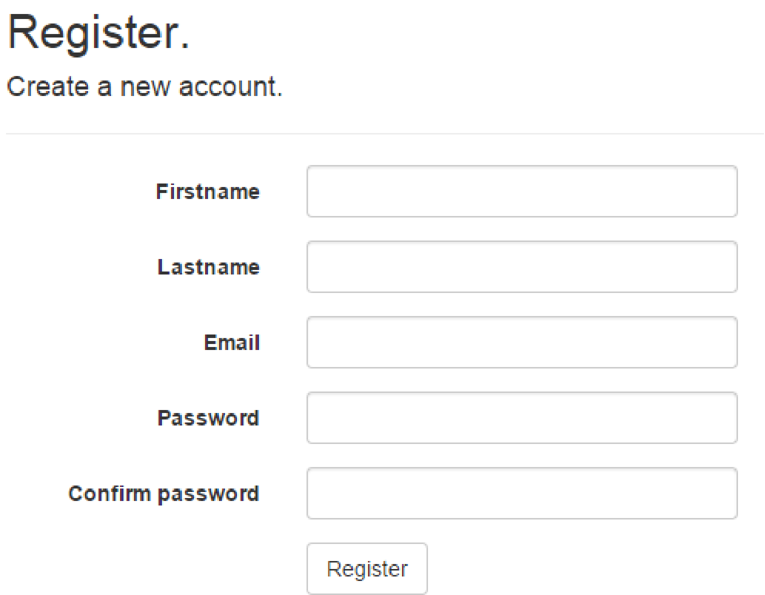

Figure 5.5: Register an account

In figure 5.6 the different access levels are shown. New access levels can easily be added, unused ones can be deleted and the existing ones can be edited. The different levels gives an indication on what the user can access in the system. The user role, *canEdit* is the administrator of the system.

Id	Name
01a9a474-1a17-4a96-b4a4-bc3d4e8f91f6	canEdit
7483443f-3b2a-4359-884c-71668c87f21c	MasterUser
b293168a-122d-4c50-a221-ed36f91bbe87	MinimumUser
96444ad5-78fb-4750-a2d0-35c8cd7fa032	SuperUser
335c87f8-3291-4826-9012-58d41ed457d7	User

Figure 5.6: The different access levels in *Nexans Tracking System*

Figure 5.7 shows the different user roles in the system, and what features of the system they can access. As of now the higher level user have the same accesses as the lower level ones in addition to some new ones. This means that the option of adding several user roles to an user is redundant in this thesis. For future work functionality can be added so that one of the lower user roles should be able to access some data, but not a higher user role. For example could the *MasterUser* not be able to edit maintenance history of a drum, but the *User* should be able to. This is why the functionality of adding more roles to a user have been implemented.

	CanEdit	MasterUser	SuperUser	User	MinimumUser	User with no role
Can add new roles to users	x					
Can add new drums to the database	x	x				
Can update spec of cable drums	x	x	x			
Can update maintenance history of each drum	x	x	x	x		
Can update general status of each drum	x	x	x	x		
Can see spec of cable drums	x	x	x	x	x	
Can see maintenance history	x	x	x	x	x	
Can see general status	x	x	x	x	x	

Figure 5.7: A table showing the different user roles in the system

The administrator account is created at first compile. It is not possible to create another administrator account runtime, the administrators of the system have to be specified in the code. The administrator have access to all data in the database, and is the only user that can add *user roles* to the users registered.

When the administrator logs in on the site, he will have access to the different functionalities implemented, but the main role of the administrator is to give other users the correct access rights. Figure 5.8 shows the home page of the administrator. The bottom table, *list of users with no role*, shows the users that still not have been assigned any *user roles*.

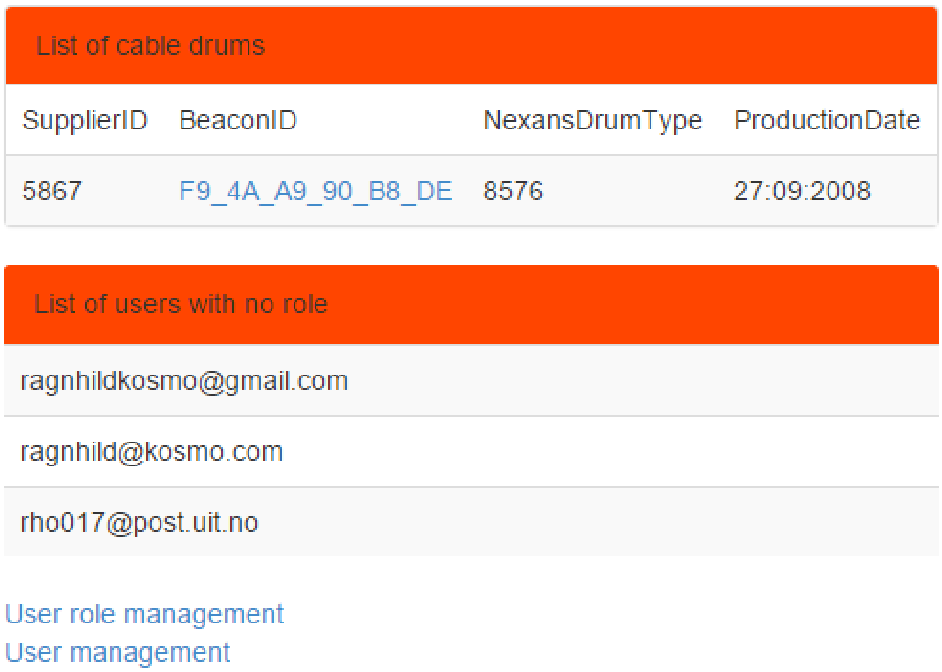

Figure 5.8: The administrator startpage

By navigating to the *User role management*, the administrator can see the different *user roles*, add and delete roles, retrieve user roles for a person or add and delete a user from a *user role*.

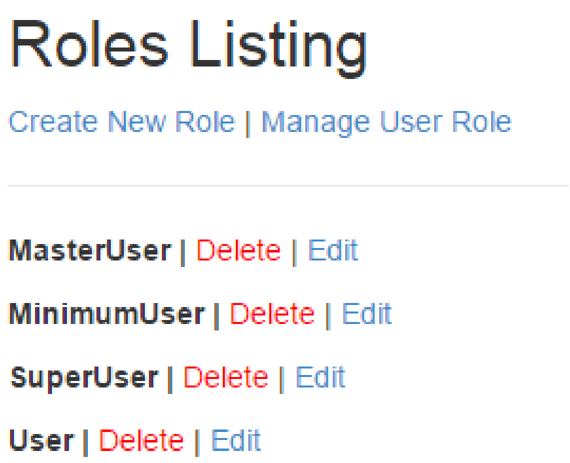

Figure 5.9: The listing of the different roles in the system

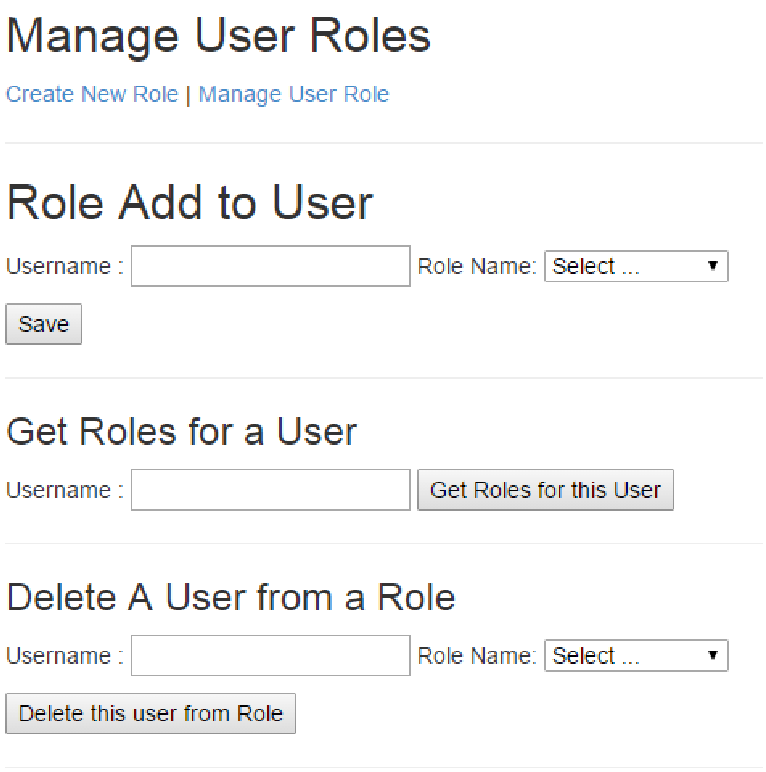

Figure 5.10: User role management

Cable Drums This section contains information about the cable drums. This will include the specifications provided by the cable drum manufacturer, the general status of the drum and maintenance history.

Each of the information fields (cable drum specifications, general status and maintenance history) requires the user to be assigned one or more *user role(s)*. The users ability of interact with the fields depends on the access level that he is given. Some roles can only inspect data, whilst others can also edit.

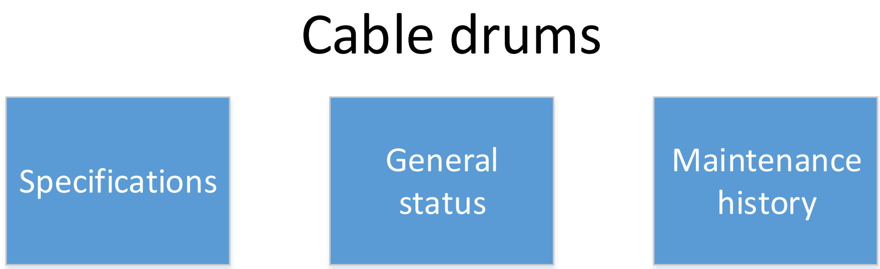

Figure 5.11: Information found within the *Cable Drums* section

Cell-based positioning The localizations of the cable drums is one of the more important parts of the system. To localize the beacons inside the plant, a grid of *reference points*, using the smart phone Huawei Ascend Y530, have been created. The technique used in this thesis is a *cell-based positioning technique*. The application *BLE Reference Point Application* is implemented to support this functionality. The factory is covered with *reference points*, where each reference will read data from nearby beacons. To create full coverage of the factory hall and to avoid deadspots, some of the circles overlap[55].

Each of the *reference points* are smart phones, which have an unique MAC-address. This address is mapped to a name, given by Nexans, in the database. The name, depending on where in the plant it is located, can for example be *laboratory*, *west emergency exit* , or machine name (for example *WaterSoak*). These names are well known for workers at Nexans.

In figure 5.2, the *reference points* are a represented with a number from 1 to 6. When a user accesses the web site, he will have information about which *reference points* that can see the beacon. If the user was looking for the red drum in figure 5.2, he will know that it is only seen by *reference point* number 2. If the user is looking for the yellow cable drum, this is seen by both *reference point* 3 and 4.

By using a grid, the possible locations of a cable drum is limited to maybe 50 square meters, instead of 18 000 square meters (total area of the factory).

How it works Within the list of cable drums on the web site, the user can search for one particular drum or drum type. The results are presented in a list, like in figure 5.12. Each of the drums contains a link which redirects the user to a page showing where the cable drums is located.

The *location* page includes the name of the cable drum, when it was last seen and which *reference points* that are nearby. In the case in figure 5.13, the cable drum is only seen by one of the *reference points*. The user can then take his Android smart phone and go to the approximate location he found on the web site. If the user starts up the *BLE User Application* the screen will be populated with a list containing the MAC-addresses of the beacons nearby as well as the RSSI measurements from these beacons. By moving towards a beacon, the signal will grow stronger, but with moving from the beacon the signal will get weaker. This way the user is able to navigate to the drum he is looking for within the grid mask of created by the *reference points*.

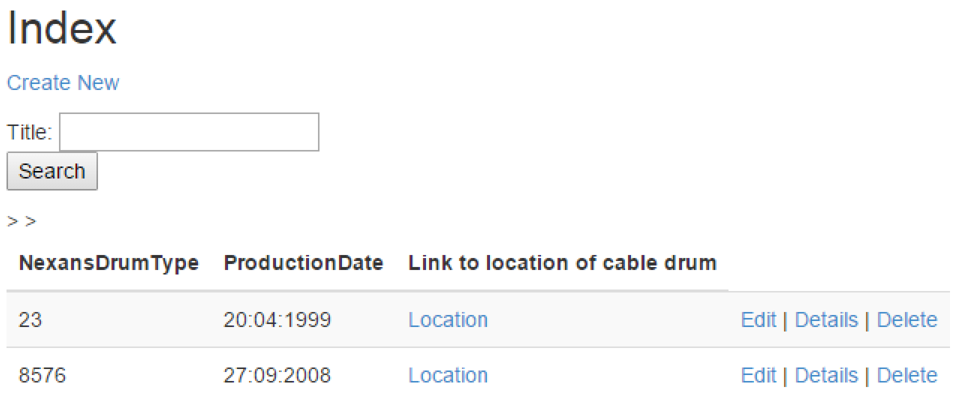

Figure 5.12: List of cable drums

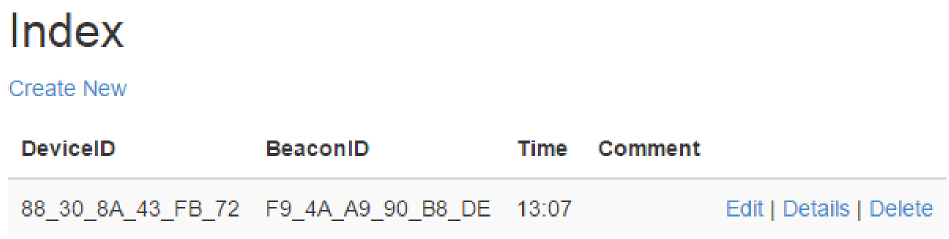

Figure 5.13: Position of cable drum

The system will only store new and up-to-date entries in the database. Old and stale entries are deleted as needed. Each time a *reference point* sends data to the middleware, the middleware will check the data up against previous entries regarding that cable drum. Depending on which reference point, time and cable drum, the database will be updated with correct information. The logic behind these updates are shown in figure 5.14, but an explanation is also presented below.

- It will check if the beacon exists in the database, if not the location is added to the database.

- If the beacon is already registered in the database, it will check whether or not it is the same *reference point* that has the old entry that sends an update, or if it is a new *reference point* that sends data.

- If it is the *old* one, the middleware will only update the time variable of the entry.

- If it is a new *reference point*, which is not registered with that beacons MAC-address before, the middleware will check if the beacon can be registered as seen by both reference points at the same time. This is an example of overlapping range circles within the figure 5.2. A beacon can be seen by two *reference points* at the same time, and this is handled in the middleware.

- If it is possible that the beacon can be seen by multiple *reference points*, a new entry with the MAC-address of the new *reference point* is added to the database.

- If the beacon can not be seen by the *old reference point* entry as well as the *new reference point entry*, the old entry is deleted and the new is added.

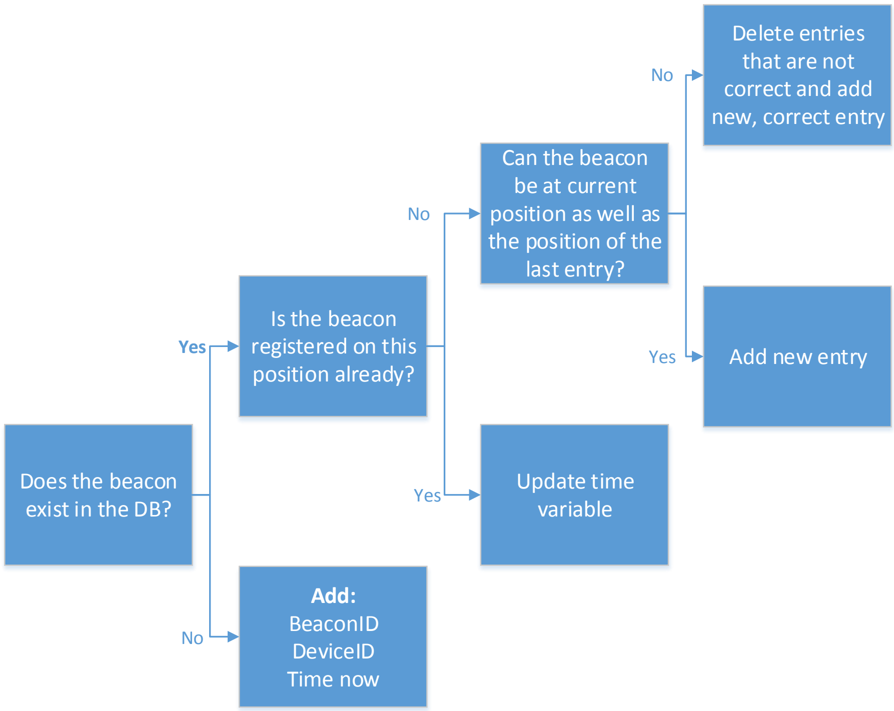

Figure 5.14: The logic behind updating the position of the cable drums

5.4 Database

The database is a an Azure SQL database. This database is made up of the schema that can been seen in appendix B. Since the Azure SQL server is built using Microsoft SQL, and is compatible with the local version of the Microsoft SQL server, is Nexans able to move their data of Azure and to a local database. By using SQL and only having the middleware communicating to the database can the SQL storage be moved to another cloud provider or other type of SQL service if needed. This might require some changes in the schema or queries based on differences in the SQL dialect.

6

Implementation

This section presents implementation details about the middleware and mobile applications created. The middleware section presents some of the frameworks used, and how these benefit the system implemented. The section *mobile applications* presents the different applications and any special methods used.

The main functionality and contribution of this thesis is designed and implemented in the middleware, so therefore it represents the majority of the chapter.

6.1 Middleware

The middleware is built upon Microsoft's ASP.NET MVC[1]. It is a open source web application framework[2]. This framework separates the application into three main components: the model, the view and the controller.

The *model* represent the core of the application, for example a list of database records. Each of the tables in the database is represented in a model, for example *CableDrums* or *Devices*. The *Controller* handles the input from the user. When an user in the system creates a new cable drum, this insert is handled in

1. Model-View-Controller
2. http://www.asp.net/mvc

the controller. It contains the logic for determining of the user has the correct access rights and if the input is in the correct format[3]. The *View* displays the data for the user.

By separating the input, business and UI logic the complexity of the application is more manageable. This is because the developer can focus of one aspect of the task at a time. It will also help the developers with parallel developing of the application. One developer can work on the controller part, while the other manages the view[4].

Controller The middleware consists of multiple controllers to separate the code for each task within the system. These controllers are shown in figure 6.1. Included in the explanation about each controller a listing of the most used URL's is presented

Figure 6.1: The different controllers created

- The *CableDrum* controller includes the functionality of adding, updating and deleting information about each cable drum at the plant. It does only keep information about the specifications, maintenance history and general status of the drum, as presented in design.

Listing 6.1: Routes within the Cable Drums controller

```
<domain>/CD
<domain>/CD/Create
<domain>/CD/Delete/DrumID
<domain>/CD/Details/DrumID
<domain>/CD/Edit/DrumID
```

- The *Devices* controller makes it possible to add new *reference points* to the cell-based positioning system. It contains variable to position the *reference point* within the grid. For example is position (1,1) and (1,2) located next to each other, while (1,1) and (10,10) might be in opposite corners within the factory.

Listing 6.2: Routes within the Devices controller

3. String, Integer, DateTime, etc.
4. https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx

```
<domain>/Devices
<domain>/Devices/Create
<domain>/Devices/Edit/DeviceID
<domain>/Devices/Delete/DeviceID
```

- The *Localization* controller functions are used by the *BLE Reference Point Scanner* to add or update the position of a beacon (cable drum). This controller collects information about the *Devices* to check whether the beacon (cable drum) can be seen by multiple *reference points* at the same time. For example can a beacon be seen by both (1,1) and (1,2), since they are neighbours, but it cannot be seen by (1,1) and (3,3) since they are too far away from each other.

Listing 6.3: Routes within the Localization controller

```
<domain>/Localizations
<domain>/Localizations/Create/BeaconID/DeviceID
<domain>/Localizations/Details/BeaconID
```

- The *Roles* controller includes functionality for adding new roles to the system. It also provides methods for adding and deleting roles to a user account.

Listing 6.4: Routes within the Roles controller

```
<domain>/Roles
<domain>/Roles/Create
<domain>/Roles/Edit?roleName=UserRole
<domain>/Roles/ManageUserRoles
```

- The *Account* controller contains methods for registering a new account and updating an existing account. It is responsible for user access, ensuring that every user is an authenticated user with the correct access rights.

Listing 6.5: Routes within the Account controller

```
<domain>/Account/Register
<domain>/Account/Login
<domain>/Manage
<domain>/Manage/ChangePassword
```

- The *Home* controller provides the user with a personalized home page. As shown in design, in figure 5.8, the home page of the administrator shows a list of users with no role, as well as all cable drums in the database.

Each controller contains a set of methods, where the main methods are shown in figure 6.2. Each of the controllers also performs checks to see if the user does have the required access rights to see the data.

- The *Index* function, for example within the cable drums controller, lists all cable drums in the database.

- *Details* lists details about one particular drum, for example diameter, core diameter and weight.

- *Create* checks input provided by the *View*, and inserts the data into the database.

- *Edit* updates the database with input from the *View*, with the proviso that the data is in the correct format[5].

- *Delete* deletes the entry from the database.

Figure 6.2: The main functionality within each controller

View The views and controllers often operate as a pair, where data shown in the view is populated by the controller. Figure 6.3 contains the different views of the system.

In addition to the views shown in figure 6.3, it also contains a *shared* view. This view contains the main design of the web page. Views in the system inherits the template created in the *shared* view. The template contains a link to *Cabledrums*, *Devices*, *Home* and *Localizations*.

5. String, DateTime, Integer, etc.

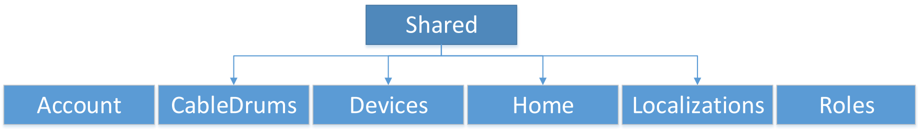

Figure 6.3: An overall figure of the *Views*

Model A model contains the data, which is retrieved by the controller and is displayed in the view. An overview of the models created in this thesis is shown in figure 6.4. More details about what which model contains is provided in Appendix B.

Figure 6.4: An overall figure of the *Models*

6.1.1 Access Rights

The middleware implemented is protected by SSL and the authorize attribute. The *Authorize attribute*[6] specifies that access to a controller or action is restricted only to those who meet the *User Role* requirement. Users registered in the system does contain an attribute specifying the granted access level. This attribute is determined by the work tasks at Nexans. Only the administrator of the system is able to add new users and then the access level attribute is set. This can be altered if the tasks of the user changes.

For implementing different user levels in the system *ASP.NET Identity* is used[7]. The Identity framework can be used with all of the ASP.NET frameworks, such as ASP.NET MVC[56]. The system stores the user information in a database, and it uses *Entity Framework Code First* to implement the persistence mechanism.

In this implementation anonymous users are allowed to look at the home page of the site. This is done by adding the attribute *Allow Anonymous*[8] to the given methods.

6. https://msdn.microsoft.com/en-us/library/system.web.mvc.authorizeattribute.aspx
7. http://www.asp.net/identity
8. http://blogs.msdn.com/b/rickandy/archive/2012/03/23/securing-your-asp-net-mvc-4-app-and-the-new-allowanonymous-attribute.aspx

6.1.2 Routing

MVC contains methods to create routes. These routes goes to different parts of the web site, and the developer can make routes that connects the different elements and makes it possible to redirect a request to the correct view/controller. In this thesis, the routes have been used to connect the mobile applications created and the middleware.

The first route created, in listing 6.6, creates an route used for sending the MAC-address of a *reference point* (DID) and the MAC-address of the beacon found (BID).

Listing 6.6: Route for passing information from "BLE Reference point application" to the middleware

```
routes.MapRoute(
    "Insert",
    "{controller}/{action}/{BID}/{DID}",
    new {
        controller = "Localizations",
        action = "Create",
        BID = "",
        DID = ""}
);
```

The second route defined is used in the NFC application. When the user reads a NFC tag placed on the drum, he will be redirected to the correct web site. This is done by a HTTP request containing the BeaconID (BID) written on the NFC tag.

Listing 6.7: Route for getting the web site containing information about the cable drum

```
routes.MapRoute(
    "Index",
    "{controller}/{action}/{BID}",
    new {
        controller = "CD",
        action = "Details",
        BID = "" }
);
```

6.2 Mobile applications

The *BLE Reference Point Scanner* did have most of its functionality implemented in the pre-project, but some alterations have been done to it. The *BLE User Scanner* is built upon the *BLE Reference Point Scanner*, but it is not connected to the backend. Additional functionality of including measurements of the RSSI from each beacon have been added.

The NFC application in the pre-project was discarded because the functionality did not suit the design of the system. A new application have been implemented in this thesis.

6.2.1 BLE Applications

The two BLE applications are built upon the same base, only with minor differences. They both contain a list, which is populated with the MAC-addresses of the nearby beacons. To enable the built-in platform support for BLE Android 4.3 (API Level 18) is used.

BLE Reference Point Scanner The Stationary application is built up as a loop, which runs every two minutes. Within a this loop, the application scans the area for beacons, populates the list, and sends the data (MAC-addresses) found to the middleware, which handles the input.

The data is sent as a HTTP request, which includes the MAC-address of the *reference point* as well as the MAC-address of the beacons nearby. The URL used to send data from application to middleware is shown in listing 6.6. It will send one request to the backend for each MAC-address populated in the list shown on the display.

BLE User Scanner The user scanner does not have a connection with the backend, and therefore it does not need to set up a connection. It does not require a internet connection.

The *BLE User Scanner* have a continuous measurement of the RSSI of each beacon that the smart device sees. To store the RSSI measurements from each device a *HashMap* is used. It contains the *BluetoothDevice* and an *Integer* representing the RSSI-measurements.

6.2.2 NFC Drum Information

This application is started at command from the user. When the user touches the NFC tag stuck on the cable drum an activity is started and the tag is read. The tag contains the *SuppliersID* provided by Nexans. This *SuppliersID* is unique for each cable drum at the plant. After a tag is read the user is redirected to the web page containing information about that drum. The URL the user is redirected to is shown in 6.7.

7 Evaluation and challenges

This chapter will present challenges that are relevant for this solution as well as an evaluation of the technology and methods used.

7.1 Testing

Due to the short time span and limited resources only a limited part of the factory hall have been used for testing. Only a few reference points (smart phones) have been used for testing the system created. This is sufficient for testing how cell-based positioning works, the range of the beacons, the battery life of the smart devices used by the users and the logic implemented within the middleware.

7.1.1 Locating the cable drums within the factory

The challenges presented in this section are time consuming tasks and have therefore not been prioritized to perform full tests on. The tests performed in this thesis have been enough to predict the behaviour of the system implemented and technology chosen. Some of the tests were started in previous projects, but have been continued in this thesis, and therefore they are presented in this thesis also.

Range

The factory hall at Nexans spans 18 000 square meters. It also have numerous storage tents outside the plant. Inside the factory hall, Nexans have different sections, machines and equipment for producing and testing cables. In the storage tents they mostly store empty cable drums, or cable drums containing some unfinished product or raw material.

One known challenge with BLE, is the possible interference of signal by other machines and objects. As presented before, the advertised range of BLE beacons are 50 meters, but the signal interferences from other objects causes limitations regarding the range. The extent of this possible issue have been tested on numerous occasions on different locations of the plant, with different cable drums. It shows that the expected range of the beacon inside the factory hall is about half of the advertised beacon range. Approximate measurements have shown that the beacons have a range of approximately 20 meters in the most difficult environments, while some areas provide a range of 30-40 meters.

These challenges have to be addressed in future works. It demands a significant *offline* or *learning* phase to populate the grid within the plant with regards to the known objects that cause signal disturbances. Thorough testing have to be performed at all locations of the plant to estimate suitable locations to place the *reference points* as well as to register which of the *reference point's* range circles that overlap.

Accuracy

The *BLE user application* shows the RSSI received from each of the beacons. The closer to zero the RSSI measures are, the closer the user is to the beacons. When using only one type of beacon all beacons will use the same broadcasting power. If Nexans use different beacons, the signal strength of each beacon emitted can be different. When two beacons send signals with different strengths, but are located at the same spot, the users scanner will detect the beacon which sends out with highest signal strength as closer than the other beacon[44].

At Nexans, only the *StickNFind Pro* beacons have been tested thoroughly, and they are all set to emit with the same broadcasting power. For the future, if Nexans changes the beacon type, this issue might have to be addressed again.

Testing at Nexans have shown that the accuracy of these are RSSI measurements are low, and that the measures RSSI cannot be directly linked to a distance in meters. The *BLE User Scanner* can only be used to give the user an indication of which direction the cable drum is located. The user will know if he is getting further away or closer to the cable drum.

The reason why the accuracy of these RSSI measurements are so low, is the amount of disturbance/interference from other objects in the factory hall. There are large machines and walls located around the plant. These stationary objects could have been plotted on the grid map, and the signal disturbance from these objects could have been calculated in the *offline phase* of the *cell-based position* technique used in this thesis. The challenge at Nexans is the many moving objects that cause signal disturbance. These objects are often trucks and other large steel cable drums. These moving objects cannot be plotted in the *offline phase* of the *cell-based position* since they are often moved.

7.1.2 BLE beacons

The main technology to test out in this thesis, is how the BLE beacons behave inside the Nexans plant. This section will present different challenges that can or have occurred.

Battery life

The advertised battery life of the beacons are 2-5 years. The testing performed this far can neither confirm nor deny this. The beacons placed on the cable drums in May 2014 are still operative in May 2015. This includes both Estimote beacons as well as StickNFind Pro beacons.

What can affect battery life significantly is temperature changes. If a battery is exposed to cold weather the battery life may be compromised. The storage tents are not warmed in the winter, nor are they air-conditioned in the summer. This creates temperature swings that can affect the beacons battery life. This have not been tested in this thesis, but is left for future work.

The *StickNFind SDK* provided a variable for *battery measurements*, and these have been measured and registered. These measurements fluctuated, and did not make any sense because they were neither measured in hours, days, years or percentages. After multiple emails to the *StickNFind* company, they responded with that these measurements are incorrect and does not give any indication of the expected battery life left of the beacons. As of now there are no methods of estimating the remaining battery power of the beacons.

Robustness

The beacons used at Nexans would have to endure a rather harsh environment, compared to other environments like hospitals or warehouses. One possible challenge was if the beacons were broken or damaged easily. It was also a possibility that the larger *Estimote* beacons were more prone to damages than the smaller *StickNFind Pro*. During the last year of testing, this have not caused any problems. Both the *StickNFind Pro* and *Estimote* beacons have endured all challenges, from water soak, voltage tests and temperature changes.

These tests have shown that the size of the beacons have not had such an impact as first assumed. This might be due to the placement of the beacons on the cable drum. The beacons are located inside the grooves of the drum, and therefore somewhat protected from damage. For future works the *StickNFind Pro* would nevertheless be less prone to damage than *Estimote*.

7.1.3 Position history

In this thesis position history have not been stored to limit the data in the database. An interesting feature could have been to store the information about the past locations of the cable drum. This can be used to track the activity of each drum, and to analyse how much each drum is in use. It would also make it simpler to find faults in the cell-based positioning. Since the range of the beacons fluctuates between 20 and 40 meters, depending on interference, inside the hall the number of *reference points* in the grid that can see the beacons will also fluctuate. The position history can be used to test the range of the *reference points*, and to see which *reference points* that has overlapping range areas.

The last position of each cable drum is always stored in the database, and is never timed out. If the location entry did time out, and got deleted after a while, the workers would not have any way of knowing where it was last seen, or why the cable drum is registered with a location anymore. If the last position is stored in the database, it is possible for the workers to find the drum again. Reasons why the drum have not been seen lately is because it are being performed tests on in the *water soak*[1], it have been moved, sold to a customer or the beacon might have run out of battery.

1. The water soak tests are where they soak the cable drum and cable in a pool of water. Then AC/DC testing is performed on the cable

7.1.4 Machine Learning

As the system is designed and implemented, it consists of a significant *offline/learning* phase where the range and interference on different locations within the factory is measured. This *offline* phase will estimate which *reference point* ranges that overlap. Where two *reference point* ranges overlap, the beacon (cable drum) can be seen by both. Since the range within the plant is not constant at for example 20 meters, maintaining which ranges overlap is very time consuming and nearly impossible to keep correct at all times. The reason why the range fluctuate is the difference in amount of objects blocking the signal. Where a cable drum is located in a cluster, as seen in figure 2.2, the signal is blocked and does not have the same range as if it was placed in an open space, such as in figure 2.3.

To limit the *manual* learning phase in this system, *machine learning* could have been implemented. Nexans does have a total of around 20-30 different machines at the plant, and up to 2000 cable drums. This implies that most of the cable drums is not in use and not moving; they are stored somewhere for future use. Some cable drums are stored for days or weeks, and therefore the need for continuous updates are not needed.

This property can be can be taken advantage of with implementing *machine learning*. If conditions at Nexans change, and the range of a beacon (cable drum) is extended to reach 40 meters, instead of 20 meters, it will be seen by more *reference points*. Figure 7.1 shows an example what could happen if the range of a beacon change without updating the backend, and which *reference points* that see the same beacon. If the range within an area was shown to be the *old range* of figure 7.1, the beacon (cable drum) would only be seen by *reference point* 1 and 2.

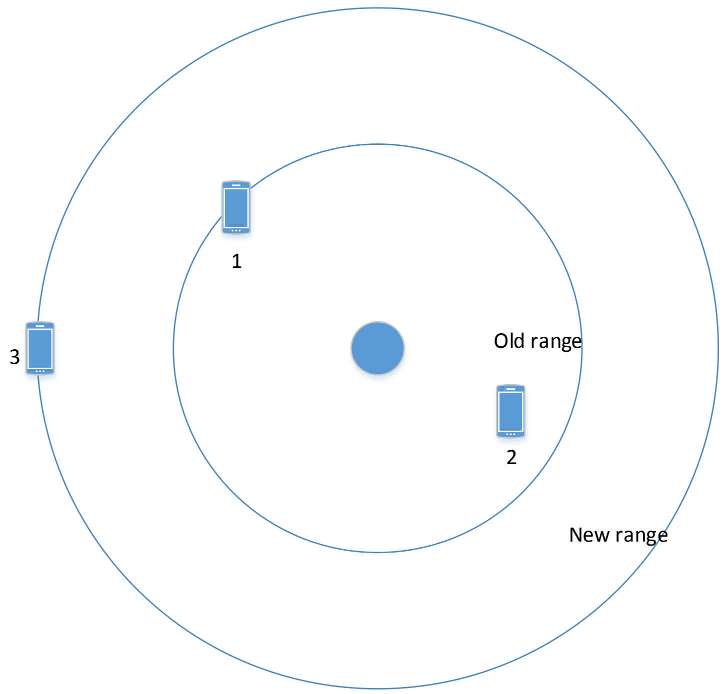

Figure 7.1: Figure showing what happens if the range in an area changes

If the range were to change, because conditions at Nexans change, the beacon (cable drum) would now be seen by both *reference point* 1, 2 and 3. The backend does not at the current moment handle changes in range, and this would have to be updated manually.

To make this happen *automatically* machine learning can be implemented. If the range of a beacon (cable drum) changes, the backend will *jump* back and forth between two (or more) *reference points*, and this will include adding new entries and deleting old entries. If this happens over a given period of time, for example 12 or 24 hours, the backend should *learn* from this. The backend will learn that the beacon in figure 7.1 can now be seen by all three *reference point*, and not just by 1 and 2. This *automatic* learning will limit the need for manual updates of the ranges at different locations at the plant.

7.1.5 Battery life of smart devices

The battery life of the *reference points* at the plant are not necessary to test because they are always connected to power.

The smart devices used by the workers at the plant will notice a drain in battery life when having their Bluetooth feature turned on. Minor tests have been performed testing the battery life of a Samsung Galaxy S3 in an environment which has multiple BLE beacons. Due to the limited testing equipment and use of an older Android device, the test results are not viable.

Because of this, some tests from *Aislelabs*[2] are used as a reference for this section. This section was discussed in the pre-project as well, but it is still just as relevant in this thesis.

The battery life of a smart phone will depend much on the amount of surrounding BLE beacons broadcasting signals. When there are no BLE beacons around when Bluetooth is turned on, the battery usage on the smart device is much lower. At Nexans there will be a considerable amount of BLE devices constantly emitting a signal, and therefore we can assume that the battery drain on smart devices will be significant.

Since we only use Android devices in this thesis, we will only focus on these. Table 7.2 shows the battery drain for three different Android devices. The Bluetooth radio is ON for the same amount of time on all devices[3]. The *Baseline* is where there are no BLE beacons around.

	Baseline	0 beacons	7 beacons	10 beacons
Nexus 4	-1.05%	-2.14%	-2.63%	N/A
Nexus 5	-0.57%	-1.43%	-2.23%	-5.18%
Moto G	-0.76%	-1.20%	-1.44%	-1.79%
Source: aislelabs.com				Aislelabs

Figure 7.2: A figure shoring the battery drain on different Android devices

The reason why the drain is so different from device to device, is the difference in efficiency of the chip within the phone. When more BLE beacons are emitting signals, the phone will have to process these signals. Newer smart devices does this processing more efficient, and therefore they consume less power. The Nexus 4 is one of the older smart devices supporting BLE, and it

2. http://www.aislelabs.com/
3. http://www.aislelabs.com/reports/ibeacon-battery-phones/

contains a more inefficient chip. On the Nexus 4 there was a bug in the BLE implementation done, and therefore Aislelabs could not get any data on 10 beacons and more.

From the table showing battery drain on the different phones, it does show a significant decrease in battery life when the area contains multiple BLE beacons. Workers at the Nexans plant would notice a difference in battery life when using their mobile at work, granted that they have Bluetooth toggled on. During an 8 hour shift they would notice a decrease in battery life of up to 40 percent more than if they have Bluetooth off.

7.1.6 User testing

The application have on several occasions been tested at Nexans Rognan. These tests have been performed both with and without the workers and managers at Nexans. The collaboration with Nexans Rognan have worked well throughout the duration of this thesis. Challenges could have arisen when working with a real customer, for example if they have difficulties to acquire hardware for testing. This have not caused any problems in this thesis. They have always been able to answer any questions, they have ordered beacons and smart phones and they have been available for testing the system created.

The tests have been performed with workers from different parts of production. They have the application installed on their smart phones, and have tested the web interface. They have proposed different functionalities to be added to the web site, and these have been taken into account and added to the web site.

Some challenges during this testing have occurred. Late arriving mobile devices limited the amount of testing done on the *cell-based positioning system*. The functionality of this system does, as far as testing have shown, work as expected, but it lacks testing in case of any special cases that have not been addressed yet.

The second challenge is the varying internet connection within the factory hall. The *BLE Reference Point Scanner* requires an internet connection for sending data to the backend. Some areas of the plant have a limited connection, and will cause an issue for the system. For future work the IT personnel have to install multiple wireless routers around the factory and storage tents so that this will not be a problem.

7.2 Challenges

The previous section presented challenges with the technologies used. This section will present any challenges that are important to consider when implementing a system like this.

7.2.1 Security

In any system there are some security risks. This system will store information about the production at Nexans. By gaining access to the information stored in the database, the person will have access to know what materials are stored on each cable drum. There are many materials that can be worth a lot of money, for example copper. But also finished products can be worth millions of Norwegian Kroner. By knowing the exact location of a drum, for example *storage tent 1*, the outsider can easily find the most valuable drum and steal it. Not all locations are as thoroughly supervised as others, and by knowing where an exact drum is located, thieves can be quick at collecting the particular drum.

By sending the data over the internet to Azure, it can be picked up my outsiders and used for malignant purposes. Web browsers often rely on SSL/TLS to prevent attackers from obtaining the data, but these security measures was bypassed in 2009. If an attacker is sniffing any packages on the host, he can pick up any credentials that pass through. The attacker can then use these credentials to log on to the site. When the SSL/TLS implementation was breached in 2009, the users were now able to perform requests to the cloud systems without these credentials [57]. This is one of the more likely breaches that Nexans may face, and a reason for considering having the data stored locally.

7.2.2 Extensibility

The middleware is, as presented before, built upon the principles in ASP.NET MVC, with controllers, views and models. Each of the features like location of cable drums specifications, cable drums and device logic is encapsulated into different controllers to provide an extensible system as well as to decrease the dependabilities between the parts. Each of the parts can easily be exchanged or altered without affecting the other functionality significantly.

The modules created are easily exchangeable if specifications and requirements at Nexans change.

7.2.3 Cloud computing

The three cloud providers presented here all advertises with a:

> Flexible, cost-effective, secure, scalable and elastic and experienced cloud service.

By having the system locally, they would have to invest in hardware and software, and resources would have to be set aside to get the system up and running. By choosing a cloud provider, Nexans will not have any up-front costs in form of new hardware and software. The costs of the system would come as a result of how much of the resources at Azure they use. In the long run, using a cloud provider might provide some extra expenses, but there are several advantages with outsourcing this job.

As presented, there are no up-front costs with taking an application up in the cloud. This means that they do not have to estimate the potential resources needed when investing in hardware and software to be used locally. By using Azure, the system is flexible, and Nexans would only pay for what they use.

By having the system in the cloud, Nexans does not need to worry about making the system fault tolerant regarding hardware or software failure, because this is handled in Azure. If a node falls down, or the data becomes corrupt, this will be handled by Azure, having multiple nodes storing the same data.

Scalability

One challenge that have not been tested in this thesis is the scalability of the system. How will the system handle up to 2000 beacons within a relatively small area? Nexans have not yet decided how many of the drums on the plant that they should include in the system, but it will atleast be around 500 beacons. Many of the drums at the plant are of the same kind and they are placed at one location most of time. These cable drums often only store raw materials used in production, and they are most likely not needed to be tracked in this system.

Since this thesis intends to run in the cloud, Azure will handle any necessary scale ups regarding computational power. To limit some of the computations done in Azure, the application on the mobile devices can do more of the work. As they are now they only collect and send data, and the middleware, which runs in Azure, does all the tests on the data received. The *BLE Reference Point Scanner* can for example check the new list of beacons towards the old list and send the changes to the middleware. The changes will be if a new beacon have entered or left the range of the *reference point*. There is limited data stored in the Azure database, and the data stored is only text, so limited storage capacity is needed in the cloud database.

Who owns the data in the Cloud?

More and more people, both individuals and businesses, are putting their data in the cloud. Information, ranging from emails, pictures, financial information, information about cable drums and peoples whereabouts are stored in the cloud. Who owns this data, and who has access to it?

Data in the cloud is spread out over multiple datacentres that are localized in multiple countries. Not necessarily in the country where the user is located.

Microsoft, Apple, Verizon, AT&T and Cisc been contacted by the US government, where the US government have requested to collect data from their storage locations outside of the US. Microsoft and its affiliates on this matter expresses that this would harm the international business as well as they do not want to compromise one country's laws for another Microsoft and the other companies have tried to prevent the government from collecting this data. They lost the first round in court, with the reason that they would hinder the law enforcement in gathering information that could lead to capturing of criminals[4].

There have been multiple questions regarding use of cloud computing in Norway. In 2011 and 2012 Narvik and Moss municipality contacted *Datatilsynet* in Norway regarding their use of Google and Microsoft Office 365. *Datatilsynet* presented some prerequisites for when deciding to use a cloud. The business should analyse what the impacts of using the cloud is, and what could possible go wrong with storing the data in the cloud? The data processor agreement should be in compliance with Norwegian law[5]. These questions should be discussed not only in public sector, but in private businesses too.

Meetings with Nexans have been done regarding any issue they might have with having the data in the cloud. The main conclusion was that as long as the system is not connected to the production system SAP, it does not contain any sensitive information that can not be presented for the public. The worst

4. http://www.cbc.ca/news/business/microsoft-and-other-tech-giants-fight-u-s-right-to-seize-cloud-data-1.2677688
5. https://www.datatilsynet.no/Teknologi/Skytjenester—Cloud-Computing/Risikovurdering-og-informasjonssikkerhet/?showContentList=true&showDetailedContentList=false&readMode=fa

case scenario would be an leakage of the database content, this would provide external parties information about the status and location of every cable drum at Nexans. Nexans has considered this risk, and decided that since this is not connected to the SAP system, and therefore does not contain any sensitive information that would impact their ability to compete, would a leakage not cause too much harm. For future works, when the system will be included in SAP, it needs to be taken down from the cloud and ported to a local server.

8

Conclusion and future works

This chapter consists of two parts. First, future work is presented. Since this is a prototype of a system there are features that can be enhanced, new features can be added for a more seamless system and the system can be integrated with the production system SAP. The second part summarizes the thesis and a conclusion is drawn from the results of testing, how the different modules of the system works together and meetings with Nexans, both production workers and production managers.

8.1 Future works

This section presents different new functionalities that can be added and parts of the system that have potential to be improved on.

8.1.1 Interoperability

The system implemented in this thesis is a standalone system, not integrated with the existing system SAP. Future work will include creating an extension in SAP for the tracking system. It will make it possible to go through SAP, where all the orders for cable drums are placed, and to this system which only handles organization about the cable drums.

8.1.2 Beacon testing

Beacon testing have been performed from May 2014 and until May 2015. This testing have included multiple features regarding the beacons, both range, if the beacon is water proof and how the beacons handle the harsh environment they are exposed to.

Testing are still being performed, and more beacons are now placed on cable drums around Nexans. This testing is regarding battery life, how the beacon responds to temperature changes, range and robustness.

8.1.3 Mobile applications

The *BLE User Scanner* is not complete. The user will have to recognize the MAC-address from the web page when searching for a beacon. The application will not collect the drum id from the database. For future works this should be implemented. With performing additional requests to the database, the application might be less responsive, but it will improve the user experience significantly.

Users have also started using the applications at Nexans. The users have and will continue to provide feedback regarding the functionality and user-friendliness of the application.

8.1.4 Installation at plant

Only a minor area have been used for testing. Future work will install more *reference points* (smart phones) around the plant to locate beacons. This installation will also include a longer *offline/learning* phase for testing range of the beacons at different locations around the factory area, both inside and outside. The range of the beacons in different locations will influence the frequency of *reference points* and how the calculations with regards to the grid that each reference point has.

Map of plant

As of now, the system only gives information on which *reference points* are near the cable drum the user is looking for. This implies that the user have to know where all the *reference points* are located. Nexans have expressed a desire for an interactive map which shows the whole plant, the location of the *reference points*, and the cable drums plotted on the map. An example of how this map would look is presented in figure 5.2.

By including this map it will be simpler for the users to navigate to the correct drum without knowing the name of each *reference point* in the plant.

8.1.5 Expanding the system

The system implemented in this thesis is a system meant for use at Nexans Rognan. Future work may include expanding the system to work at other locations, for example Nexans Halden, Karmøy or Namsos. It is also possible to create a solution which, with few alterations, can be used in many other industries. For example tracking cars or real-time tracking of inventory in a department store.

Nexans Rognan has the most potential to improve their efficiency. The other factories in Norway does mostly produce standard cables, while Nexans Rognan produces specially made cables. This means that the other factories does not need such a wide range of different cable drums as Nexans Rognan. The other factories stock many drums of each type, and therefore have several drums available of each type and they are more often stacked in clusters including the same type of drum. Therefore the need for tracking each individual drum is not needed in the same extent as Nexans Rognan. There can be exceptions where they order a few cable drums of one type, and in these cases it might be a reason to adopt the tracking system from Rognan.

8.1.6 Accuracy of location

For future works a more accurate positioning can be implemented. The beacons are now set to emit power to support an range of approximate 20 meters. Without disturbances from objects and walls, these beacons would have had a range of approximately 50 meters. For a more accurate solution, for example 10 meters, the broadcasting power can be set lower to achieve this. This also have the positive side effect of better expected battery life.

In this thesis only a smaller area at Nexans have been used for testing, and it seems that a 20 meter range does work well. Testing performed by several workers at Nexans have resulted in an overall consensus that a 20 meter range works well for their use. When the system is implemented throughout the plant, this might be reconsidered, and a shorter range might be chosen. A longer range would impact the battery life of the beacons badly, and they would have the problem of beacons running out of battery too often.

8.2 Conclusion

The prototype of *Nexans Tracking System* implemented in this thesis does perform as expected. The system implements the specifications provided by Nexans Rognan, and it solves the goals presented in the project statement.

8.2.1 Beacons

Testing performed in this thesis have shown that, so far, the beacons used for this purpose does works in the challenging environment at Nexans Rognan. Because of the size of the StickNFind (SNF) Pro vs. the Estimote beacons, the SNF Pro did show the most promise. It was expected that the Estimote beacon would get damaged and broken because of the size of it. This have not yet happened, and both the SNF beacons and Estimote beacons are still running. Neither are broken or have run out of battery yet. The only difference between the two beacons is the price per beacon. The Estimote company sells only three beacons in a package, and this costs 99 dollars for three beacons. This is a price of 33 dollars per beacon[1]. The StickNFind providers does sell beacons in packages of 10, 20 and 30 beacons, with the 10-pack costing 25 dollars per beacon, the 20-pack costing 24.5 dollars per beacon and the 30-pack costing 24 dollars per beacon[2]. So price-wise, the SNF Pro is considerable cheaper than the Estimote beacons[3].

8.2.2 Finished product

This section will present how the system implemented compares to the project goals presented in chapter chapter 1. It will include what the customer, Nexans Rognan, thinks about the system implemented, if they find it promising and if

1. http://estimote.com/#jump-to-products
2. https://www.sticknfind.com/store.aspx
3. A query have not been sent to neither Estimote nor SNF to get an offer for if Nexans were to buy 1500 beacons

it is worthwhile pursuing further.

Testing done in collaboration with the workers at Nexans have shown that the system is intuitive to use and provides enough functionality for the users to use the system as it is now. The users are able to add, update and delete information about the drums. The tickets, as shown in figure 2.1, and the excel sheet used to add remarks about the drum is now redundant. The users can easily find information about the drum, using the NFC application, and they can also update the cable drum information when the NFC application redirects them to the page containing data about the particular drum. The cell-based positioning technique, using Android devices and BLE beacons, provides a good overview over the location of each cable drum inside the plant.

If Nexans were to fully adopt this system, and continue to implement the features presented in *future work*, they have estimated that they can save 2-3 FTE's each year because the workers no longer need to search for cable drums. Also, if they at all time have enough cable drums available at all time, and they are all in good state, they can have lower downtime on the machines they. Each machine costs a significant amount to buy and install, and if they can have less downtime, the cost/hour per machine will decrease. This also means that Nexans Rognan can meet more deadlines, or they can offer shorter deadlines to their customers.

Appendices

A

Layout of factory hall

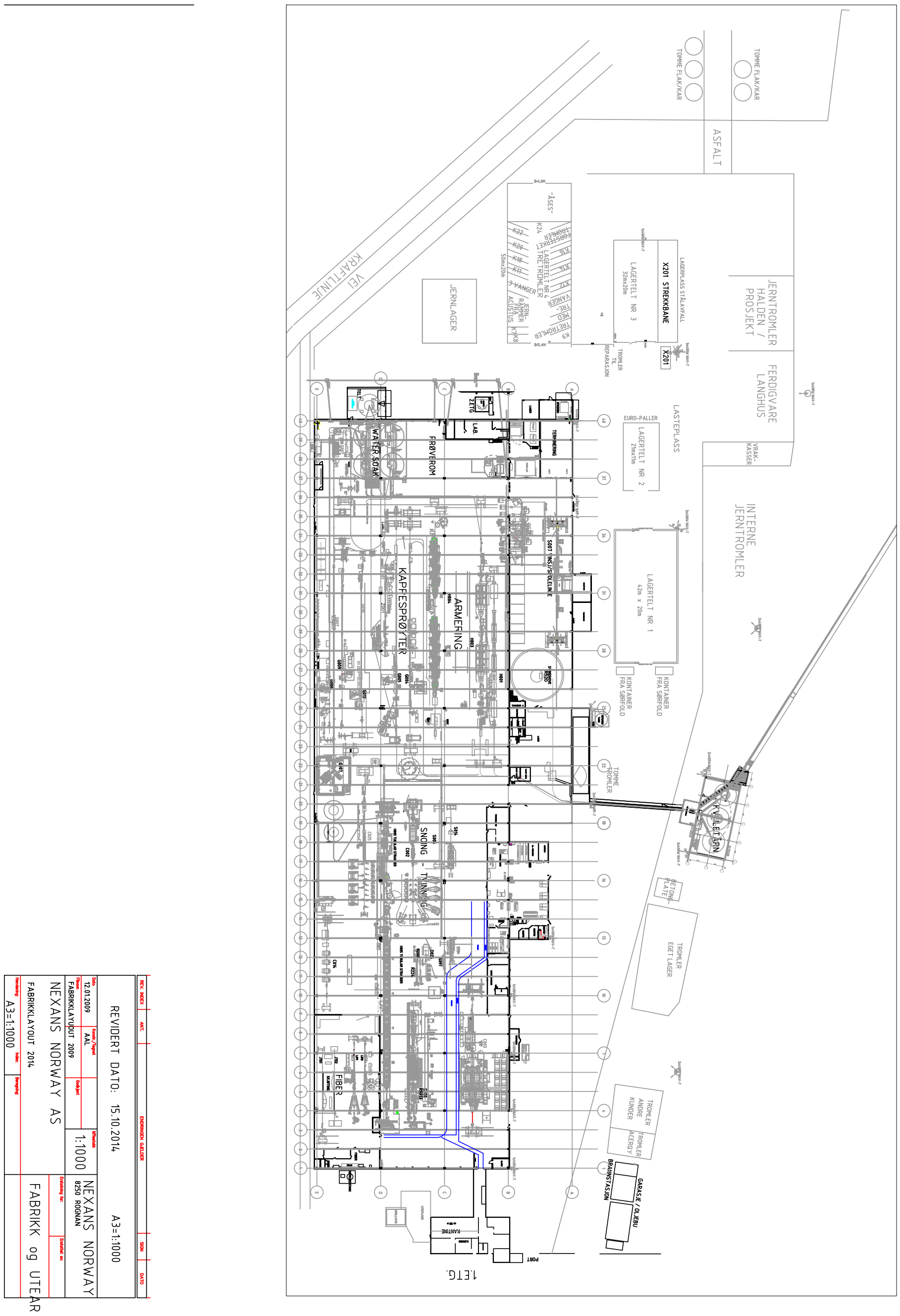

Figure A.1: Figure showing the layout within the factory. The figure is unclear unless it is shown in a format larger than A3

B

Database

Listing B.1: Database Schema for the table "Users"

```
CREATE TABLE [dbo].[AspNetUsers] (
    [Id]                    NVARCHAR (128) NOT NULL,
    [Email]                 NVARCHAR (256) NULL,
    [EmailConfirmed]        BIT            NOT NULL,
    [PasswordHash]          NVARCHAR (MAX) NULL,
    [SecurityStamp]         NVARCHAR (MAX) NULL,
    [PhoneNumber]           NVARCHAR (MAX) NULL,
    [PhoneNumberConfirmed]  BIT            NOT NULL,
    [TwoFactorEnabled]      BIT            NOT NULL,
    [LockoutEndDateUtc]     DATETIME       NULL,
    [LockoutEnabled]        BIT            NOT NULL,
    [AccessFailedCount]     INT            NOT NULL,
    [UserName]              NVARCHAR (256) NOT NULL,
    [Firstname]             NVARCHAR (MAX) NULL,
    [Lastname]              NVARCHAR (MAX) NULL,
    CONSTRAINT [PK_dbo.AspNetUsers] PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

Listing B.2: Database Schema for table "User Roles"

```
CREATE TABLE [dbo].[AspNetRoles] (
    [Id]   NVARCHAR (128) NOT NULL,
    [Name] NVARCHAR (256) NOT NULL,
    CONSTRAINT [PK_dbo.AspNetRoles] PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

Listing B.3: Database Schema for table "CableDrums"

```
CREATE TABLE [dbo].[CableDrums] (
    [SuppliersID]         NVARCHAR (128) NOT NULL,
    [NexansDrumType]      NVARCHAR (MAX) NULL,
    [NetWeightEmptyDrum]  NVARCHAR (MAX) NULL,
    [MaxLoad]             NVARCHAR (MAX) NULL,
    [ProductionDate]      NVARCHAR (MAX) NULL,
    [FlangeDiameter]      INT            NOT NULL,
    [OverallWidth]        INT            NOT NULL,
    [InsideWidth]         INT            NOT NULL,
    [CoreDiameter]        INT            NOT NULL,
    [BoreDiameter]        REAL           NOT NULL,
    [CableEntry]          NVARCHAR (MAX) NULL,
    [DrivePinSlit]        NVARCHAR (MAX) NULL,
    [SpindleSpeed]        INT            NOT NULL,
    [TurnoverSpeed]       INT            NOT NULL,
    [TopCoatColor]        INT            NOT NULL,
    [CableWireDimension]  NVARCHAR (MAX) NULL,
    [MinCableTension]     INT            NOT NULL,
    [MinPayloadFullDrum]  INT            NOT NULL,
    [LastMaintenance]     DATETIME       NOT NULL,
    [LastMaintenanceWho]  NVARCHAR (MAX) NULL,
    [LastMaintenanceWhat] NVARCHAR (MAX) NULL,
    [NextMaintenance]     DATETIME       NOT NULL,
    [GeneralStatus]       INT            NOT NULL,
    [CleanlinessGrade]    INT            NOT NULL,
    [RustGrade]           INT            NOT NULL,
    [Color]               NVARCHAR (MAX) NULL,
    [DateOfRemark]        DATETIME       NOT NULL,
    [Comment]             NVARCHAR (MAX) NULL,
    [BeaconID]            NVARCHAR (MAX) NULL,
    CONSTRAINT [PK_dbo.CableDrums] PRIMARY KEY CLUSTERED ([SuppliersID]
);
```

Listing B.4: Database Schema for the table "Localizations"

```sql
CREATE TABLE [dbo].[Localizations] (
    [Id]        INT            IDENTITY (1, 1) NOT NULL,
    [DeviceID]  NVARCHAR (MAX) NULL,
    [BeaconID]  NVARCHAR (MAX) NULL,
    [Time]      DATETIME       NOT NULL,
    [Comment]   NVARCHAR (MAX) NULL,
    CONSTRAINT [PK_dbo.Localizations] PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

Listing B.5: Database Schema for the table "Devices"

```sql
CREATE TABLE [dbo].[Devices] (
    [DeviceID]  NVARCHAR (128) NOT NULL,
    [GridX]     INT            NOT NULL,
    [GridY]     INT            NOT NULL,
    [Comment]   NVARCHAR (MAX) NULL,
    CONSTRAINT [PK_dbo.Devices] PRIMARY KEY CLUSTERED ([DeviceID] ASC)
);
```

Bibliography

[1] D. Evans, "The internet of things, how the next evolution of the internet is changing everything," 2011. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the internet of things," *Internet Computing, IEEE*, vol. 14, no. 1, pp. 44–51, 2010.

[4] M. Chui, M. Löffler, and R. Roberts, "The internet of things," *McKinsey Quarterly*, vol. 2, no. 2010, pp. 1–9, 2010.

[5] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[7] A. A. T. Hardersen and N. Schirmer, "Privacy for cloud storage," in *ISSE 2014 Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe 2014 Conference*. Springer, 2014, p. 211.

[8] S. Vaudenay, "Security flaws induced by cbc padding—applications to ssl, ipsec, wtls..." in *Advances in Cryptology—EUROCRYPT 2002*. Springer, 2002, pp. 534–545.

[9] apprenda, "Iaas, paas, saas (explained and compared)," 2015. [Online]. Available: http://apprenda.com/library/paas/iaas-paas-saas-explained-compared/

[10] G. Lawton, "Developing software online with platform-as-a-service tech-

nology," *Computer*, vol. 41, no. 6, pp. 13–15, 2008.

[11] B. Stineman, "Why windows azure?" February 2011. [Online]. Available: https://brentdacodemonkey.wordpress.com/2011/02/18/why-windows-azure/

[12] Google, "Google cloud platform," 2015. [Online]. Available: https://cloud.google.com/appengine/?gclid=CjoKEQjw_pmoBRDu986bpISz5ZsBEiQANiuHDDDMO8meZpwnoLNie3V2vLekq75IcDeTR-sQqbmlsPoaAvHp8P8HAQ

[13] R. Prodan and S. Ostermann, "A survey and taxonomy of infrastructure as a service and web hosting cloud providers," in *Grid Computing, 2009 10th IEEE/ACM International Conference on*. IEEE, 2009, pp. 17–25.

[14] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci *et al.*, "Windows azure storage: a highly available cloud storage service with strong consistency," pp. 143–157, 2011.

[15] ScriptRock, "Aws vs azure," June 2014. [Online]. Available: http://www.scriptrock.com/articles/aws-vs-azure

[16] A. E. C. Cloud, "Amazon web services," *Retrieved November*, vol. 9, p. 2011, 2011.

[17] S. Hazelhurst, "Scientific computing using virtual high-performance computing: a case study using the amazon elastic computing cloud," in *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*. ACM, 2008, pp. 94–103.

[18] E. Ciurana, "Google app engine," *Developing with Google App Engine*, pp. 1–10, 2009.

[19] P. Wayner, "Amazon vs. google vs. windows azure: Cloud computing speed showdown," March 2014. [Online]. Available: http://www.computerworld.com.au/article/539633/amazon_vs_google_vs_windows_azure_cloud_computing_speed_showdown/

[20] W. V. Winkle, "Microsoft azure vs. amazon web services: Cloud comparison," January 2015. [Online]. Available: http://www.tomsitpro.com/articles/azure-vs-aws-cloud-comparison,2-870.html

[21] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*. IEEE, 2010, pp. 1–10.

[22] M. Rouse, "fog computing (fogging)," March 2014. [Online]. Available: http://whatis.techtarget.com/definition/fog-computing-fogging

[23] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," pp. 13–16, 2012.

[24] C. reseach, "Fog computing, ecosystem, architecture and applications," 2013. [Online]. Available: http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp13078.html

[25] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.

[26] Y. Luo, O. Hoeber, and Y. Chen, "Enhancing wi-fi fingerprinting for indoor positioning using human-centric collaborative feedback," *Human-Centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–23, 2013.

[27] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.

[28] T. Higuchi, "Monte carlo filter using the genetic algorithm operators," *Journal of Statistical Computation and Simulation*, vol. 59, no. 1, pp. 1–23, 1997.

[29] A. Doucet, N. De Freitas, and N. Gordon, *An introduction to sequential Monte Carlo methods*. Springer, 2001.

[30] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 376–382, 1991.

[31] J. Haartsen, "Bluetooth-the universal radio interface for ad hoc, wireless connectivity," *Ericsson review*, vol. 3, no. 1, pp. 110–117, 1998.

[32] S. Hegenderfer, "Everything you always wanted to know about beacons," April 2014.

[33] P. Murphy, E. Welsh, and J. P. Frantz, "Using bluetooth for short-term ad hoc connections between moving vehicles: A feasibility study," in *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 1. IEEE, 2002, pp. 414–418.

[34] C. Bisdikian *et al.*, "An overview of the bluetooth wireless technology," *IEEE Commun Mag*, vol. 39, no. 12, pp. 86–94, 2001.

[35] P. McDermott-Wells, "What is bluetooth?" *Potentials, IEEE*, vol. 23, no. 5, pp. 33–35, 2004.

[36] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.

[37] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. IEEE, 2007, pp. 46–51.

[38] A. Andersen, R. Karlsen, and A. Munch-Ellingsen, "Nfc provided user friendliness for technologically advanced services," in *Human Interface and the Management of Information. Information and Interaction for Health, Safety, Mobility and Complex Environments*. Springer, 2013, pp. 337–346.

[39] K. Curran, A. Millar, and C. Mc Garvey, "Near field communication," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 2, no. 3, pp. 371–382, 2012.

[40] A. H. Omre and S. Keeping, "Bluetooth low energy: wireless connectivity for medical monitoring," *Journal of diabetes science and technology*, vol. 4, no. 2, pp. 457–463, 2010.

[41] J. Decuir, "Bluetooth 4.0: Low energy," *Cambridge, UK: Cambridge Silicon Radio SR plc*, 2010.

[42] Y. Masumoto, "Global positioning system," May 11 1993, uS Patent 5,210,540.

[43] R. Alfred Kleusberg, "The limitations of gps," March 1990.

[44] Y. Zhou and Y. Fang, "A location-based naming mechanism for securing sensor networks," *Wireless Communications and Mobile Computing*, vol. 6, no. 3, pp. 347–355, 2006.

[45] S. C. Alliance, "Bluetooth low energy (ble) 101: A technology primer with example use cases," June 2014.

[46] P. Kinney *et al.*, "Zigbee technology: Wireless control that simply works," in *Communications design conference*, vol. 2, 2003, pp. 1–7.

[47] V. B. Misic, J. Fang, and J. Misic, "Mac layer security of 802.15. 4-compliant networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[48] S. Woo, S. Jeong, E. Mok, L. Xia, C. Choi, M. Pyeon, and J. Heo, "Application of wifi-based indoor positioning system for labor tracking at construction sites: A case study in guangzhou mtr," *Automation in Construction*, vol. 20, no. 1, pp. 3–13, 2011.

[49] A. H. Lashkari, M. M. S. Danesh, and B. Samadi, "A survey on wireless security protocols (wep, wpa and wpa2/802.11 i)," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*. IEEE, 2009, pp. 48–52.

[50] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: indoor location sensing using active rfid," *Wireless networks*, vol. 10, no. 6, pp. 701–710, 2004.

[51] R. Want, "An introduction to rfid technology," *Pervasive Computing, IEEE*, vol. 5, no. 1, pp. 25–33, 2006.

[52] J. Hightower, R. Want, and G. Borriello, "Spoton: An indoor 3d location sensing technology based on rf signal strength," *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA*, vol. 1, 2000.

[53] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[54] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

[55] R.-H. Jan, H.-C. Chu, and Y.-F. Lee, "Improving the accuracy of cell-based positioning for wireless networks," *Computer networks*, vol. 46, no. 6, pp. 817–827, 2004.

[56] T. D. Pranav Rastogi, Rick Anderson and J. Galloway, "Introduction to asp.net identity," October 2013. [Online]. Available: http://www.asp.net/identity/overview/getting-started/introduction-to-aspnet-identity

[57] D. Jamil and H. Zaki, "Security issues in cloud computing and countermeasures," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, no. 4, pp. 2672–2676, 2011.