Master's degree thesis

LOG950 Logistics

Exploring Object Naming Service in the Internet of Things

Marie Svane

Number of pages including this page: 62

Molde, 31st May 2015

Molde University College
Specialized University in Logistics

Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. Failure to complete the statement does not excuse students from their responsibility.

*Please complete the mandatory statement by placing a mark **in each box** for statements 1-6 below.*		
1.	**I/we hereby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than mentioned in the paper/assignment.**	☒
2.	**I/we hereby declare that this paper** 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my/our previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: 1. ☒ 2. ☒ 3. ☒ 4. ☒ 5. ☒
3.	**I am/we are aware that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations** section 14 and 15.	☒
4.	**I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check**	☒
5.	**I am/we are aware that Molde University College will handle all cases of suspected cheating according to prevailing guidelines.**	☒
6.	**I/we are aware of the University College's rules and regulation for using sources**	☒

Publication agreement

ECTS credits: 30

Supervisor: Bjørn Jæger

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).
All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author(s).
Theses with a confidentiality agreement will not be published.

I/we hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☒yes ☐no

Is there an agreement of confidentiality? ☐yes ☒no
(A supplementary confidentiality agreement must be filled in)
- If yes: **Can the thesis be online published when the period of confidentiality is expired?** ☐yes ☐no

Date: 31st May 2015

Preface and acknowledgment

This Master thesis presents the main results of my academic work as a master student at Molde University College. The main part of the research work was carried out from January 2015 to May 2015 and has been conducted to obtain an MSc degree in Logistics.

This thesis could not have been completed without the help of the following people:

I want to thank my thesis supervisor at Molde University College, Professor Bjørn Jæger, for all the technical and conceptual discussions, if not for his drive and interest in the Internet of Things and the GS1's concepts, this thesis would not been a success.

I would like to express my greatest gratitude to Professor Hans Fredrik Nordhaug for his flexibility and technical support along the way. Not forgetting PhD student Anastasya Karalkova, GS1 France employee Nicolas Pauvre and Professor Ketil Danielsen for all the valuable inputs during the writing of this thesis.

I would also send a "big thanks" to all the employees & students at Molde University College and all my friends & family that made my work possible. Especially my boyfriend Nicklas Rosenkrantz Jørgensen, for inspiring and helping me during this process.

Thank you all.

Sincerely,
Marie Svane

Summary

The area of my study was an exploration of the IoT concept, while using the digitalization of the seafood industry as an example. I ended up focusing on the thing-oriented perspective of the IoT concept in the wild-caught fish segment.

I divided my research tasks into two parts, one practical and one theoretical. My first research task was to create an IoT platform. The research question was to find out what technology components that was needed to build an IoT platform for seafood information.

To answer that I started with an explorative methodology within the design science field, to find out what technology components and standards that was needed to explore this research question. During this thesis I have explained and installed different technologies to create two demonstrators.

My second research task was the theoretical part connected to how this IoT platform concept could be applied to add value for the actors in the supply chain. This was a theoretical research, and my findings from this study are concluded in regards to supply wide information sharing, branding & sustainability and lastly food health & environment.

Contents

Content of tables

Table of figures

1.0 Introduction

In today's world we have a high interconnectivity, and the Internet, as a source for easy and accessible information, is often taken for granted. This interconnectivity opens up for new forms of communication methods, where a thing can communicate with the general population. The idea behind this kind of communication is the Internet of Things.

The Internet of Things (IoT) is a concept that enable addressing of physical objects via the internet. Giving each thing a unique identification, some computational and communication abilities. The thing can be monitored providing geographical location, time and a range of sensor data.

To address the concept of IoT I will use the Norwegian seafood industry as an example. Not only because of its importance in the Norwegian economy, but also because of its importance for the community and it being a field currently in change.

In Norway fishing has been an important source of income since we first learned how to fish. In the 1970s, large-scale commercial fish farming started and today fishing is one of the largest industries in Norway[1]. It is a resource Norway will depend on as the oil and gas activities are declining.

There are two main fields in the fishing industry. They are wild-caught fish and farmed fish. Farmed fish industry usually is associated with the same ownership or close cooperation between actors throughout the supply chain (SC). While the wild-caught fishing industry is a slow moving, fragmented industry with multiple actors.

Both fields are strongly regulated by the Norwegian government. The government controls fishing quotas, legislate integrations and are instrumental in keeping the harvesting on a sustainable level, e.g. that quotas are upheld and fish stock is kept at acceptable levels. In addition, the authorities in Norway and abroad have strict requirements for reporting information about the fish.

With the fish industry being such a big part of the economy and of such great importance for the coastal regions, research and development in this field is valuable and needed for Norway to reach its goal of becoming the leading seafood nation in the world as expressed e.g. in the national budget of Norway for 2014.

Last year the export of fish was at 69 billion NOK (Rune Ytreberg 2015), while the export of oil & gas was at about 550 billion NOK (Oljedirektoratet 2015). This

[1] Conference: Sporing og informasjonsflyt i verdikjeden for fisk i det globale markedet, Gardemoen 16.09.14

means that fish export have now passed one tenth of the oil & gas export, and there is high optimism that this year we will surpass last year's results in the fishing industry.

The IoT development combined with the ongoing digitalization of the seafood supply chain has created possibilities for new information driven services. This thesis explores these issues by demonstrating how IoT-technologies can be combined with new global standards for labelling of seafood or fish transportation units.

1.1 Background

The concept of IoT is however quite fuzzy with many parallel initiatives labelled as IoT application. I divided the initiatives into two different perspectives, the "network"- and "thing"- oriented perspectives. Both the network and thing perspectives mainly follows the reasoning that was suggested in the article "*The Internet of Things: A survey*" by Atzori, Iera and Morabito (2010).

With the use of the Norwegian wild-caught fishing industry as an example, I will explore the thing-oriented perspective of IoT application. The main reason for this is the current use of labels and possibly RFID tags to identify fish transport units. These things do not have the computational or communication abilities as required by the "network"-IoT applications.

Since the 1950's the biggest catch vessels have kept catch logs, both for research and statistics purposes. 40 years later, when the fishing quotas arrived, this became a way to check how much was hauled (Fiskeridirektoratet 2015). The digitalizing of the information handling in the fishing industry started around year 2000, first with the positioning of the fleet. Later, in 2004, the possibility to e-mail catch data was introduced. In 2010, the electronic catch log system was implemented, and this lead to an electronic landing or contract note, instead of doing it manually (Directorate of fisheries 2010). Digitalization of the contract notes lead to an easier and quicker way to share information.

Another of these governmental changes to the fishing industry was the food law passed by the European Union in 2002 (EU). This law demands that company's record whom they receive food from and to whom they deliver foods. Since Norway is a part of the EFTA agreement, this had an impact on the Norwegian food laws also. With the added demands on documentation of point of origin, it put forward a demand for food tracing in the food value chain.

Recently the implementation of the Norwegian standard NS 9405:2014, a standard set for the labelling of fish-crates or pallets[2], gave us the opportunity to follow the fish from start to the retailer. Currently, large actors like Marin Harvest has or are in the progress of implementing the standard (Innovasjon Norge 2014). This globally unique labelling of fish crates has the potential for receivers of the fish crates (the retailers) to use the label to look up extended information about the fish-crate they have received. This information can be given to the end-customers, an important feature to promote Norwegian fish globally. At the same time a lookup from retailers enables the fisheries to collect information of their customers/retailers. Over time, this information can be used to tailor the information provided from the suppliers, to the various needs by retailers worldwide.

With all these changes in the industry, and the need for effective information sharing across the supply chain, the idea is to apply the “Internet of Things” (IoT) concept to enhance information flow and create benefits. With the IoT solution, it is possible to get tracking data from the whole supply chain, which in theory could provide an enormous amount of information at the disposal of the different actors in the supply chain.

1.2 Problem description

My focus is on exploration of IoT and the digitalization of the seafood industry.
In the seafood industry I have decided to focus on the wild-caught segment and to start at the fisheries, and skip all the different steps up until the end retailer (see Figure 1). Both these positions can be considered as “gatekeepers” in regards to what is shipped in what crates, and what information that is shared to the general population.

Figure 1 Fish value chain

[2] Conference: Sporing og informasjonsflyt i verdikjeden for fisk i det globale markedet, Gardemoen 16.09.14

In the IoT concept I decided to have my focus on the thing-oriented perspective. The application of the ONS concept is of interest, mainly because there is just one currently in operation, located in France and administrated by GS1 France[3], and therefore it provides a good opportunity for further research and development.

I divided my research questions into two parts, one practical and one theoretical. Both parts require different methods to answer the research questions, and are briefly addressed in the chapter about Research method.

The research questions are:

1. *What technology components are needed to build an IoT platform for seafood information?*

To answer this question I will build a demonstrator using all main components and explore different technologies. Technologies like Electronic Product Code Information Service (EPCIS) and Object Name Service (ONS), where the ONS is what connects us to the thing-oriented IoT concept.
These and other technologies are further explained in Previous work.

2. *How can the platform be used to add value for the actors in the supply chain?*

Here the research method will be a literature study of the value creation initiatives the implementation of IoT type technology adds.

1.2.1 Research question 1

The creation of demonstrators with an EPCIS and a local ONS. The initial goal is to find extended information about a Serial Shipping Container Code (SSCC). When I have created and tested the local ONS, the added goal becomes to either connect it to the federated ONS, or find out how an F-ONS would work theoretically. To achieve this, I need to describe the setup of the systems and create one or more demonstrators.

I made a general explanation of what I want to accomplish, that I further divided it into three examples. As earlier mentioned, this is a simplified version and in the real world it is significantly more complex.

Fish arrives at the fisheries
The fisheries register the fish crates, filled with fish, as packed and shipped in their EPCIS system. The fish crates unique identifier is the Serial Shipping Container

---

[3] Personal communications with GS1 France employee Nicolas Pavure.

Code (SSCC). The fishery have then made an ONS IoT platform, so customers can access the data accumulated or entered into the EPCIS repository.

Fish arrives at the retailer
When the retailer receives the fish crate, assume that the retailer wants to know if there is any extended information about the fish crate that he received. The retailer can use SSCC to look up information regarding the fish. The fishery in questions has a search webpage they administer, that is connected to the local ONS. The webpage is shared with their customers, either as a printed address on the crate or as a QR-code.

Query of the fish crate
When the webpage has received a query, it then knows that the retailer wants information about the specific fish crate, and identifies the EPCIS system of the queried SSCC code. Then the webpage answers with a link to another webpage with the pre-approved information that is available about that particular fish crate.

1.2.1.1 Example 1 description

In this example, I am making a local ONS, i.e. setting up and configuring an ONS enabled DNS server. below you can see a visualization of the setup.

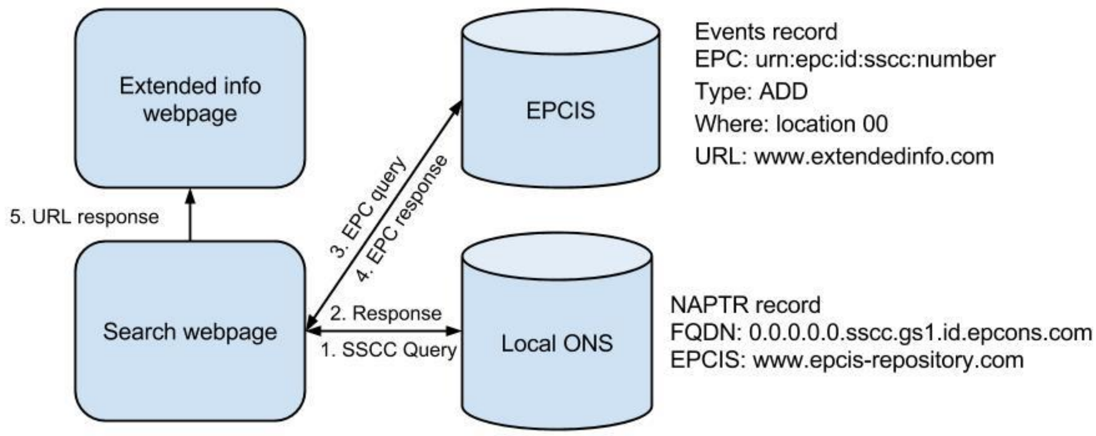

Figure 2 Example 1 setup simplified

In the EPCIS repository, I will register events with a common link to a semi-dynamic webpage I am making for extended information. The last thing I will do is to create a search portal page to collect that URL out from the EPCIS.

1.2.1.2 Example 2 description

Here I will use the same setup as in example 1, but instead I will rather count the number of events that are registered in the EPCIS on that specific SSCC. This works better for generic searches, since it is not looking for a URL that was added to the events. In this example you can later add a database, with extended information, where you use certain information in the EPCIS repository to make lookups in that database.

1.2.1.3 Example 3 description

In this example, you are not interested in showing extended information, but rather interested in connecting yourself to other ONS. Here, a search of a SSCC, if not found in your local ONS, will lead you to the correct ONS location in the Federated ONS network. Then I will further research how to connect a local ONS to a foreign ONS, with the use of Non-Terminal DNS Name Redirection (DNAME).

1.2.2 Research question 2

For the last research question, I will connect these demonstrators to the different value adding aspects that the implementation can add. This part will be theoretical, since I have not been able to test this system in a company. In addition, it will be more focused toward technology in general and not directly to ONS, because of its limited implementation.

1.2.2.1 Value creation initiatives

In value creation initiatives, it is not always easy to pinpoint where they create value or to whom. In regards to the fishing industry, we might think society's requirement is most important, especially in consideration to food safety and the added demand for traceability. This because it is shown to be an increasing trend for the public to be aware off (Li et al. 2014). In addition, the value perspective of each actor is considered.

It is worth mentioning the importance to focus on the positive sides, like the traceability, added room for marketing (marketing differentiation) and the point of origin. The goal then becomes to look at how these factors can add value in the supply chain. This can be in regards to ecological friendly seafood (no preservatives and sustainability), short travelled food or proof of "point of origin". This means that the final step of the research task will be to look at the different dimension that technology can add, to see if it has any value adding possibilities.

1.3 Research questions

As stated in the problem description, the research questions are as follows:

1. What technology components are needed to build an IoT platform for seafood information?
2. How can the platform be used to add value for the actors in the supply chain?

1.4 Research method

Both the IoT concept and the wild-caught fish supply chain come off as slightly unclear. The IoT concept, because there are multiple perspectives of how to apply it, and what needs to be done to make it a reality, and the wild-caught fishing industry with its fragmented supply chain and multiple actors.

This makes it necessary to apply an explorative method with focusing on just a part of the supply chain. Janet M. Ruane (2005, 12) defined exploratory research as a type of research that is “*typically conducted in the interest of "getting to know" or increasing our understanding of a new or little researched setting, group, or phenomenon.”* Within the explorative method I followed a design science approach, which is exploring through design (Holmström, Ketokivi, and Hameri 2009). It seeks to explore new solution alternatives to solve problems by constructing a solution.

With the explorative method as base, I will also apply descriptive and qualitative measures to this thesis. Descriptive methodology when explaining what I am doing to explore the concept of IoT, and qualitative methodology when I am studying the value creation initiatives the IoT application can add.

2.0 Previous work

In this section, I will describe the background of this thesis. From previous work, to certain concepts that are vital to form an understanding of the solutions and standards that this thesis is based upon.

First, we start with a quick explanation of what the internet is, before I move on to the concept of “Internet of Things”. This to give you a basic understanding of the environment this thesis is built on. Then, I address the GS1 standards, procedures and systems, that later will be applied to create the demonstrators. Lastly, I will briefly mention some trace projects and other work, which can be related to what I am doing in this thesis.

2.1 Internet

The Internet have a very rich history, is a worldwide, publicly accessible network of interconnected computer networks that transmit data with the use of the standard Internet Protocol (IP). It is a "network of networks", and connecting a plethora of interconnected networks, where each network is identified by a unique IP-address.

Over the years, as computing and networking capabilities have improved, each computer can handle the networking duties leading to the contemporary situation where each computer has its own IP address. Thus, the Internet has turned into a network of IP-enabled computers. A trend is to integrate IP-enabled computers into devices like refrigerators, ovens, printers, etc. thereby creating an network-*oriented* Internet of Things (Atzori, Iera, and Morabito 2010). This subject is further discussed in the Internet of things chapter.

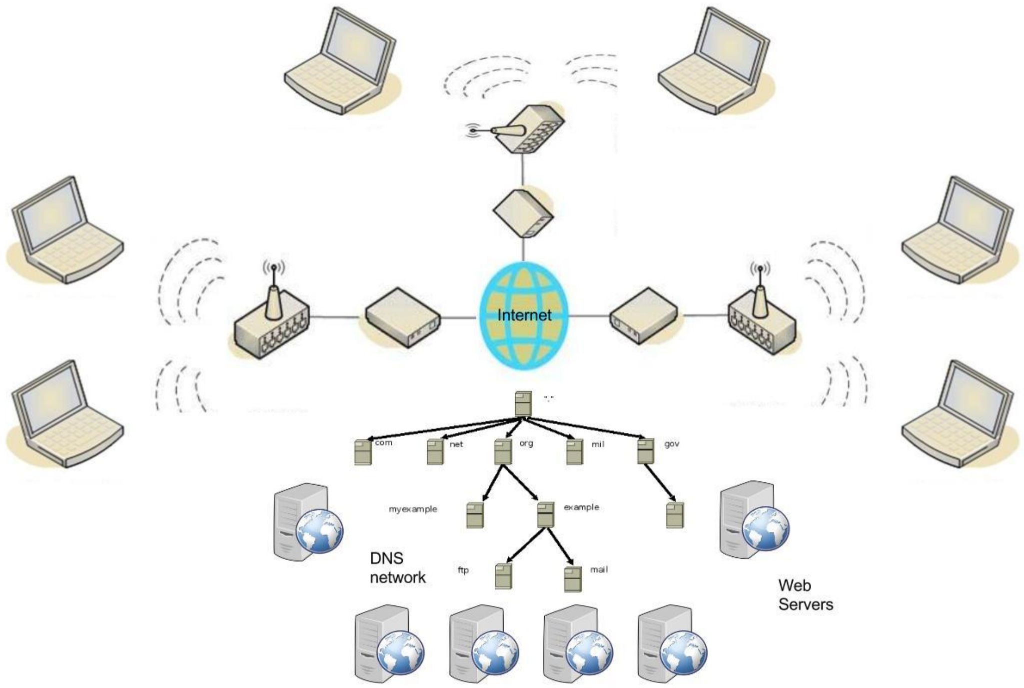

Figure 3 An illustration of how the internet works

From Figure 3 we see that the internet consist of a multitude of connected networks. However, to help make readable domains or addresses, a Domain Name System (DNS) network is required. This hierarchical distributed naming system translates the IP addresses into domain names, and is the directory service of the internet.

2.2 Internet of things

The Internet of Things (IoT) was first suggested by the Auto-ID Center, a collaboration between multiple universities and MIT (Sarma, Brock, and Ashton 2000). Their idea was that each object or item would get a low-cost Radio Frequency Identification tag (RFID) attached to them, which was set up in a global network to create a universal standard for both identifying products and sharing information.

In 2003, the Auto-ID Center divided into two different units, EPCglobal and Auto-ID labs. Where EPCglobal is a part of GS1, who have developed the standards used in this thesis, and Auto-ID Labs, who are still composed of several of the original universities, are working with further development of the IoT infrastructure (Cambridge Auto-ID Lab 2015).

In the IoT it is common to divide products into two groups, “smart” objects and “dumb” objects. An example of a “smart” object could be your fridge, with an embedded computer. Where the “dumb” objects are all the groceries that you have in your fridge. The idea is that you control your fridge over the internet in regards to being informed of temperature, receiving lists of suggested items that needs refilling or which “dumb” objects needs refill.
Having illustrated an idea of how the IoT concept can work, it leads us to the different types of looking at the IoT.

2.2.1 Perspectives in the Internet of Things

The concept of IoT consists of different perspectives, where the two main methods of thinking of the concept is the network-oriented and the thing-oriented perspective. The third perspective is the semantic web, which is the data about the data (W3C 2015).

In the article “*The internet of things: A survey*” by Atzori, Iera and Morabito (2010) they discuss these perspectives, and comes up with a model that tries to explain what the IoT should or could be (see Figure 4). In this model they suggest that the IoT should be a convergence of the three perspectives mentioned above. We also see that the GS1 suggestion, on how to deal with the IoT concept, is firmly placed into the things-oriented perspective. While the vision to the network perspective, is equipping everything with an IP-stack.

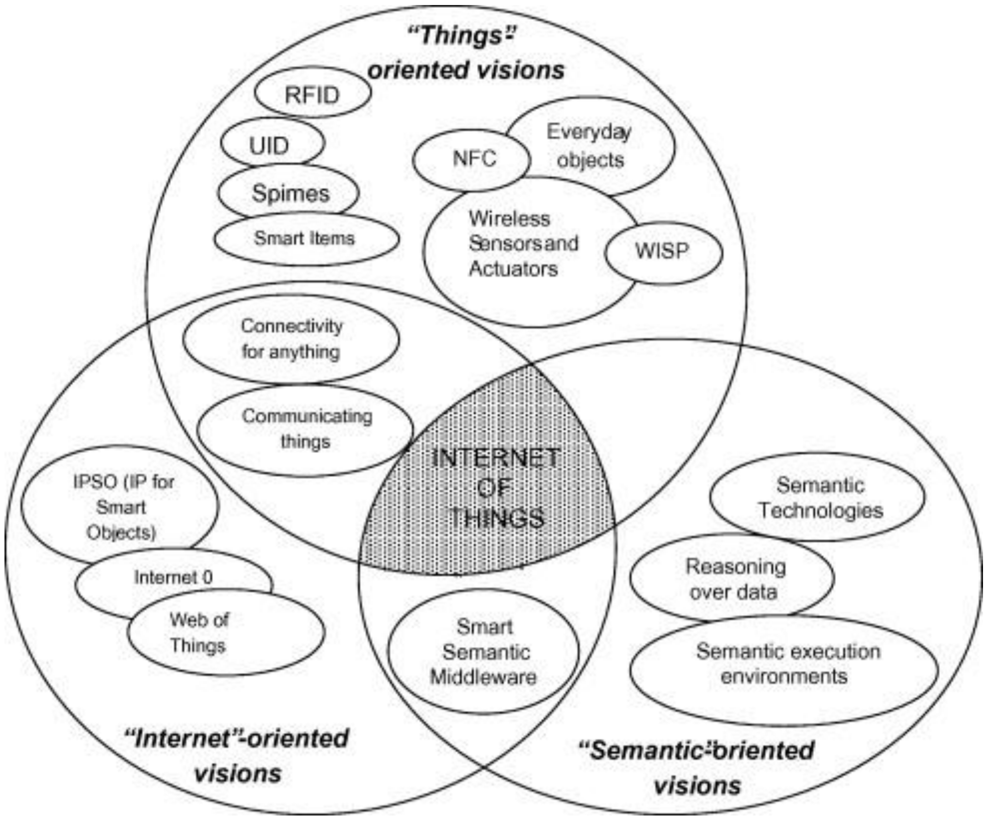

Figure 4 IoT paradigm convergence.

Source: Atzori, Iera and Morabito (2010)

2.2.1.1 The "smart" objects – network- oriented

The network-oriented want to give every object an IPv6 address. For this to work for more than just a label, you need an embedded computer in each object. With this change implemented, you make the "dumb" objects smart. Most commonly through the already existing data chipset in the electronic devices, or adding "smart" sensors into the "dumb" objects.

Using the previous example from the fridge, to illustrate this concept further, by adding a "smart" sensor that are equipped with an IP-stack to a carton of milk. This sensor will measure how much milk is left, give warning about temperature, or if the expiration date is close. Using this method, it will guarantee that it works in the existing internet scheme illustrated in Figure 3. The drawback however is that it will require both time and money, possible more than its worth, to make the "dumb" products smart. Therefore, for now it is about connecting the already "smart" (everyday) objects up to the internet.

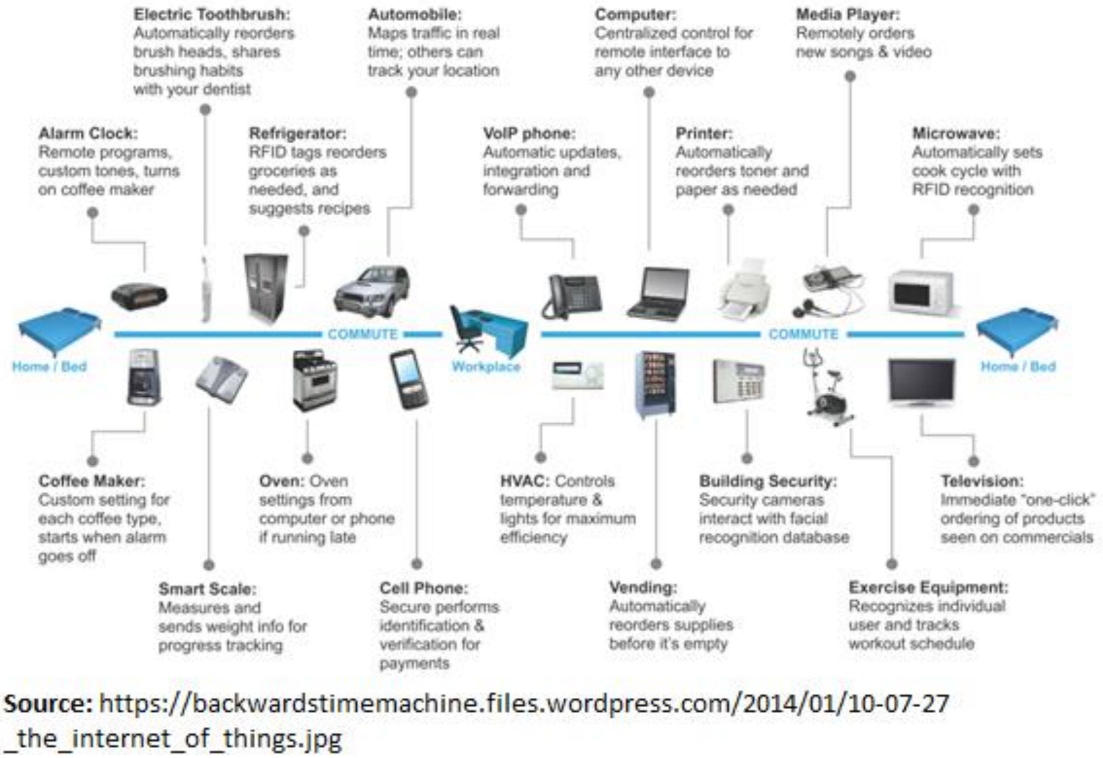

Figure 5 Example of network-oriented Internet of Things

As can be seen in the figure above, the entire picture is full object that has the possibility to have an embedded computer integrated. Therefore, with the use of apps the user can control or check if everything is functioning as required over the internet.

2.2.1.2 The "dumb" objects or non-electric – thing- oriented

In this thesis I mainly discuss the role of the "dumb" objects or non-electric, this in regards to the fish-crates that can be electronically addressed on the internet (see figure below). In the logistic world the thing-oriented perspective is key.
There are multiple actors in this field, that works on different methods of how to get the IoT concept to work within the thing-oriented vision (Atzori, Iera, and Morabito 2010). Where the most important actor for me is GS1.

Figure 6 Illustration of non-electric things for the thing-oriented IoT

2.2.1.2.1 GS1 (ONS)

GS1 decided that instead of making the “dumb” objects smart, they rather keep the object as it is, and add a translating step (EPCIS) to make the objects compatible with the internet. Because of this, a few changes in the existing internet schemed had to be done, and is further explained in chapter 2.6 about the Federated Object Name Service (F-ONS).

In this infrastructure, you do not have a direct communication to each object, but rather set up checkpoints where the object can be added, observed or deleted. These checkpoints are connected to the EPCIS server, which works as a translation step for us.

2.3 Object Identification in a things oriented perspective

To differentiate between objects, it is necessary that each identification is globally unique and machine-readable to enable automatization of the work.

2.3.1 Techniques of Object Identification

There are two different techniques present today can hold information about an object or a thing. The first technique is the printed barcodes or datamatrix, and the other is RFID tags (see Figure 7). However, the method of object identification is not the main subject in this thesis, but rather a means to an end. The important part is to identify the object in a deterministic way.

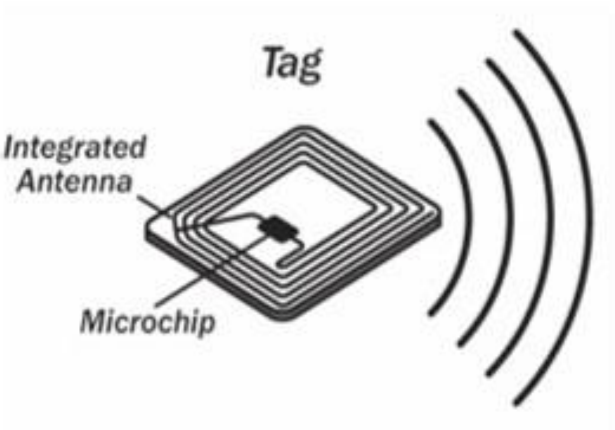

Figure 7 Example of RFID tag.

Source: http://cdn.barcodesinc.com/cats/rfid-readers/tag.jpg

2.4 GS1 Standards

GS1 history started on the 26th of June in 1974, all because they decided to put a barcode on a packet of chewing gum. They now manage the barcode standard used in supply chains all over the world. Even though they started with barcodes, it now reflects the different needs we have in today's world. GS1 is a non-profit organization, with departments in almost every country and a huge network they cooperate with (GS1 2015).

With the knowledge that the fishing industry recently had implemented a standard for labelling of fish crates and that GS1 is a provider for these kind of codes, a search on the GS1 homepage lead me to what I needed to know about the SSCC codes and how to register them in my EPCIS repository. I also got the information about how to transform them, so I could use them in the ONS construction.
GS1 have multiple barcodes and product identification methods, but I will only apply a few of those methods in this thesis.

2.4.1 Barcode types

To encode a product group with an identification number, a barcode is often used. The most common and used barcode is EAN-13. There is also a rising use in 2d barcodes such as the DataMatrix. Both of these types are GS1 standards.

In this thesis a Serial Shipping Container Code (SSCC-18) barcode is used, and are on slightly higher level than the Global Trade Identification Number (GTIN-13). The SSCC-18 is used on transportation packs, unlike the GTIN-13 that is used on consumer packs. While the EAN-13 have very limited encoding possibility, the SSCC-18 can store more.
You'll find an example of a SSCC-18 barcode in the figure below.

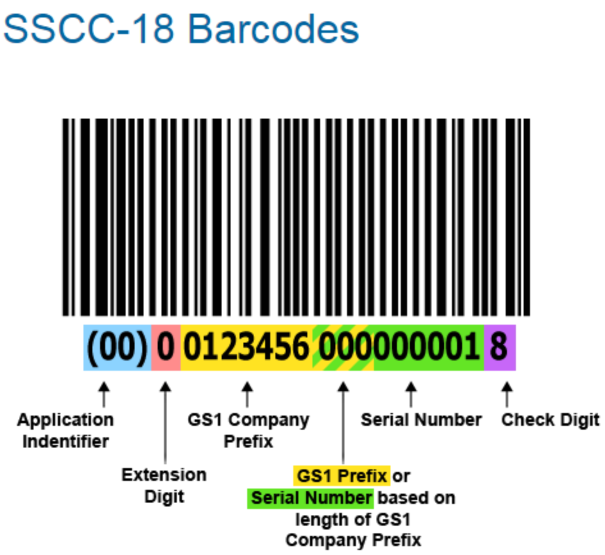

Figure 8 Example of a SSCC barcode and explanation

In total, the SSCC-18 will consist of 18 digits plus the Application Identifier (AI). The AI tells the reader what kind of barcode it can expect to read, while the Extension Digit is a number between 0-9, in Norway the number 3 is usually used. The next 16 digits tells us what country, company and the serial number assigned for the transportation unit, while the last number is a calculated check digit.

The DataMatrix on the other hand is a two-dimensional barcode, with the possibility to encode both numbers and alphanumeric characters (see Figure 9). The maximal limit for encoding on these DataMatrix is 3116 numbers, or 2335 characters. This format requires a reader that is configured to 2d scanning.

Figure 9 DataMatrix example picture

Without a numerical representation added to this label, humans without a scanner cannot read it.

2.4.2 Product identification

Too identify a large number of products, you need a clearly specified and uniquely formed standard are needed. With the diverse need for identification, GS1 have many different product identification solutions. For this project the SSCC and Electronic Product Code (EPC) is used.

Example 2.1: GS1 System
SSCC-18: (00)370101000000000019
SSCC: 3701010.00000000019
EPC: urn:epc:id:sscc:3701010.0000000001

As explained the SSCC contains 18 digits, where the second and up to the fourth digits identify the country prefix. In Norway, this prefix is 70.
GS1 Norway is responsible for managing the number series and assigning them to companies.

Example 2.2: SSCC

#Company Digits	AI	I	CC	CID + number series	CD
4-digits	(00)	3	70	LLLLSSSSSSSSSS	K
5-digits	(00)	3	70	LLLLLSSSSSSSSS	K
7-digits	(00)	3	70	LLLLLLLSSSSSSS	K

AI = Application identifier, I = Identifier/Extension digit, CC = Country Code
CID = Company Identification =>
L…L = Company Identification number + S…S = Product/article number
CD = Control Digit => K

2.4.3 Translating product tags

GS1 have algorithms to translate tags between different GS1 standards. For this project, two types of translation was needed. The first one was transforming the SSCC to EPC and the second was SSCC to a domain name (GS1 2013).

2.4.3.1 SSCC to EPC

If you read the barcode manually, the AI and the check digit are dropped from the SSCC-18, you are now left with the raw SSCC code. To get it into a EPCIS server you need to start with urn:epc:id:product code type:, in this case sscc. You are now left with urn:epc:id:sscc:, before you append the 17 first digits of the raw SSCC code.

Example 2.3: Translation procedure EPC

1. (00)370101000000000019
2. SSCC: 3701010.00000000019
3. EPC: urn:epc:id:sscc:3701010.0000000001

2.4.3.2 SSCC to domain name

In order to do lookups via the DNS infrastructure, the product identification need to be made into a Fully-Qualified Domain Name (FQDN). When the SSCC-18 is read or scanned, it will look something like this: (00)370101000000000019.
To transform this into a domain name, the scanner application needs to have the knowledge of one of the peer-roots, so that its ending can be appended to the created domain name.

In regards to peer-roots, preferably the closet one geographically should be chosen, making the lookup as fast as possible. For this instance it would mean registering `onsepc.no`, or the use of `onsepc.se`. However, in the project `onsepc.com` peer-root was chosen, since no regional peer-root had been registered and uncertainty if the `onsepc.se` is a valid ONS peer-root domain name (GS1 2014b).

Example 2.4 Translating from SSCC to Domain name

1. **(00)**37010100000000001**9**
2. 37010100000000001**9**
3. 31000000000010107
4. 3.1.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com.

The steps to transform a SSCC to a domain name can be seen in the example above.
Step 1 is the original product identification number; step 2 is removing the AI and dropping the calculated end digit from the SSCC code, while step 3 is to keep the extension digit in place then reverse the rest. The last step is to separate all the digits with dots, and then append with `sscc.gs1.id` and the peer-root domain name.

The result of this translating procedure are to be used for a lookup in the Name Authority Pointer (NAPTR) in the DNS, and is a type of resource record. What and how the NAPTR RR function is further explained in chapter 2.6.2.

2.5 EPCIS

The Electronic Product Codes Information Services (EPCIS) contains historical event data about the EPCs (GS1 2014b). Multiple event types can be generated, to provide a serial level tracking of the event data, the events are EPCIS Aggregation, Object, Quantity and Transactions. This gives us the chance to follow what happens to the product, from production to end-user.

The EPCIS repository is created to store the events, but not to delete events, so as long as the database exist, the events are always present.

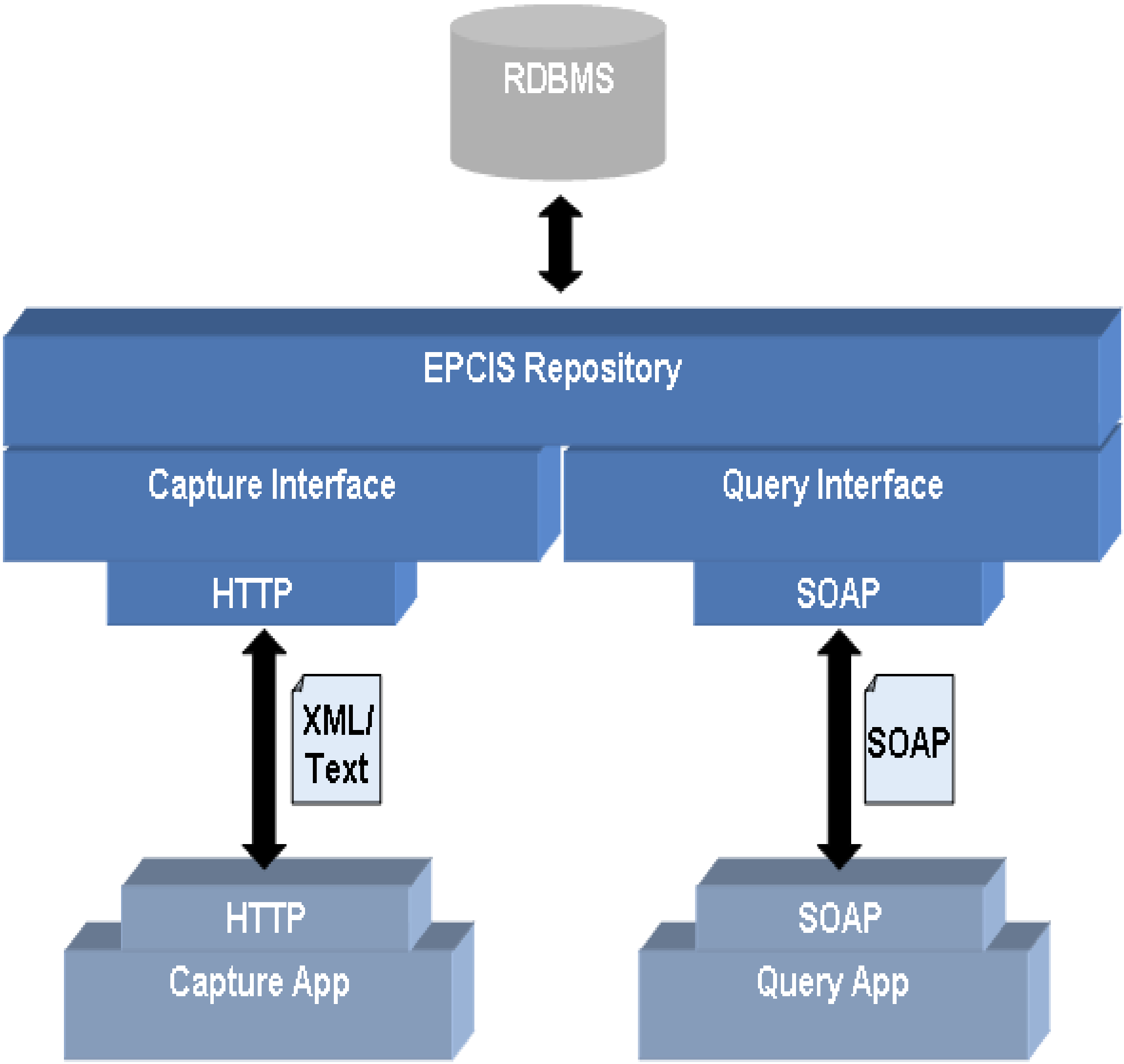

Figure 10 Fosstrak EPCIS Implementation

Source: https://code.google.com/p/fosstrak/wiki/EpcisMain

The SOAP interface is used to access the data from the repository, while a HTTP Post where the payload is an XML file is used to add events. All these events are stored in a MySQL database (see figure above).

2.5.1 EPCIS events

An EPCIS transaction event contains information about what, when, where and why. It is constructed after a set of rules defined in the Core Business Vocabulary Standard (GS1 2014a) to ensure that all parties that exchange EPCIS data will have a common understanding of what is registered.
In the table below, the different data fields are explained.

Table 1 EPCIS description

EPCIS Event Field	**Dimension**	**Description**
EPCs	What	An instance-level identifier structure. That SHALL be represented as a URI.
Action	What	Denotes if the event was added (again) or observed.
Occurred	When	Timestamp of event, including time zone offset.
Recorded	When	Time recorded into the EPCIS
Read Point ID	Where	The location where the EPCIS event took place. In this case the default is used
Business step	Why	Specifies what business process step that was taking place. E.g. ‘production’ and ‘shipping’.
Disposition	Why	Specifies the condition of the subject of the event. E.g. ‘pending shipping’ and ‘in transit’.
Business transaction	Why	In this case this is used to add a URL element to the event, that later can be found again.

2.6 Federated Object Name Service

Federated Object Name Service (F-ONS) is a Domain Name System (DNS) based structure, and was developed by Afnic and the French GS1, for resolving product identification numbers (AFNIC 2013). This research resulted in the new global standard Object Name Service (ONS) in 2013 from GS1.

A federated version of ONS is a system where the root is built up of several peers to share the load of the ONS root, and to prevent a country to have full control (GS1 2013, 12). An example of how it can be built up is shown in the figure below.

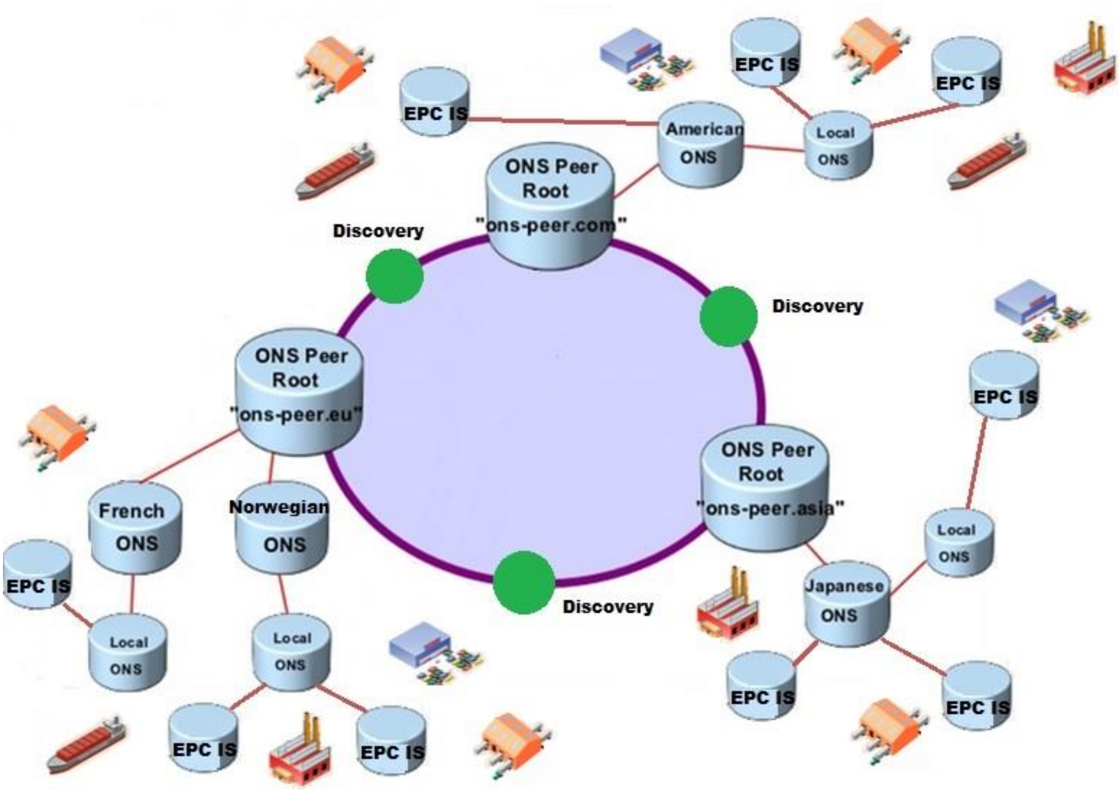

Figure 11 Federated ONS Architecture

This method ensures that the infrastructure is set up in a hierarchical way, where multiple peer roots have the same architectural privileges. Another advantage is that each company can administer their own local ONS, connected to the country's ONS and further the regional peer root.

AFNIC also suggest that the administration of which country that is located under which peer-root can be managed with the use of a Common Mapping Table (CMT), where with the use of the GS1 country prefixes the different countries can be identified (Balakrichenan, Kin-Foo, and Souissi 2010). This mapping table need to be manual managed by the members in the ONS federation.

As can be seen in the figure above, the next level on the F-ONS is the country level. This means that all the countries need to have their own ONS node, and delegate all the company name-spaces that shall be allocated under the country code. This act as a pointer to the companies own local ONS servers, enabling them to administrate their name-spaces themselves.

2.6.1 Dynamic Delegation Discovery Service

In the "Dynamic delegation discovery system (DDDS) part one: The comprehensive DDDS" by Mealling. The DDDS is explained like this:

"The Dynamic Delegation Discovery System is used to implement lazy binding of strings to data, in order to support dynamically configured delegation systems. The DDDS functions by mapping some unique string to data stored within a DDDS Database by iteratively applying string transformation rules until a terminal condition is reached (Mealling 2002)."

In regards to the ONS, the database used for the rules is DNS. This means that the key has to be a valid domain name for finding the correct rule in the database, and these valid domain names will be found in the NAPTR Resource Records (Daniel and Mealling 2000).

2.6.2 Naming Authority PoinTeR Resource Record

The Naming Authority Pointer (NAPTR) Resource Record (RR) was originally made by the IETF URN working group for storing rules in DNS for handling re-delegations of Uniform Resource Identifiers (URIs) (Daniel and Mealling 2000), and later adopted by EPCglobal in their ONS standards document (GS1 2013).

The NAPTR record consist of six different fields (see Table 2), where four of these are in active use according to the ONS standard (GS1 2013).
As we can see in the table below and described in the standards, the order field is not used and shall therefor have the value 0.
If multiple records exist, the preference field indicates what record is prioritized, where 0 is highest, and the flag field indicates what the regexp field contains.

In ONS a "u" indicates a URI, while a "t" indicates that regexp contains plain text. The service field shall contain a URL that indicate what type of service that can be found in regexp, and in the ONS case it is usually an EPCIS server.

The two last fields are the regexp field and the replacement field. The regexp field always starts with !^.*$!, that means "match anything", then the URL followed by an exclamation point ("!"). While the replacement field is not used in the ONS setup, and a period (".") is the default value.

Table 2 Example of NAPTR record

Order	Pref	Flags	Service	Regexp	Replacement
0	0	U	Service URL	!^.*$!http://example.com/cgi-bin/epcis!	. (a period)

2.6.3 Non-Terminal DNS Name Redirection

To get access to the corresponding ONS Peer roots, and gain the right peer domain name string, the DNAME RR are used (Crawford 1999). This enables us to redirect a query from our domain name, to another domain name.

There are certain advantages with the use of DNAME, since it can redirect entire sub trees (GS1 2013). below you can see an example of a DNAME record.

```
$ORIGIN 1.0.7.sscc.gs1.id.onsepc.com.
$TTL 86400 ; 1 day
@ IN SOA ns.1.0.7.sscc.gs1.id.onsepc.com. example.com. (
                                    2008010801 ; serial
                                    21600 ; refresh (6 hours)
                                     1800 ; retry (30 minutes)
                                  1209600 ; expire (2 weeks)
                                    86400 ; minimum (1 day)
)
  IN NS root1.preprod.onsepc.fr.
@ IN DNAME 1.0.7.sscc.gs1.id.preprod.onsepc.fr.
```

Figure 12 DNAME RR

2.7 Trace Projects and other work

There are a few food traceability projects already existing, where with the use of RFID, barcodes or QR-codes, is connecting up to an EPCIS and perhaps an ONS, where information is gathered throughout the supply chain and shared. Where one is a German projects, and another is a Swedish project that was done with local fish.

In regard to other work, there are different projects both in regards to developing the different perspectives in IoT and working with the ONS solutions from GS1. However, there are too many to mention all of the different projects existing in these fields. Instead, I am mentioning shortly the two projects that have helped me.

2.7.1 fTrace

An example of this is fTrace, which is a food traceability project from Germany. They have implemented a system where each step in the supply chain, identify themselves and only add what they have done with the product. This to prevent redundancy before shipping it to the next part in the food value chain (GS1 2014c).

2.7.2 eTrace

Another project was the Swedish eTrace program; this was done by using an EPC information service (EPCIS), and following the fish from sea to sale, where they offered the extended information about the fish for verification to customers. Results showed that better information visibility, lead to increased sales (eTrace 2013).

2.7.3 A Master Thesis on F-ONS

Three Swedish students wrote a master thesis called *"Investigating Federated Object Naming Service as Directory Service for the Internet of Things"* (Fredriksson, Pehrson, and Sandmark 2011), where they cooperated with GS1 France and Sweden to build two demonstrators. They did this to show the benefits of having the information about the products on the internet, and the robustness and scalability with the use of a directory service in the internet of things.

2.7.4 Oliot ONS project

Oliot is the abbreviation of Open Language for Internet of Things, and is a spin-off from the Fosstrak project. Oliot's goal is being a worldwide standard based IoT Infrastructure Platform, and they are aiming at a complete implementation of the GS1/EPCglobal standards (Oliot 2014a).

Oliot have published how they have setup an ONS in Bind (Oliot 2014b). You get a chance to copy or just look at how the different files you need to change in the original Bind configuration.

3.0 System Description

To implement this demonstrator, I had to take several decisions about different programs and systems. With limited time and funding, I was in need of free and accessible programs. Freeware was my solution to this.

I will start this chapter by defining what I mean with freeware in this thesis, and say something about the advantages and disadvantages. Then I will go through the different steps of installing the EPCIS, ONS and the creation of the demonstrations. I will illustrate most of the steps with pictures or codes, to make the explanation of the setup approachable.

3.1 Freeware

Freeware is defined as a software program that can be used at no cost, unlike a licensed system, where you pay for the right to use the software program.
In this case freeware is used as a term to describe all type of software that is free to use, may it be free software, shareware or open source software (The Linux Information Project 2006).

There are certain advantages and disadvantages connected to choosing freeware instead of buying a licensed system.
The biggest advantage is that it is free and anyone can download and use it. A slight disadvantage is that you need to know what programs you should download and in what order you should install them; furthermore, you will also need to have some prior knowledge of how to fix the problems you might encounter during the installation process.
The second advantage with freeware is that if you are a programmer, you have the chance to play with the source code and contribute to the whole community. This can however also be a disadvantage if you are not used to an open source code, if it is still in the experimental phase or you don't have the experience in how to customize programs to meet your specific needs.

3.2 EPCIS

For this master thesis fosstrak, an open source RFID platform, was used to register events about a RFID element. They have a detailed how-to-install guide on their website (Fosstrak 2012).

During this master thesis, three different EPCIS installations were used. The first one was installed on my personal laptop running Windows 7. The second was a pre-existing installation on the school network, used for educational purposes in RFID seminars, and the last one was a fresh installation on a Debian GNU server and is also the installation used for the end results in this thesis.

The installation progress was quite straight forward, and my experience with the installation on my laptop is explained in detail below.

3.2.1 Installation

The first program that was required to install fosstrak EPCIS repository, was a web-server that supports fosstrack. I used an Apache Tomcat open source web server with servlet of version 5.5 or higher (Apache Tomcat). For this, I also needed a Java Development Kit (JDK) or Java SE Runtime Environment (JRE). On this project I went with JRE.

The Apache Tomcat version that was installed on my Windows 7 laptop, was the 6.0.43 version (32-bit/64-bit Windows Service Installer). I then continued with verifying that Tomcat was up and running, before moving to the next step in the installation guide.

The next step was to download and copy a WAR file into the webapps directory in Tomcat, then perform a restart and get the repository packed out. After checking that this had occurred, I moved on to download and install MySQL server and the MySQL connector/J driver.

The install guide instruct you in how to do this and how to set up a database. I used the default setup as it was outlined in the user guide, and had now created a database called "epcis". With limited experience of MySQL syntax/language, I had problems with setting up the database schema, or check if the database was connected.

A restart of the environment and two commands in the MySQL command line client, `USE epcis;` and then `SHOW tables;` alerted me that there were no tables was present in the database "epcis". A new try with `SOURCE <path-to-epcis>/epcis_schema.sql` was attempted, and this time the schema was loaded into the database.

Now the tables command could be rechecked and potential error messages received under this step could be addressed and fixed. The result of this process can be seen in the figure below.

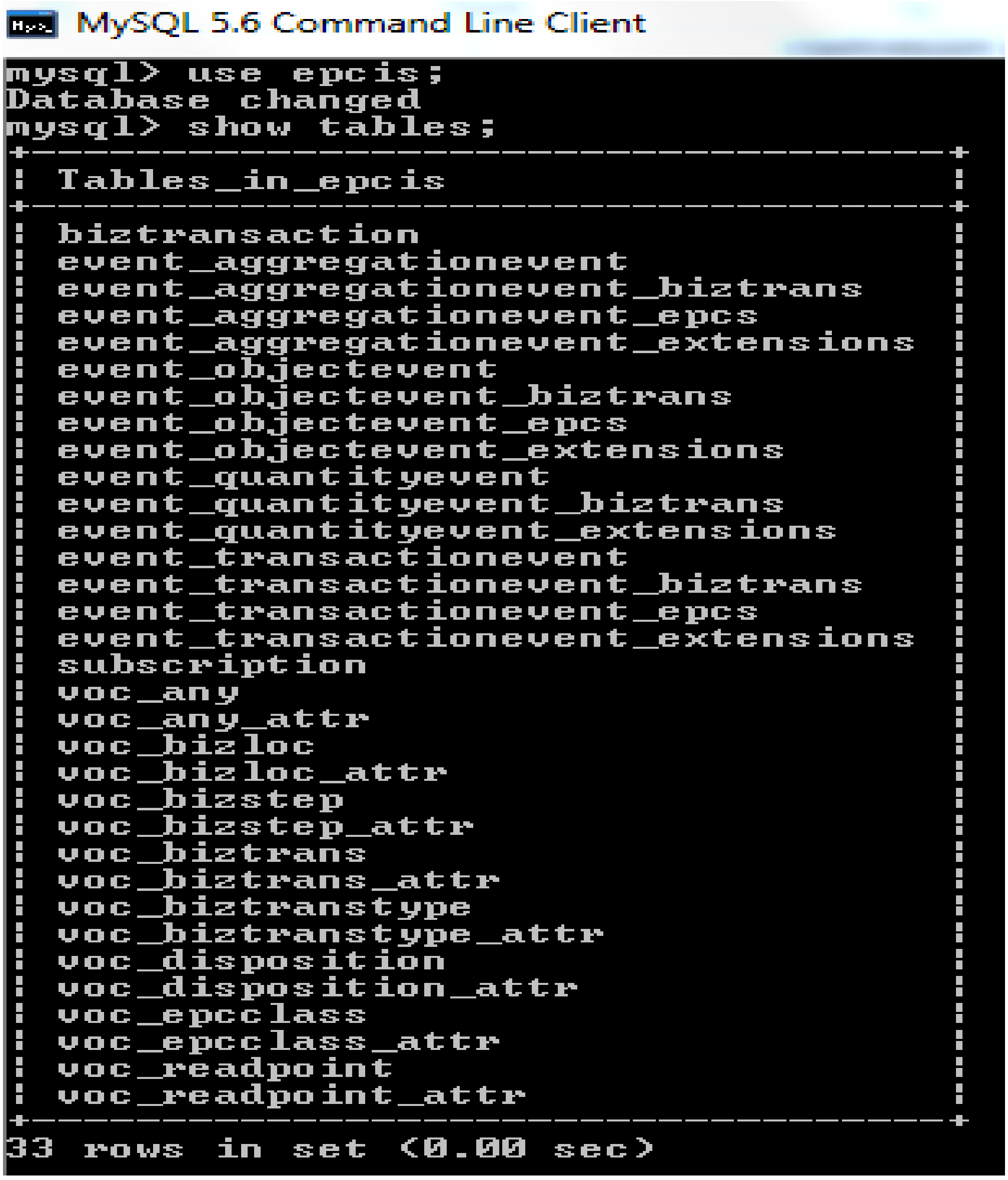

Figure 13 MySQL Command Line Client shows how many tables there is in the database epcis.

Now as the database is ready to be populated with data, a test set was loaded in. This confirmed that the database was in working order.

I now had to check that the repository was connected to the newly created database. This was done through an xml document in the repository folders, inside of the Tomcat installation folders. The xml document had to be checked to verify that the username, password and url corresponded with what was created during the MySQL setup. This was further tested, and then verified by entering http://localhost:8080/epcis-repository-0.5.0/ in the web browser.

The repository had now been successfully installed, and almost ready for use. The last parts that needed to be downloaded was the EPCIS capture and query clients. When those clients were downloaded and packed out, the opportunity to generate and query the events was fulfilled.

If problems arise with the capture client or the HTTP version, the answer was to log into Tomcat's status management and reload the repository, and then try again. At most, you would have to re-deploy the EPCIS repository.

3.2.2 Creating events and query.

Now that the system is up and running, events can be created. Therefore, the first task was to decide what kind of events that was to be registered and what they meant for the system. In the end two events from the seller side and one event from the buyers side were elected. The seller register the packing and the shipping of the fish crates, while the buyer register the fish crate(s) as received. In this case, the seller side is simulated, and only packing and shipping events are registered in the repository

The reasons behind these events are connected to the work they do in the fisheries. The receiving of fish is not registered since different boats have different ways of handing off the haul. While some might come pre-packed and sorted, others might come in bulk packaging. Therefore, re-packaging or labelling of the pre-packed will be registered in the EPCIS system as packaging, and then the next major step is when the fish crates leaves the facility. Here they will be registered as shipped in the EPCIS system as an observation. A third event that was considered was a return event if the fish crates is re-used in the system, but for this thesis it was not necessary, and the assumption will be that each crate get a new SSCC number when it is packed in the fishery with their identification. However before I could generate those two types of events I had to find out how the fish crates was uniquely identified

3.2.3 Capture

Having decided what events to generate and how to convert to the XML standard, the capturing client a SSCC would be captured like this: "`urn:epc:id:sscc:3701010.1234567890`".

As we can see, it consist of 17 digits, instead of the original 18. Where 3, is the common Norwegian extension digit, 70 is the country code and 1010 is the company code.
Now the process of generating a few events can commence. In the figure below you can find an example of capturing a packaging event.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
creationDate="2008-03-06T11:42:15.016+01:00" schemaVersion="1.0"
xmlns:myNs="http://my.unique.namespace">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2015-01-26T15:02:17Z</eventTime>
        <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sscc:3701010.0000000001</epc>
        </epcList>
        <action>ADD</action>
        <bizStep>urn:fosstrak:demo:bizstep:fmcg:packaging</bizStep>
        <disposition>urn:fosstrak:demo:disp:fmcg:pending shipping</disposition>
        <readPoint>
          <id>urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,432</id>
        </readPoint>
        <bizLocation>
          <id>urn:fosstrak:demo:fmcg:ssl:0037000.00729.210</id>
        </bizLocation>
        <bizTransactionList>
          <bizTransaction type="urn:epcglobal:fmcg:btt:po">
            http://home.himolde.no/~110853/SSCC.html
          </bizTransaction>
        </bizTransactionList>
        <!-- uncomment the following for capturing an event field extension -->
        <!-- <myNs:myExtensionField>My Extension Value</myNs:myExtensionField> -->
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Figure 14 Example of a packaging event

The important fields here are <eventTime>, <eventTimeZoneOffset>, <epc>, <action>, <bizStep>, <disposition> and <bizTransaction>. On the rest I used the default demo version settings in regards to <readpoint> and <bizLocation>.
The different fields in EPCIS events are explained in chapter 2.5.1.

3.2.4 Query

From the figures below you will find an illustration of what an EPCIS query will produce. The fields that are blank are minimized, or not shown in these figures. One of the fields not shown, is the business transaction field. This field was left blank during the first event generations, but in the end product it is used to link to a website that contains information about the content in the shipping crate in the test pilot.

Query results

Event	occurred	recorded	...	...	EPCs	Action
Object	2009-08-21T15:28:17.000+02:00	2015-01-15T13:33:02.425+01:00			'urn:epc:id:sgtin:0057000.123780.7788'	ADD
Object	2009-08-21T15:28:17.000+02:00	2015-01-26T15:08:41.763+01:00			'urn:epc:id:sgtin:0057000.123780.7788'	ADD
Object	2009-08-21T15:28:17.000+02:00	2015-01-26T15:13:43.528+01:00			'urn:epc:id:sgtin:0057000.123780.7788'	ADD
Object	2015-01-26T16:02:17.000+01:00	2015-01-27T11:22:47.323+01:00			'urn:epc:id:sscc:0614141.1234567890'	ADD
Object	2015-01-26T16:02:17.000+01:00	2015-01-27T11:24:40.434+01:00			'urn:epc:id:sscc:0614141.1234567890'	ADD
Object	2015-01-26T16:02:17.000+01:00	2015-01-27T11:24:49.399+01:00			'urn:epc:id:sscc:0614141.1234567890'	ADD

Figure 15 First half of an EPCIS query

Business step	Disposition	Readpoint ID
urn:fosstrak:demo:bizstep:fmcg:production	urn:fosstrak:demo:disp:fmcg:pendingQA	urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,432
urn:fosstrak:demo:bizstep:fmcg:production	urn:fosstrak:demo:disp:fmcg:pendingQA	urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,432
urn:fosstrak:demo:bizstep:fmcg:production	urn:fosstrak:demo:disp:fmcg:pendingQA	urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,432
urn:fosstrak:demo:bizstep:fmcg:packaging	urn:fosstrak:demo:disp:fmcg:pending shipping	urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,432
urn:fosstrak:demo:bizstep:fmcg:shipped	urn:fosstrak:demo:disp:fmcg:in Transit	urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,433
urn:fosstrak:demo:bizstep:fmcg:shipped	urn:fosstrak:demo:disp:fmcg:in Transit	urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,433

Figure 16 Second half of an EPCIS query

3.3 Object Name Service

The Object Name Service (ONS) is not a system that are downloaded and then installed, but rather a Domain Name System (DNS) that are modified to fit the required specifications. As with the EPCIS system a freeware solution, bind version 9 was chosen for the creation of an ONS. Unlike the EPCIS, who I used multiple installations, only one DNS was modified to create the ONS.
The system I used to configure the ONS, was a Debian Gnu/Linux Server with Bind9 installed. To access and modify the bind configuration files, I used PuTTY.

3.3.1 Installation and configuration

When bind was installed, the configuration could begin. First task was to check if all was in working order. A test domain was entered, and then a dig command was used to check if the DNS responded (Figure 17).

```
marie@lpc200:~$ dig @localhost mssvane.org

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @localhost mssvane.org
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12099
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;mssvane.org.                   IN      A

;; ANSWER SECTION:
mssvane.org.            604800  IN      A       158.38.74.200

;; AUTHORITY SECTION:
mssvane.org.            604800  IN      NS      mssvane.org.

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Mar 19 14:31:59 2015
;; MSG SIZE  rcvd: 59
```

Figure 17 Check of DNS answer (Dig)

Having checked that the DNS server is functional, and the SSCC code was transformed, it was the possible to start entering the zones in the named.conf.local file.
In the figure below, you can see an example of how a zone is entered.

```
zone "3.1.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com." {
        type master;
        file "/etc/bind/zones/3.1.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com-zones";
        allow-query {any; };
        allow-update {none;};
};
```

Figure 18 Named.conf.local zone configuration

From the example you see that the named.config.local file points to the zone record. The zone record contains the information about the expected prefix of the domain, length and the address to the EPCIS server or a service record.
An example of a zone record can be found **Error! Reference source not found.**.

```
$TTL 86400 ; 1 day
@ IN SOA ns.3.1.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com. example.com. (
                                    2008010801 ; serial
                                    21600 ; refresh (6 hours)
                                     1800 ; retry (30 minutes)
                                  1209600 ; expire (2 weeks)
                                    86400 ; minimum (1 day)
)
                                         IN NS ns.3.1.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com.
ns.3.1.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com. IN A 127.0.0.1
$ORIGIN 3.1.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.7.sscc.gs1.id.onsepc.com.
0.1 IN NAPTR 0 0 "t" "http://www.gs1.org/ons/prefix" "!^.*$!70,6!" .
0.1 IN NAPTR 0 0 "u" "http://www.gs1.org/ons/epcis" "!^.*$!http://158.38.74.201:8080/epcis-repository-0.5.0/query!" .
0.2 IN NAPTR 0 0 "t" "http://www.gs1.org/ons/prefix" "!^.*$!70,6!" .
0.2 IN NAPTR 0 0 "u" "http://www.gs1.org/ons/epcis" "!^.*$!http://158.38.74.201:8080/epcis-repository-0.5.0/query!" .
0.3 IN NAPTR 0 0 "t" "http://www.gs1.org/ons/prefix" "!^.*$!70,6!" .
0.3 IN NAPTR 0 0 "u" "http://www.gs1.org/ons/epcis" "!^.*$!http://158.38.74.201:8080/epcis-repository-0.5.0/query!" .
1.1 IN NAPTR 0 0 "t" "http://www.gs1.org/ons/prefix" "!^.*$!70,6!" .
1.1 IN NAPTR 0 0 "u" "http://www.gs1.org/ons/epcis" "!^.*$!http://158.38.74.201:8080/epcis-repository-0.5.0/query!" .
3.4 IN NAPTR 0 0 "t" "http://www.gs1.org/ons/prefix" "!^.*$!70,6!" .
3.4 IN NAPTR 0 0 "u" "http://www.gs1.org/ons/epcis" "!^.*$!http://158.38.74.201:8080/epcis-repository-0.5.0/query!" .
3.5 IN NAPTR 0 0 "t" "http://www.gs1.org/ons/prefix" "!^.*$!70,6!" .
3.5 IN NAPTR 0 0 "u" "http://www.gs1.org/ons/epcis" "!^.*$!http://158.38.74.201:8080/epcis-repository-0.5.0/query!" .
```

Figure 19 Zone record

This is also called a NAPTR record, and was previously explained in the GS1 standards in chapter 2.6.2.

3.4 Example 1 - Webpage

To enable the EPCIS repository to be queried with the use of the ONS, a search page was created.

3.4.1 Validation

When the user enter the SSCC code, the first thing that needs to be done is a validation of the users input. This validation step checks that only numbers are entered into the search field, as dictated by the SSCC standard, and the length of numbers entered. An appropriate error message is given if the validation steps fails.

Below you will see the first validation test. This test checks if what is entered or chosen from the pulldown menu, is not null:

```
if (isset($_POST["sscc"])) {
```

Then it is the initial transforming of the user input:

```
//get raw input
$field= $_POST["sscc"];
//transform into a fully qualified domain name (FQDN)
$id=
substr($field,0,1).".".chunk_split(strrev(substr($field,1,16
)),1,'.')."sscc.gs1.id.onsepc.com";
```

The second validation is a new if-test, to check length and if it consists of just digits:

```
if (strlen($field)==17) {
          if(ctype_digit($field)){
               $ok= true;
          } else {
               echo "<p>Warning: Invalid SSCC - should be
digits.</p>";
          }
     } else {
     echo "<p>Warning: Invalid SSCC, wrong length - should
be 17 digits</p>";
}
}
```

3.4.2 Dig after NAPTR record in local ONS

When the input validated and deemed satisfactory, we can ask ONS about the NAPTR record for that specific SSCC code.

The if-test for NAPTER dig is illustrated below:

```
if ($ok) {
$host = '0.1.'.$id;
$cmd = sprintf('dig @158.38.74.200 %s NAPTR
+short',escapeshellarg($host));
     exec($cmd, $dns_records);
```

Here I check out each record to filter out unwanted responses:

```
     foreach ($dns_records as $dns_record) {
          if (!strpos($dns_record, 'prefix')) {
          $data= explode(" ",$dns_record);
          $addr= substr($data[4],7,-2);
          $found= true;
          }
}
if (!$found) {
          echo "<p>Warning: ONS record not found for this
SSCC.</p>";
}
```

As can be read by the code above, if there is not found a hit in the NAPTR record, a warning message will be given to alert the user that no record is found for that SSCC.

A few thing that is worth noticing is that this search portal is hardcoded. The fields in question is host and address. The host has a fixed prefix, in this case “`0.1.`”, while the address is hardcoded to a local ONS on IP: `158.38.74.200`. This demo is also for only SSCC code searches, and if this search portal should function like a general search these settings would have to be altered.
In the case of this demo, this is not a problem since it is querying for something particular.

3.4.3 Setup of querying in the EPCIS

If the user has managed to enter a SSCC that don’t prompt an error message of the two previous steps, the site can then move on to loading and querying the EPCIS.
First, we add some lines to get the EPCIS address and end up with the right query format of the SSCC for the EPCIS repository.

```
$wsdl = $addr.'?wsdl';
$sscc = substr($field,0,7)."."substr($field,7);
```

Before we move on to define the poll query and the calling the poll function.

```
// Defining the (poll) query
$params = array(
'queryName' => 'SimpleEventQuery',
     'params' => array(
               array(
                    'name' => 'eventType',
                    'value' => 'ObjectEvent',
               ),
               array(
                    'name' => 'MATCH_epc',
                    'value' => "urn:epc:id:sscc:$sscc",
               ),
     ),
);

// Calling the poll function - one of the functions
described in the WSDL.

$result = $client->poll($params);
```

After this is done, we can start looping through the results to find the field and information needed, before echoing the result out to the user.

```
// Handling the result: Looping over returned events to find
the ADD event that contains the URL.

foreach ($result->resultsBody->EventList->ObjectEvent as
$event) {
     if ($event->action == 'ADD') {
$url = event->bizTransactionList->bizTransaction->_;
          break;
     }
}
echo "<h2>Result of query</h2><p>Read more about
<a href='$url'>$sscc</a>";
}
```

From the code above, we see that this poll is querying something particular, and collecting a URL from the EPCIS server. This particular URL is added in all my “ADD” events in the bizTransaction field in the EPCIS. Because of this, the search portal is not helpful for any other types of queries.

3.5 Example 2 - Webpage

This demonstrator should be able to be query both SSCC and GTIN codes, and is an altercation from how the first demonstrator works. Where the first one looked for a special URL in bizTransaction in the ADD event, this demonstrator is counting the number of events on the given EPC code.

3.5.1 Alterations from demo 1

Since this demonstrator has the possibility to take more than just one type of code, even though the focus on this thesis is SSCC, I altered the validation step slightly. You can see the changes in the code below.

```
$ctype= $_POST["type"];
$field= $_POST["code"];
$id=substr($field,0,1).".".chunk_split(strrev(substr($field,
1,strlen($field)-1)),1,'.').$ctype.".gs1.id.onsepc.com";

if (($ctype=="sscc" AND strlen($field)=="17") OR
($ctype=="gtin" AND strlen($field)=="13")){
     $ok=true;
} else{
     echo "<p>Warning: You entered $field, it doesn't match
the ".strtoupper($ctype)." EPC standard.</p>";
}
```

From this, we see that instead of just checking if it is a SSCC and consist of 17 digits, we add the check for GTIN with 13 digits also.

Another thing worth noticing is the `$ctype`, that the user chooses when he is querying, and this alters how the `$id` variable is built up.
I kept the rest as it was, with small changes in regards to the fields that are depended on the type of EPC code.

I also altered the polling part in regards to value, to reflect the chosen code type.

```
// Defining the (poll) query
     $params = array(
          'queryName' => 'SimpleEventQuery',
          'params' => array(
               array(
                    'name' => 'eventType',
                    'value' => 'ObjectEvent',
               ),
```

```
                array(
                     'name' => 'MATCH_epc',
                     'value' => "urn:epc:id:$ctype:$code",
                ),
           ),
     );
```

In the result part, I altered this in regards to its output statement.

```
echo "<h3>Result of query</h3> EPCIS name:
urn:epc:id:$ctype:$code <br> EPCIS address: $addr <br> No.
of events:
".count($result->resultsBody->EventList->ObjectEvent);
```

Instead of looking for an “ADD” event, with the bizTransaction field, we instead print out the converted EPCIS name and the address to the EPCIS server, before we count the number of events that are on that EPC code.

4.0 System demonstration

In the previous chapters, I described the background as well as making individual descriptions for each of the systems I have setup. In this chapter, we will see how the different queries works in the demonstrators, and the answers they will produce.

First, I will illustrate how a typical ONS query is done (see the figure below), before I explain demonstrator 1 and 2.

4.1 Typical ONS query

First, you scan or type the SSCC into the application (see the figure below). Depending on how you enter the SSCC, the application strips extra digits and then transform the SSCC (see example 4.1).

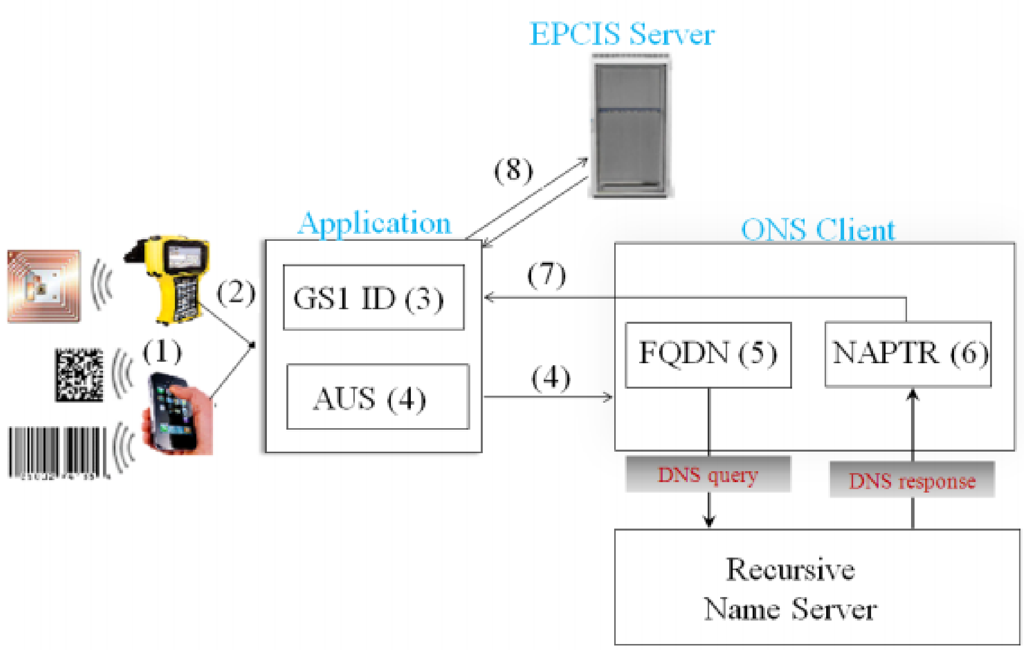

Figure 20 Typical ONS query

Source: http://www.gs1.org/sites/default/files/docs/epc/ons_2_0_1-standard-20130131.pdf P.11

After the transformation of the SSCC into a FQDN, it is ready for querying in the ONS client. It then asks for a hit in the NAPTR record. If it is a hit, the application will then get the address to the EPCIS server or a service URI from the ONS client.

Example 4.1: Transforming
SSCC raw: (00)370101010000000009
GS1 SSCC: 370101010000000009
AUS: en|no|sscc|370101010000000009
FQDN: 3.0.0.0.0.0.0.0.0.0.0.1.0.1.0.1.0.7.sscc.gs1.id.onsepc.com
You can read about the full transformation process in chapter 2.4.3.

Depending on how the application is set up, it can use the information obtain from the query for further querying of the EPCIS, or give the service link to the user.

4.2 Example 1 – Specialized Query

You can see how the interface between the SSCC entry and the repository looks below.

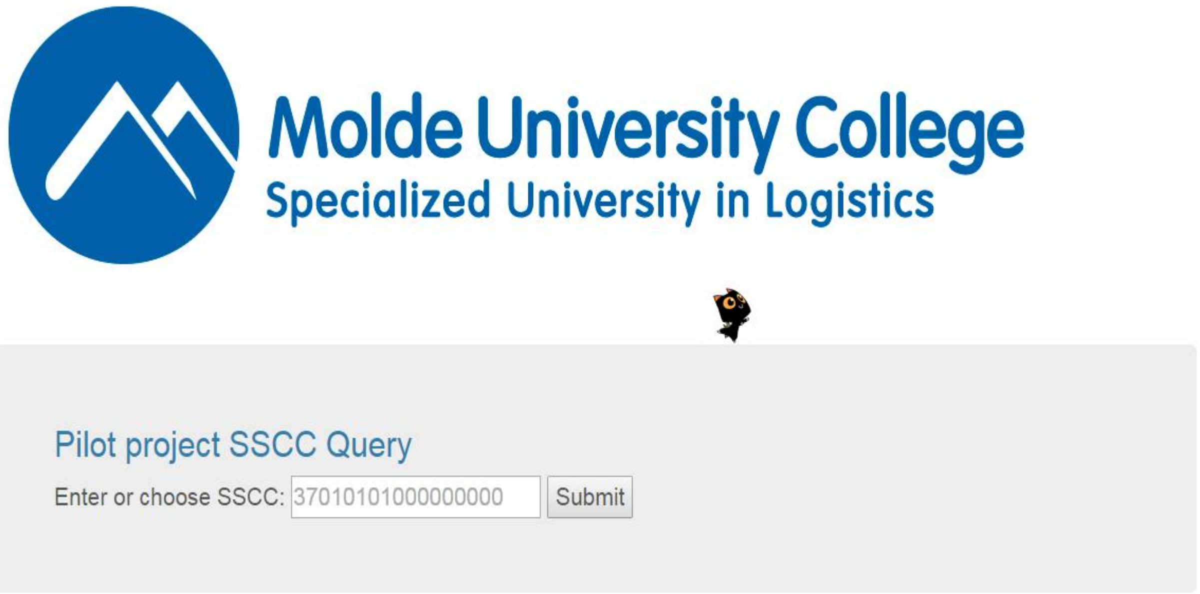

Figure 21 Demo 1

From this picture we see that the only possible search is SSCC codes. In the live version, I have a pull down list with 10 pre-approved SSCC test codes. After a SSCC is entered, validated and you have received a hit, you will get a page that looks like the figure below.

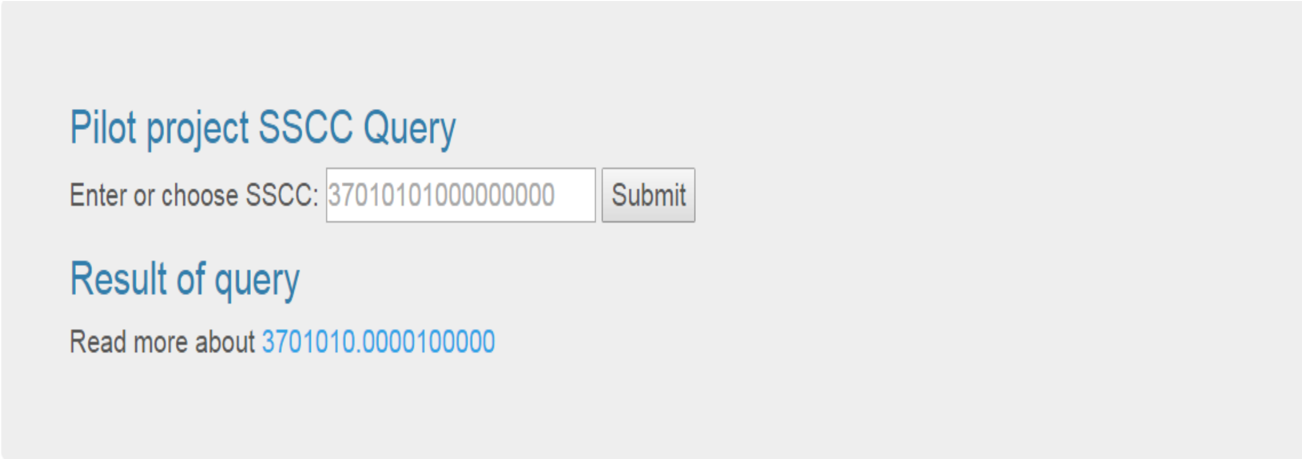

Figure 22 Demo 1 query example 1

By clicking on the light blue SSCC code, you will end up on a page with extended information. In this thesis, the extended information page is a static page with a few dynamic elements. I did this to show the possibilities of extended information that you can share in the supply chain.

From the figure below we can see the information field where general information is given. In the live version of this site, it is also a map on the bottom showing the catching areas close to the landing area.

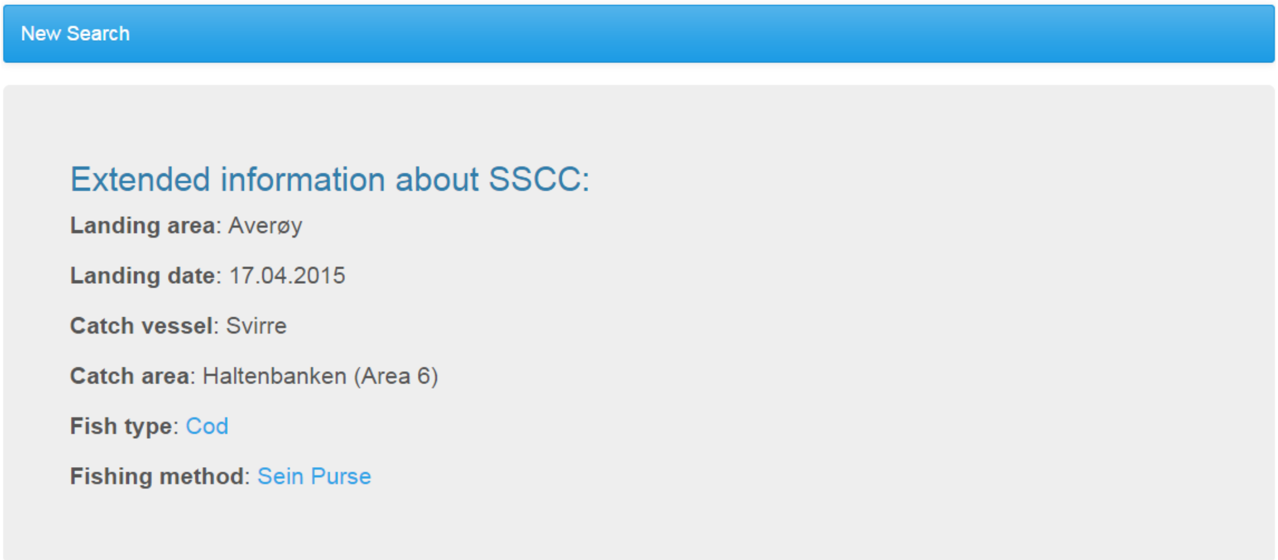

Figure 23 General information

As earlier suggested, this is just a sample of information that you can share, this is depended on the willingness of the "owner" to do this. If the contract note is the basis of information, the directorate of fisheries sets guidelines on what information that is required for sale in Norway (Fiskeridirektoratet 2015).

4.3 Example 2 – General Query

On this page, I wanted to have an option for expansion. I did this by adding a radio button to indicate what type of GS1 code you enter. I also took away the pull down list, with the pre-approved SSCC test codes.

In the figure below, you can see the design of the second search page. The default type choice is SSCC, and is the used GS1 standard in this thesis.

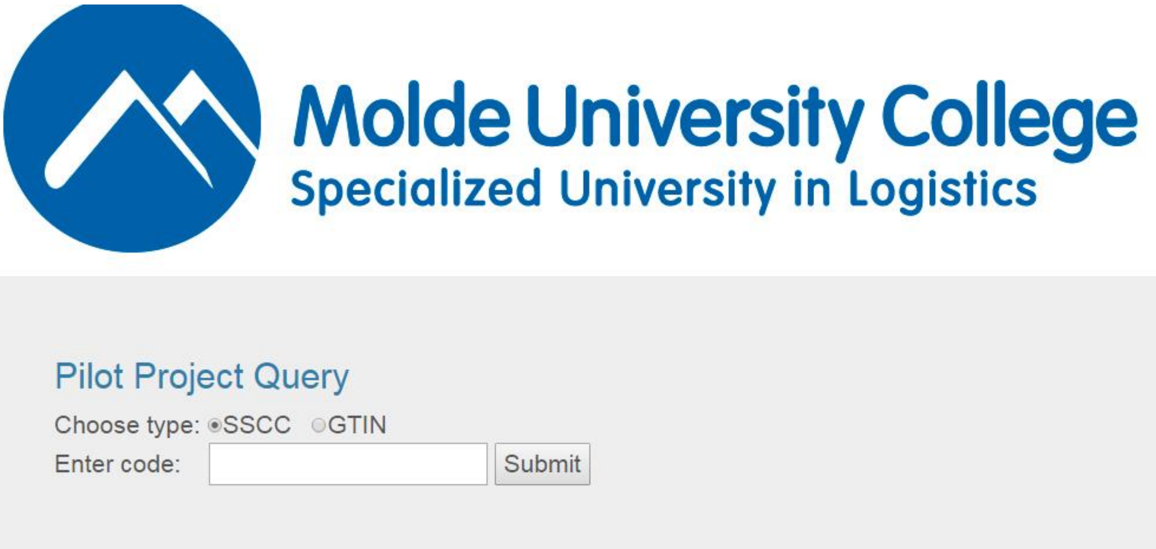

Figure 24 Demo 2

In the figure below, you can see how a query of one of the ten test codes looks.

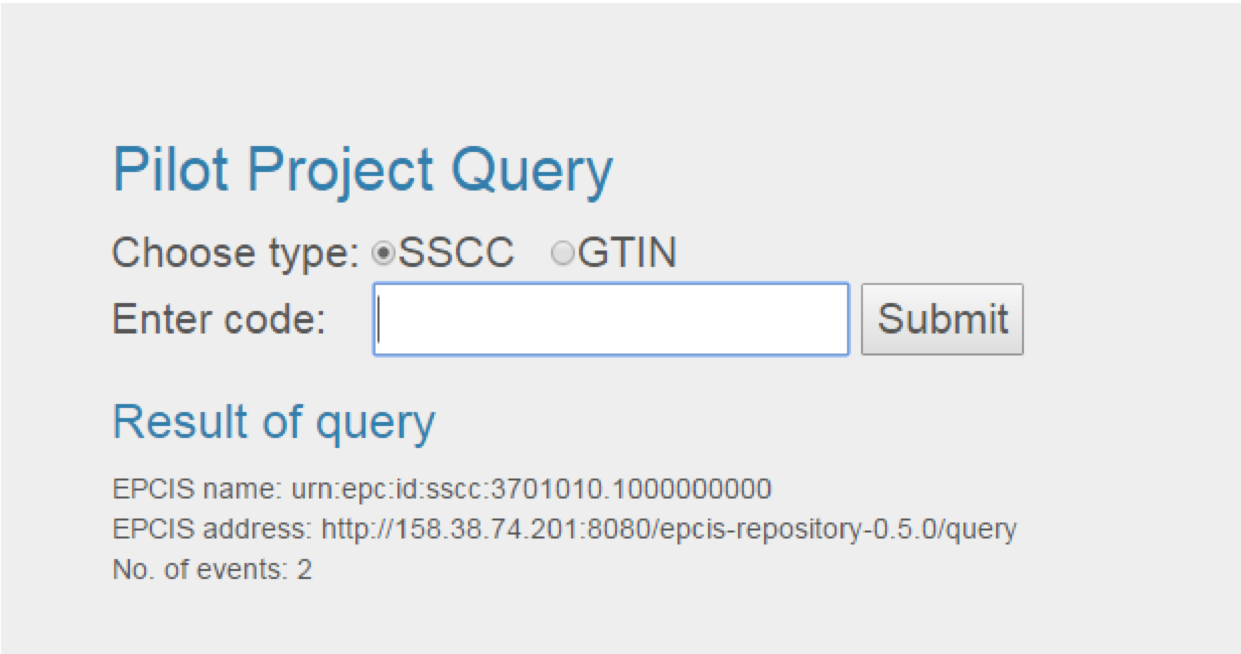

Figure 25 Demo 2 query example 1

You will notice that both the EPC and EPCIS address is given, this to enable the user to query further in the programs that fosstrak supplies, and to distinguish the EPCIS servers used.

Figure 26 Demo 2 query example 2

Figure 25 and Figure 26, are examples of queries done on two different SSCC set, who both leads to different EPCIS servers.

5.0 Result and discussion

In this chapter, I will look at the result of the demonstrator and discuss it against my research questions. From the start, I divided this thesis into two parts, one practical with the creation of the demonstrator, and the other theoretical. Both these parts will be brought together here, where I look at the demonstrators and the value these ideas can add.

5.1 Demonstrator

The idea behind this demonstrator was first to create a local ONS and show how you could use that to give extended information about an object. With that accomplished, the next step was to connect it to other ONS in the network, a so-called federated ONS.

One of the Oliot's project was an ONS, and while creating the demonstrators I could copy the setup from their files (Oliot 2014b). I was able to do this because they are based on the GS1 standards and they used the same freeware I used. Having an actual ONS setup to look at helped me immensely.

5.1.1 Example 1 and 2 – Local ONS

In these examples I meant to connect an EPCIS and an ONS together. As noted earlier, this demo is pointing directly to a local ONS, and have a very specific purpose. below you can see an illustration on how the local ONS is set up.

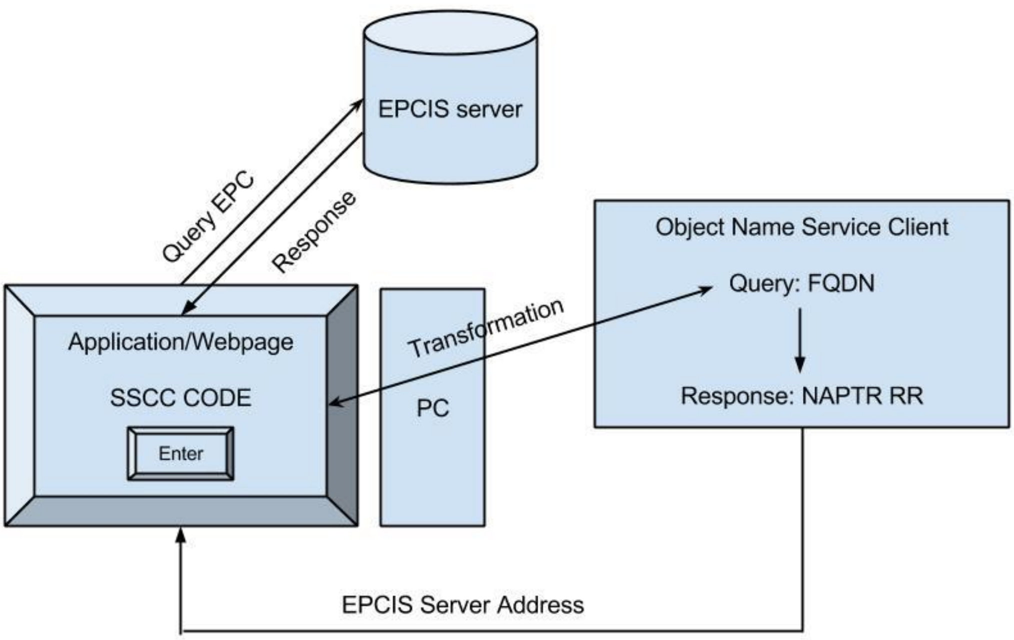

Figure 27 Example of local ONS setup

Example 1:
In the EPCIS server, the field that is queried (bizTransaction) contains an URL for extended information, which in this case require a specialized search. This was done to give the customers the chance to collect extended information about the fish-crate they have received. Having successfully configured a local ONS, I could move on to example 2.

Example 2:
Small changes were made to make it more versatile. I also wanted to show that you could connect multiple EPCIS servers to a local ONS. You still have the possibility to add extended information in this version, but I opted not to make a database. Fisheries can register the contract notes and other information into a database and use certain event data from the EPCIS repository to do lookups in that information database.

5.1.2 Example 3 – Connected F-ONS

In this example I am meant to connect or communicate between ONS (see the figure below). GS1 France was contacted on the background for this delegation, because they are the only ones with an operational ONS registered.

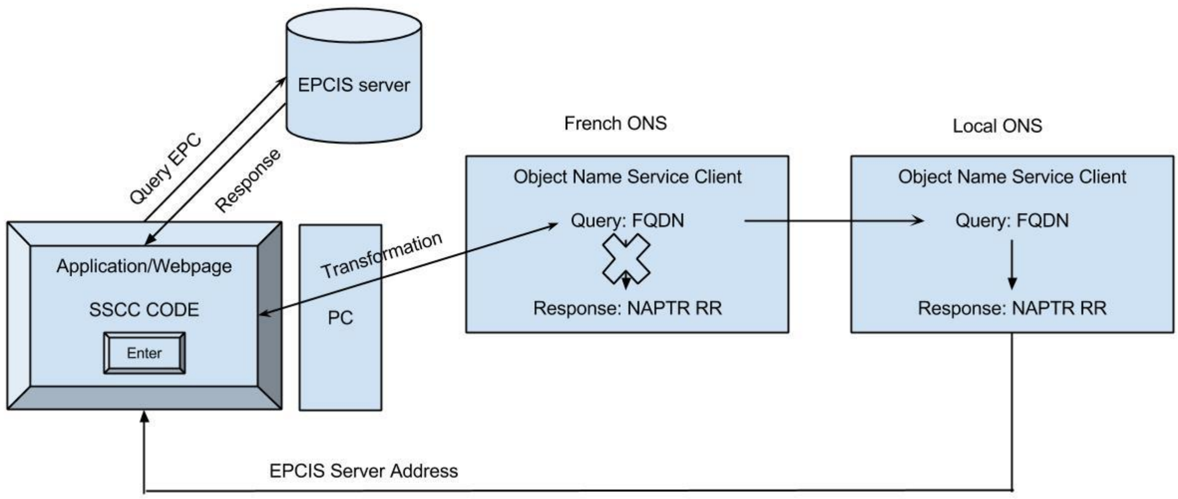

Figure 28 Example of ONS communication

The whole idea behind F-ONS was having multiple root ONS nodes that delegated down to region or country nodes, and then to local nodes. This will work in approximately same way as the DNS. This usually means that on the highest level of ONS, it is dealing with country prefixes, which further delegates to the country ONS, who has the company prefixes from that country. The local ONS have the unique object codes, connected to the company prefixes.

The French ONS is on country level, and they have delegations to the company prefixes. They agreed to add a NAPTR record on my test code company prefix in the GS1 France ONS, but when attempting to contact that record I never got a response back from the French ONS. In that regard, I was unable to verify that a NAPTR record exist.

To delegate queries from my local ONS to the French ONS I made a DNAME record. This record would delegate all the queries with ending "`1.0.7.sscc.gs1.id.onsepc.com`" it did not find in my NAPTR, to ending "`1.0.7.sscc.gs1.id.preprod.onsepc.fr`".
However, I never got a working SSCC to test this concept from GS1 France, since they operate on a prefix level and not full object addresses.

5.2 Value adding initiatives

The value aspect of IoT enabled technology is well discussed in multiple articles, a few of them I will highlight in this discussion. Value creation is a broad topic, so to explore what is of greatest importance to the supply chain and the society. I divided the value adding initiatives into the three categories discussed below.

5.2.1 Supply wide sharing

In the article "*Industry-wide supply chain information integration: The lack of management and disjoint economic responsibility*" by Henningsson and Hedman (2010) they try to address information integration across organizational boarders, and are strong advocates of information sharing. They say that what is important is getting the right information to the right time, to create a competitive advantage. This opens up for the use of the IoT aspect that are proposed in this thesis. Where the objects are able to be scanned and the extended information about that object is made shareable.

Henningsson and Hedman illustrated their paper with four cases in the food industry in Sweden. They looked at the flow of milk, pork, sugar and peas. Findings in all the four areas were quite interesting. First, is that end customers receive limited information, even though a few of these flows have the possibility to offer more information. Another fact was that the industry-wide supply chain was shaped like a sandglass, so the few actors in the middle dictated information integration. This meant that unless they saw a clear advantage for them, or government demanded it, information sharing would not be set into action.

From a fishing industry point of view, we have a similar picture, but instead of a few key point actors, there are many actors all over. This point us to the importance of having a standardized IoT system, that is easy to put up and has a global reach. GS1 have really come along way when it comes to standardization, and with the setup in the demonstrators all the steps in the chain have the possibility to look at the extended information. If we alter it to a traceability network and use the same concept as fTrace have applied on their system (see figure below). You could in theory add your information to the database, to keep what is in the transportation units up to date, to both enhance and improve the information flow in the seafood supply chain.

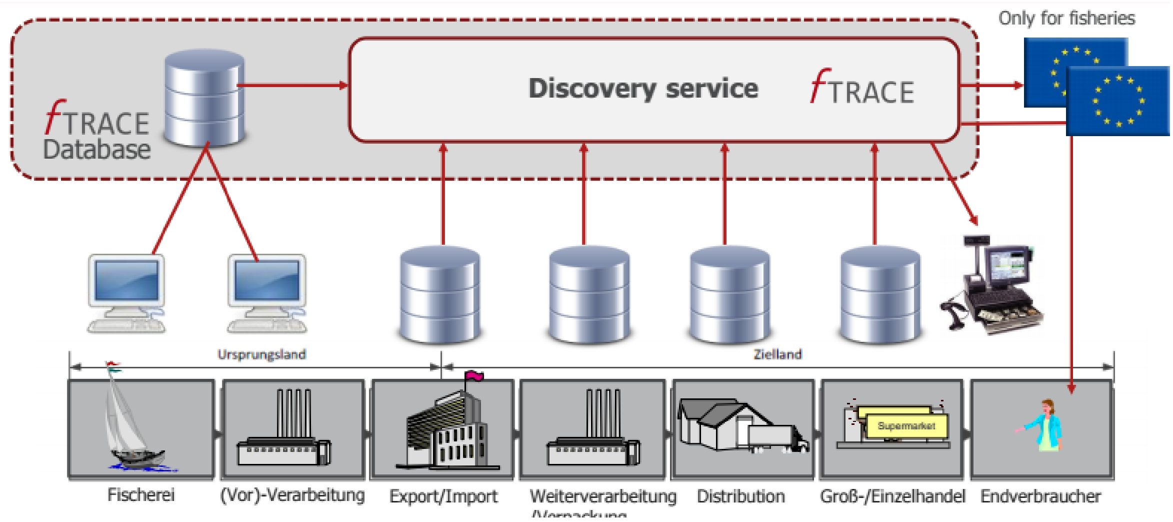

Figure 29 IT infrastructure fTrace

5.2.2 Branding and sustainability

If Norway intends to depend on the fishing industry, with the declining of the oil & gas industry, our fish products needs to be able to be differentiated from the rest of the world's fish products. This means that strong "branding" and sustainability is important for Norway as a nation (Kvalvik, Noestvold, and Young 2014).

Norway already have a strong brand in seafood, Norge – Seafood from Norway (Norwegian Seafood Council 2014). This means with IoT enabled technology we have the chance to combine the strong brand name, with the traceability and extended information possibilities this technology provides.

If we look back to example 1 system demonstration, I mentioned the possibility to show a map. This map can contain information or visualization of where the fish came from and what fishery it was shipped from. The traceability project eTrace tested this concept (see below).

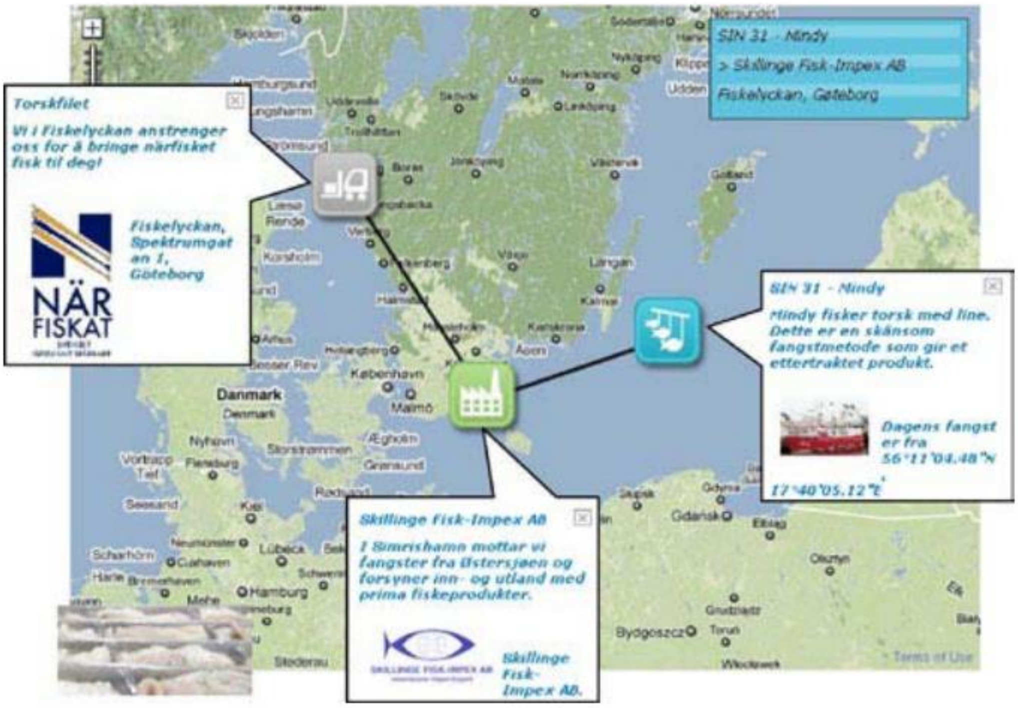

Figure 30 Example of eTrace traceability

Source: http://www.tracefood.org/images/9/9d/Swedish_pilot_presentation_at_end_meeting_4-2011.pdf p.10

A testimonial after the eTrace project was shop owner Peter Kallstrom, he said: "*Traceability have been a driver for increase sales. Next to the cod we posted a map showing where the fish was caught and processed. The map told the history that the consumers have been waiting to hear, namely that the fish is local…*" They found an increase in sales with displaying extended information about the fish. From the example above, he went from a few kilos a day to more than 150 kilos over 4 days (Nicklas Hild 2013, Page 6).

Norway have also opted for the Marine Stewardship Council (MSC) to third part check our sustainability (Kvalvik, Noestvold, and Young 2014), this to check that we are not exhausting our natural resources.

5.2.3 Food health and environment

It is not always noticed if you go "green", and consumers cannot always know what sustainable food is. Here IoT enabled technology get a great chance to teach a consumer, what is done with the product, how it is fished and maybe even tell what consequences this type of fishing method have. Since it is extended information and not only what fits on the food packaging or label.

Ola Bø (2011) asks if “*Tracking systems can let consumers locate the most sustainability harvested products among the offering on the fish counter*”. Where he discovers that few manufactures offers the method of fishing on their packages, and those who do, do not really give much knowledge unless you have prior experience in how fish are best harvested to preserve the environment.

Donnelly and Olsen (2012) did a case-study in the Norwegian white fishing sector, where they looked at traceability and said that it can be used as a tool for increasing market access and share. Traceability means that each step record what is done with the product and that you can find information about the procedures later. They divide the traceability in three areas, internal, along the supply chain and industry-wide. Often companies have a good internal system, but different data standards hinders supply/industry-wide sharing.

In the case study, at CodTrawl Inc (CTI), research showed that they had a good system that gave each trip a trip number, that is connected to all the fish in the haul and the haul number. It was further divided into trade units with unique ID that was traceable to the trip number. Therefore, they had complete traceability on the boat. However, this traceability stopped there, it was no link between boat and harbour. Here IoT enabled technology and the access to CTI's trade unit Id's would have been possible for further utilization of the traceability data, but it was not considered practical or economical.

In the “*Investigating Federated Object Naming Service as Directory Service for the Internet of Things*” they addressed part of the food health issues also, this in regards to recall of products (Fredriksson, Pehrson, and Sandmark 2011). With the use of F-ONS and Trace & Track procedure, they would identify and locate all products that was deemed unfit for consumption. This would prevent companies from having to recall full product groups, where only part of the products where affected. This will both have an effect on the environment, less food is thrown away, and economically for the supply chain. This can also have the possibility to reflect back to the brand name, when you can easily remove bad batches with the Trace and Track solution.

6.0 Limitations and Future Work

In this chapter, I will look at the limitations and the future work that can be done in this field. I will start with explaining the limitations this thesis have, before I move on to suggest the future work that can be done in regards to this work.

6.1 Limitations

As with each project, there are certain limitations that will arise. I have divided these challenges or limitations into supply chain limitations, and limitations in the creation of the demonstrator.

6.1.1 Supply chain

As previously mentioned, the supply chain in the fishing industry is not easy to navigate. Often the fish and the information about the fish separate early on, since fish brokers are buying and selling the fish multiple times.

Another challenge is that firms that buy the fish might break the fish crate and then break the SSCC chain. When or if they switch transportation labels or fish crate. This makes it a perquisite that the SSCC is not changed in the SC.

6.1.2 Demonstrator

In regards to the demonstrator, there are certain limitations to take in consideration.

The first one is that I have disregarded the safety aspect, when creating and discussing these demonstrators. I know this is a huge topic and is widely discussed, in regards to data safety and stealing. This leads us to notice the simplicity of a demonstrator. I have tried to simplify my designs, and the way these demonstrators works.

Furthermore, I am only using a small part of the GS1 standards. This means that I am mostly using SSCC codes, and the standards that are connected to the use of this. On top of this, all my SSCC codes are test codes, e.g. they are not registered in GS1 as a valid code.

Lastly, there are different methods to get tag data. This meant that the method of collecting tag data is not important. During this thesis, I have chosen to either enter manually or choose a pre-approved code from a pull down menu.

6.2 Future work

With this thesis having an explorative methodology and just on the fringe of the IoT applications possibilities it opens up for further scrutinizing and exploration of the “fuzziness” of the IoT concept.
In this chapter I will look at the future work that can be done in the area of IoT application.

To get a real picture of how this would work in the real world, I would recommend doing this again, but this time have real GS1 codes and connect to a real company or another university that are doing research on this.
There is also a possibility to make a national ONS that handles the company prefixes, instead of the product codes, with delegating down to local ONS. Like the one that is operating in France.

In regards to value creation initiatives a case study of a French supply chain who has implemented ONS, or implement an ONS in a Norwegian supply chain and look at the values this add if any.

7.0 Conclusion

My area of study was an exploration of the IoT concept, while using the digitalization of the seafood industry as an example. I decided to focus on the thing-oriented perspective of the IoT concept, in the wild-caught fish segment.

My first research task was to create an IoT platform and my research question was finding out what technology components that are needed to build an IoT platform for seafood information. By starting with an explorative methodology, I first researched what an ONS is, the standards connected to it and what components was needed to build the demonstrators.
I installed the EPCIS repository, and registered events based on my test codes. After this was accomplished, I moved on to the ONS configuration. To connect multiple ONS together DNAME records for redirecting of entire trees was implemented, but not verified. Now that the local ONS was created and the NAPTR RR created to point to the EPCIS server, I needed an application to bring this all together. The solution for this was a webpage, where you query the SSCC test code and communicate with both the ONS and the query function in the EPCIS setup. For the seafood information part there are two different possibilities, I chose to make a semi-dynamic website that was added to EPCIS events. In the real implementation however, the brand owner or the fishery would have a separate database with information about the object(s), and then use different fields in the EPCIS to search up extended information about the specific object(s).

The second research task was connected to how the platform could be used to add value for the actors in the supply chain. This was a theoretical research, since it was not implemented in a real supply chain.
My findings from my study of the three categories, supply wide sharing, branding & sustainability and food health & environment, are as follows:

- An increased information to the end customers increased the sales in the eTrace project.
- Unless actors saw a clear advantage for them, or government demanded it, information sharing would not be set into action.
- With the use of F-ONS and Trace & Track procedures, identification and localization of all products that are unfit for consumption becomes easier. This prevents recall of full product groups, where only parts are affected. This show a positive effect on the environment since less food is thrown away. This can also reflect back to the brand name, since you can remove bad batches.
- If we compare ONS with fTrace, we see that in the ONS solution the brand owners have full control over his extended information, while in the fTrace they have standardized parts of the information.

Glossary

AI
Application Identifier - Standard for encoding different identification types into a barcode or DataMatrix.

Auto-ID
“Automatic Identification” – a set of technologies used to identify anything, anywhere, automatically.

BIND
The Berkeley Internet Name Domain is the most widely used DNS implementation on the Internet.

CMT
Common Mapping Table - Mapping table linking GS1 prefixes with which ONS Peer Root it corresponds.

DBMS
Database Management System - A set of programs that control the creation, maintenance and use of a database.

Domain name
A hierarchical, ’dot’ (.) separated namespace used to identify hosts on the Internet.

DataMatrix
A two-dimensional barcode consisting of black and white ”cells” or modules arranged in either a square or rectangular pattern. The encoded information can be text or raw data.

DNAME
A DNAME record is a DNS zone file entry that map a subtree of the DNS name space to another domain.

DNS
Domain Name Service - An infrastructure level Internet service used to discover information about a domain name.

EAN
European Article Number - Barcode standards defined by GS1 used worldwide for identifying products.

EPC
Electronic Product Code - An abstract namespace made up of an EPC Manager Number, an Object Class code, and a Serial Number, or a subset thereof.

EPCglobal
A joint venture between GS1 and GS1 US. It is an organization set up to achieve worldwide adoption and standardization of Electronic Product Code (EPC) technology.

EPCIS
Electronic Product Code Information Service - A series of EPC Network specific standards defining various methods of data exchange and metadata lookup.

FQDN
A fully qualified domain name (FQDN), sometimes also referred as an absolute domain name, is a domain name that specifies its exact location in the tree hierarchy of the DNS. It specifies all domain levels, including the top-level domain and the root domain. A fully qualified domain name is distinguished by its unambiguity; it can only be interpreted one way.

F-ONS
Federated Object Naming System - Multiple ONS peer roots collaborating at the same hierarchical level to resolve the response for queries.

GS1
Global Standards 1 - An international nonprofit association dedicated to the development and implementation of global standards.

IoT
Internet of Things - A conceptual name describing the area of Internet-connected objects.

NAPTR
Naming Authority PoinTeR - A DNS record type (35) that contains information about a specific delegation point within some other namespace using regular expressions.

ONS
Object Name Service - A resolution system based on DNS, used for discovering authoritative information about an EPC.

ONS peer-root
An ONS node at root level in a Federated ONS.

RFID
Radio Frequency IDentification - An object identification technique that is using radio waves to transfer data.

Regular Expression
A standard language for pattern matching within a string of characters and for composing new strings based on matched subcomponents of the original string (i.e. a search and replace function)

SC
Supply chain - a system of organizations, people, activities, information, and resources involved in moving a product or service from supplier to customer.

Serial Number
A number that identifies a particular instance of an object class.

SSCC
Serial Shipping Container Code - used to identify traded objects and services.

TDT
Tag Data Translation - Translation between different types of tags. EPC to DNS etc.

URI
Uniform Resource Identifier - The superclass of all identifiers that follow the ’scheme:scheme-specific-string’ convention.

URL
Uniform Resource Locator - A URI that identifies a resource via a representation of its primary access mechanism (e.g., its network location), rather than identifying the resource by name or by some other attribute(s) of that resource.

XML
eXtensible Markup Language - A set of rules for encoding documents in machine-readable form.

8.0 References

AFNIC. *The Internet of Things - GS1 France & Afnic major contributors to the ONS 2.0* 2013 [cited 27.04.2015. Available from http://www.afnic.fr/en/about-afnic/news/general-news/6703/show/the-internet-of-things-gs1-france-afnic-major-contributors-to-the-ons-2-0.html.

Apache Tomcat. *Apache Tomcat,* . Available from http://tomcat.apache.org/index.html.

Atzori, Luigi, Antonio Iera, and Giacomo Morabito. 2010. "The Internet of Things: A survey." *Computer Networks* no. 54 (15):2787-2805. doi: http://dx.doi.org/10.1016/j.comnet.2010.05.010.

Balakrichenan, Sandoche, Antonio Kin-Foo, and Mohsen Souissi. 2010. Federated ONS Architecture for the Internet of Things–A Functional Evaluation. Paper read at Internet of Things 2010 Conference, Tokyo.

Bø, Ola. 2011. The potential of IOPTSs in a sustainable supply chain - observation from the wild seafood supply network. In *EUROMA*. Cambridge, UK: EUROMA.

Cambridge Auto-ID Lab. *History of Auto-ID Labs* 2015 [cited 22.04.15. Available from http://www.autoidlabs.org.uk/.

Crawford, Matt. 1999. "Non-Terminal DNS Name Redirection."

Daniel, Ron, and Michael Mealling. 2000. "The Naming Authority Pointer (NAPTR) DNS Resource Record."

Directorate of fisheries. 2010. Marine life – our common responsibility. edited by Directorate of fisheries. http://www.fiskeridir.no/english/publications: Directorate of fisheries.

Donnelly, Kathryn A. M., and Petter Olsen. 2012. "Catch to landing traceability and the effects of implementation – A case study from the Norwegian white fish sector." *Food Control* no. 27 (1):228-233. doi: http://dx.doi.org/10.1016/j.foodcont.2012.03.021.

eTrace. *International:SAFEFOODERA eTrace* 2013 [cited 23.04.2015. Available from http://www.tracefood.org/index.php/International:SAFEFOODERA_eTrace.

EU. 2002. *Regulation (EC) No 178/2002 of the European Parliament and of the Council of 28 January 2002 laying down the general principles and requirements of food law, establishing the European Food Safety Authority and laying down procedures in matters of food safety*. http://eur-lex.europa.eu/ (02.10.2014): EU, ed.

Fiskeridirektoratet. *Elektronisk rapportering og fangstdagbok* 2015 [cited 18.05.15. Available from http://www.fiskeridir.no/fiske-og-fangst/elektronisk-rapportering.

Fosstrak. *EpcisUserGuide* 2012. Available from https://code.google.com/p/fosstrak/wiki/EpcisUserGuide.

Fredriksson, Marcus, Andreas Pehrson, and Dennis Sandmark. 2011. "Investigating Federated Object Naming Service as Directory Service for the Internet of Things."

GS1. *Object Name Service (ONS)* 2013 [cited 27.04.2015. Available from http://www.gs1.org/sites/default/files/docs/epc/ons_2_0_1-standard-20130131.pdf.

GS1. *Core Business Vocabulary (CBV)* 2014a [cited 27.04.2015. Available from http://www.gs1.org/sites/default/files/docs/epc/cbv_1_1-standard-20140520.pdf.

GS1. *EPC Information Services (EPCIS)* 2014b [cited 27.04.2015. Available from http://www.gs1.org/sites/default/files/docs/epc/epcis_1_1-standard-20140520.pdf.

GS1. *fTRACE: A GS1 technology based traceability solution*. GS1 fTrace 2014c [cited 04.10.2014. Available from http://static.ftrace.de/pdfs/fishpilot/One-Pager_fTRACE_EPCIS_en.pdf.

GS1. *About GS1* 2015 [cited 20.05.2015. Available from http://www.gs1.com/about.

Henningsson, Stefan, and Jonas Hedman. 2010. "Industry-wide supply chain information integration: The lack of management and disjoint economic responsibility." *International Journal of Information Systems and Supply Chain Management (IJISSCM)* no. 3 (1):1-20.

Holmström, Jan, Mikko Ketokivi, and Ari-Pekka Hameri. 2009. "Bridging practice and theory: a design science approach." *Decision Sciences* no. 40 (1):65-87.

Innovasjon Norge. 2014. Sporbarhet fra fangst til tallerken.

Kvalvik, Ingrid, Bjoerg H. Noestvold, and James A. Young. 2014. "National or supranational fisheries sustainability certification schemes? A critical analysis of Norwegian and Icelandic responses." *Marine Policy* no. 46 (0):137-142. doi: http://dx.doi.org/10.1016/j.marpol.2014.01.015.

Li, Dong, Xiaojun Wang, Hing Kai Chan, and Riccardo Manzini. 2014. "Sustainable food supply chain management." *International Journal of Production Economics* no. 152 (0):1-8. doi: http://dx.doi.org/10.1016/j.ijpe.2014.04.003.

Mealling, Michael. 2002. "Dynamic delegation discovery system (DDDS) part one: The comprehensive DDDS."

Nicklas Hild. *Information about origin and sustainability lead to increased sales for Swedish fresh fish. Can traceability supply better information to consumers than labeling?* 2013. Available from http://www.tracefood.org/images/4/46/11%29_Niklas_Hild_ROI4USwedish_Board_of_Fishery_-_Can_traceability_supply_better_information_to_consumers_than_labeling.pdf.

Norwegian Seafood Council. *Trademarks and labelling* 2014 [cited 02.10.2014. Available from http://en.seafood.no/Marketing/Trademarks-and-labelling.

Oliot. *About Oliot* 2014a [cited 16.05.2015. Available from http://gs1oliot.github.io/oliot/#About.Oliot.

Oliot. *Object name service (ONS) implementation for Oliot project* 2014b [cited 16.05.2015. Available from https://github.com/gs1oliot/oliot-ons.

Oljedirektoratet. *Eksport av Norsk Olje og Gass* 2015 [cited 21.05.2015. Available from http://www.norskpetroleum.no/okonomi/eksport-av-norsk-olje-og-gass/.

Ruane, Janet M. 2005. *Essentials of research methods: A guide to social science research*. Blackwell.

Rune Ytreberg. 2015. "Norsk fisk verdt 69 milliarder." *Dagens næringsliv*, 07.01.2015.

Sarma, Sanjay, David L Brock, and Kevin Ashton. 2000. "The networked physical world." *Auto-ID Center White Paper MIT-AUTOID-WH-001*.

The Linux Information Project. *Freeware Definition* 2006 [cited 22.04.15. Available from http://www.linfo.org/freeware.html.

W3C. *Semantic web* 2015 [cited 16.05.2015. Available from http://www.w3.org/standards/semanticweb/.