NTNU
Innovation and Creativity

The mobile phone as a contactless ticket

Magnus Egeberg

Master of Science in Communication Technology
Submission date: June 2006
Supervisor: Van Thanh Do, ITEM
Co-supervisor: Juan Carlos Lopez Calvet, Telenor R&D

Norwegian University of Science and Technology
Department of Telematics

Problem Description

The creation of the Near Field Communication (NFC) technology has given the mobile phone the possibility to interact with its surroundings by simply "touching" them. It is now possible for the mobile phone to operate as a contactless smartcard, thereby adding new value to mobile phone users. Indeed, it will be very convenient for the user if the mobile phone can function as a bus, tram or train ticket. Such an electronic ticket should be capable of interacting both with ticket reader via the NFC (Near Field Communication) interface and the ticket system via the mobile network for ticket refilling.

The thesis work consists of the following task:
- Design of a ticketing system using NFC
- Implementation
- Testing

Assignment given: 13. January 2006
Supervisor: Van Thanh Do, ITEM

Preface

This master thesis is submitted to the Norwegian University of Science and Technology (NTNU) completing a five year Master of Technology program. The thesis is carried out at Telenor R&D in collaboration with the Department of Telematics, NTNU.

The topic of Near Field Communication is new and it has been challenging to find correct material on the topic. The thesis work has taught me to research and use material to develop new solutions.

I would like to thank everyone who helped me on this master thesis. Professor Do van Thanh has been academic responsible and Juan Carlos López Calvet from Telenor R&D has been project supervisor. I would like to thank them both for their support and feedback on my work. Special thanks go to Kjell Myksvoll at Telenor R&D, your assistance has been crucial to the success of this thesis. Last I will pay my appreciation to Telenor R&D for providing me with equipment needed in my work.

Fornebu, June 29, 2006

Magnus Egeberg

Abbreviations

APDU	-	Application Protocol Data Units
API	-	Application Programming Interface
CAD	-	Card Acceptance Device
CAP	-	Converted Applet
CDC	-	Connected Device Configuration
CLDC	-	Connected Limiter Device Configuration
CPA	-	Content Provider Access
DCU	-	Digital Control Unit
DoD	-	US Department of Defense
ECMA	-	European Computer Manufacturers Association
EEPROM	-	Electrically Erasable Programmable Read-Only Memory
EMV	-	Europay - MasterCard – Visa
EPC	-	Electronic Product Code
ETSI	-	European Telecommunications Standards Institute
GPRS	-	General Packet Radio Service
GSM	-	Global System for Mobile Communications
GUI	-	Graphical User Interface
ICC	-	Integrated Circuit Card
IDE	-	Integrated Development Environment
ISO	-	International Organization for Standardization
JAD	-	Java Application Descriptor
JAR	-	Java Archive
JCRE	-	Java Card Runtime Environment
JCRMI	-	Java Card Remote Method Invocation
JCVM	-	Java Card Virtual Machine
JNI	-	Java Native Interface
JVM	-	Java Virtual Machine
J2EE	-	Java 2 Enterprise Edition
J2ME	-	Java 2 Micro Edition
J2SE	-	Java 2 Standard Edition
KVM	-	“Kilobyte” Virtual Machine
LGPL	-	GNU Lesser General Public License
MIDlet	-	Mobile Information Device Application
MIDP	-	Mobile Information Device Profile
MSC	-	Mobile-services Switching Centre
msc	-	message sequence diagram
NFC	-	Near Field Communication
NFCIP	-	Near Field Communication Interface and Protocol
NTNU	-	The Norwegian University of Science and Technology
OTA	-	Over The Air
OTP	-	One Time Programmable
PBP	-	Personal Basis Profile
PCD	-	Proximity coupling device
PICC	-	Proximity integrated circuit(s) card
PIN	-	Personal Identification Number
PKI	-	Public Key Infrastructure
PP	-	Personal Profile
PSTN	-	Public Switched Telephone Network

RF	-	Radio Frequencies
RFID	-	Radio Frequency Identification
RMI	-	Remote Method Invocation
SDK	-	Software Development Kit
SIM	-	Subscriber Identity Module
SMS	-	Short Message Service
UHF	-	Ultra high frequency
UMTS	-	Universal Mobile Telecommunications system
URL	-	Uniform Resource Locator
VCD	-	Vicinity coupling device
WAP	-	Wireless Application Protocol
WMA	-	Wireless Messaging API
WORM	-	Write Once - Read Many
3GPP	-	3rd Generation Partnership Project
µC	-	Micro Controller

Terminology

Active Poster	-	This is a poster with an embedded chip making the poster able to offer some kind of service. The chip communicates with a reader by RF.
CPA	-	Represents several products enabling content providers to deliver services to mobile network operator's subscribers and at the same time bill the subscribers for using the service
EMV	-	Working group sponsoring the global standard electronic financial transactions, owned by Europay, MasterCard and Visa.
MIDlet	-	This is an application that conforms to the MIDP standard.
MIDlet suite	-	A collection of MIDlets packaged into a JAR-file. The suite also contains a JAD-file describing the suite.
MIDP	-	This is a set of J2ME APIs that define how software applications interface with cellular phones.
PSTN	-	Public Switched Telephone Network is the world's collection of voice-oriented circuit-switched telephone networks.
Transaction data	-	This is data from a transaction that is needed for a user to be able to use the results of a transaction, e.g. ticket.
Web Service	-	Web Service is a software system designed to support interoperable machine-to-machine interaction over a network.

Figure list

Table list

Abstract

Near Field Communication (NFC) offers contactless communication simply by bringing the communicating devices close. A NFC enabled device can operate as a contactless reader or as a contactless smart card, and is backward compatible with existing smart card standards. These characteristics make the technology suitable and attractive for contactless ticketing.

By including NFC in a mobile phone it is possible for the handset to operate as an electronic ticket in a contactless ticketing system. The ticketing system can then distribute electronic tickets trough the short message service (SMS) capabilities of the mobile phone.

This Master thesis presents current smart card- and contactless communication standards. It gives an analysis of contactless ticketing and designs a system that allows the mobile phone to operate as a contactless ticket. A prototype of the system is implemented and an evaluation of the prototype is given.

Contents

1 Introduction

The mobile phone has become more important in everyday activities for most people over the last decade. It has yet to see a breakthrough in electronic payment and ticketing, but this might change with the introduction of Near Field Communication (NFC). The RFID based technology eases the setup of device interconnection, and it also offers a basis for the mobile phone to incorporate contactless payments and electronic ticketing.

This thesis is a continuation of a project submitted the fall semester 2005 [1].The background section of this document is mainly an edited version of that project which explored the existing smartcard, RFID and NFC technologies and elaborates the requirements for electronic ticketing with NFC. The project also proposed a high level architecture for electronic payments with NFC enabled mobile phones, which has served as the basis for the development process of this thesis. The thesis is written in cooperation with Telenor R&D and the work will be continued in a Master thesis.

1.1 Motivation

Electronic ticketing is spreading as ticketing companies replace their analogue systems. The old tickets are replaced by smart cards communicating either trough a contact or contactless interface.

The introduction of NFC in cellular handsets opens for the possibility of the mobile phone acting as a contactless ticket in these systems. There are several reasons why this can be an interesting combination:

1. The mobile phone network can be used as a new distribution and sales channel for electronic tickets.
2. The mobile phone can store different types of tickets.
3. As the mobile phone can hold numerous tickets, there will be fewer items for the user to carry.

1.2 Challenges

For the mobile phone to operate as a contactless tickets there are some challenges that stand out as important:

1. For the tickets to be provided trough SMS there has to be a way to receive the SMS and get the ticket into the smart card.
2. The mobile phone should support different kinds of tickets.
3. The phone needs to make sure it stores the right ticket in the right place

These issues are important to address for the phone to make use of the SMS interface which regular electronic tickets do not have. It is also important for the phone to be able to support tickets from multiple electronic ticketing systems.

1.3 Objectives

There are three objectives of this master thesis.

First it should provide an overview of existing technology that is important in electronic ticketing. This will allow people that are new to electronic ticketing and contactless communication to get a picture of the possibilities given by this rather new technology.

Second it should analyze electronic ticketing, and design a system that will allow the mobile phone to operate as a contactless ticket.

Last it should implement a prototype of the system, based on the available technology. The prototype should form the basis of pilot study on the topic.

1.4 Related work

There is large interest in the evolvement of a contactless infrastructure. Different businesses see the opportunities the technology can provide to their sector, and as a result there is a lot of material on contactless technology in general. The following papers study different impacts of contactless communication.

1.4.1 Security aspects of RFID based e-payments

Peter Longva wrote a Master Thesis on "Security aspects of RFID based e-payment" at the Norwegian University of Science and Technology (NTNU) in 2004 [9]. It focuses on the security aspects of RFID based e-payments both with NFC enabled handsets and other RFID technologies. It explores how the RFID technology satisfies security requirements and describes some proposals for securing RFID e-payments. This is a different approach to the technology than will be seen in this project as the main focus of this project is data handling within the mobile phone. The study is yet important because it highlights the significance of security when dealing with contactless communication.

1.4.2 Advantages of contactless smartcards

The smart card alliance has written a white paper on the advantages of contactless smartcard technology and a comparison of existing standards [2]. The white paper focuses on the use of contactless technology for physical access systems and things to consider when choosing a standard for implementation of a system. This is interesting to this document because the NFC technology is compatible with the smartcard standards described, and it gives a deeper understanding of considerations that have to be made when companies enter the contactless domain.

1.4.3 Advantages of contactless payments

VeriFone, a global provider of payment technologies, presents its view on contactless payments in [3]. The white paper introduces the entities involved in contactless transactions, and it highlights the benefits of contactless payment over magnetic stripe and contact smartcard solutions. A description of possible usage of a contactless scheme is presented, but the paper does not involve anything about the NFC technology. It is more concerned with the general advantages of using a contactless technology for payment compared to different advantages with different contactless technologies.

1.5 Methodology

There are numerous ways to carry out research, and depending on the goal of the study it is common to use either a descriptive or a prescriptive approach in information technology research [4]. Descriptive research seeks knowledge about the nature of reality whereas prescriptive research, also known as design science, seeks to improve the performance of a task or system.

The work of this project has been carried out after a design science approach. Figure 1-1 shows the five steps involved in a design science process [5] and depicts what is done in this project and what the master thesis will continue.

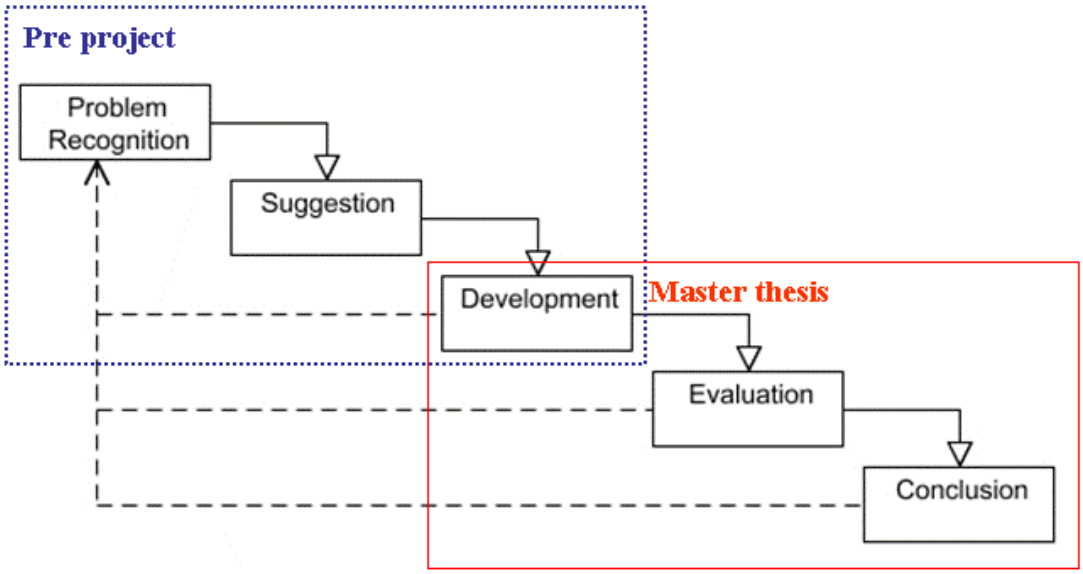

Figure 1-1: The design science research process.

Problem Recognition : This process involves studying the current situation and locating where there is room for improvements. It further contains work on how to conceptualize the environment and decisions rendering improvements.

Suggestion : This process involves deciding where the solution should focus and it introduces cognitive bias theory as a foundation for the reasoning to achieve a solution. Cognitive biases can be viewed as predictable results from rationality and is present in human reasoning [5].

Development : This is the main part of a system development project. Figure 1-2 shows an overview of the main parts of such a process.

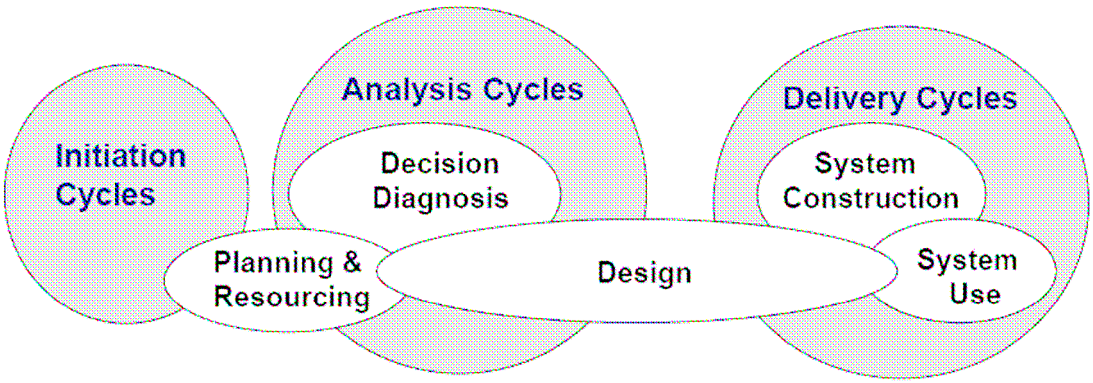

Figure 1-2: System development process leading to artifact [5].

The blue circles represent the main cycles in a development project, and the white ovals show how these cycles are linked together by shared activities.

Evaluation : This involves evaluating the system against the results of the suggestion process.

Conclusion : The success of the process can be determined by factors other than the suggested improvements given in the suggestion phase. Solution adoption and commercial realization are among factors that can determine the success of a project. This will be taken into consideration along with proposals for further studies.

This document is the result of a design science process, where the first steps were carried out as a pre project.

1.6 Organization of document

The document is organized into nine chapters, one bibliography and eight annexes. Depending on your level of knowledge and your goal for reading the report you can approach it from different angels. A reader seeking knowledge about using the mobile phone for payment and ticketing, but with little or no experience with RFID and NFC technology should probably read the report as it is outlined. More advanced readers can jump straight to chapter 3, 4 or 5 depending on the goal of the reading. It is recommended to read the parts of the background where the reader finds its knowledge inadequate. It is also possible to use the background section as a resource to learn about the different technologies. Table 1-1 gives an introduction to the content of each chapter.

Chapter	Description
1. Introduction	It provides an introduction to the problem at hand and an overview of the objectives with the project. In addition it explains the methodology used throughout the project.
2. Background	This part provides background information on technologies that are relevant to the project.
3. Problem statement	The problem statement gives a thorough explanation of the specific problem this project addresses.
4. Analysis	This part analyzes the problem and uses different tools to come up with possible user behavior and requirements to the system.
5. Design	The design provides a design of the system and its components.
6. Realization	Presents the realization of the design.
7. Summary	Presents a summary of the thesis.
8. Discussion	The discussion part analyzes the solution with respect to both a technical aspect and a business aspect.
9. Conclusion	The conclusion summarizes the results gained from the project.
Bibliography	Lists the references that have been used.
Annex A: ISO 7816-4 field values	Presents APDU values
Annex B: MasterCard PayPass questions	A response from MasterCard regarding PayPass
Annex C: Pilot studies	Current interesting pilot studies
Annex D: Scenarios	Additional scenarios
Annex E: Payment use cases	Additional use cases
Annex F: Specification of Nokia 3220	The specification of the phone used in the prototype
Annex G: Ticket from the Norwegian Public Roads Administration	An introduction to an electronic ticket offering interoperability.
Annex H: UML diagrams	Additional UML diagrams

Table 1-1: Organization of document.

Figure 1-3 outlines an overview of the document. The arrows show ways of going trough the document for different readers. The more the arrow is to the right, the more knowledge and experience should a reader following it have. For readers who want to get a quick overview of the document it is recommended to read the summaries at the end of every chapter and the main summary in chapter 7. The annexes are provided to give the reader more background information.

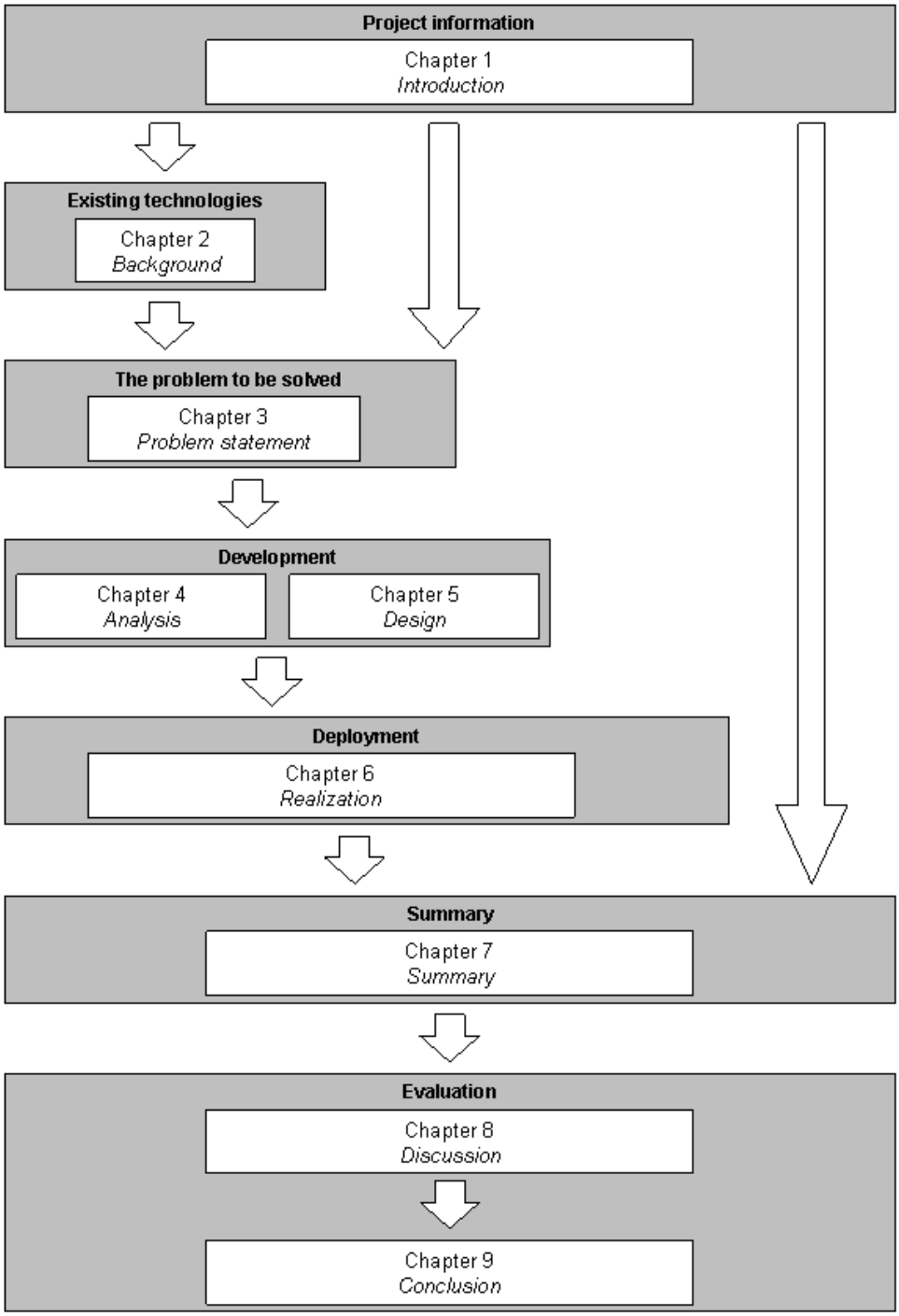

Figure 1-3: Document outline.

1.7 Summary

This chapter has provided an overview of the project. It has given a brief introduction to the problem at hand and presented the objectives and challenges of the study. The design science methodology which the project follows has been introduced in addition to an overview of this document.

2 Background

This section provides background information to better understand the problem statement and the scope of the assignment. It explores smartcard, RFID and NFC standards and provides an overview of their use as of today. The section also provides information about the java technology which is used to implement the prototype.

2.1 RFID

Radio frequency identification (RFID) is a technology for contactless identification of transponders trough a reader [6]. A transponder is basically a microchip connected to an antenna and a reader is an antenna able to read information from the tag. Objects can be labeled with transponders containing a variety of data, giving an opportunity to uniquely identify and track the objects. This is a capability that is highly desirable in many situations and the technology is expected to have a rapid growth in the next years.

2.1.1 How it works

The two basic components of an RFID system are the reader and the transponder, the transponder often referred to as a tag. The tags come in a great range of varieties with different capabilities and are often the main focus of these systems. They are often categorized by their power source. Active tags have an internal power source while passive tags are powered by the signal from the reader. The communication happens by the antennas emitting radio frequency fields and modulating a signal.

2.1.2 Active versus Passive Tags

The internal power source of active tags powers a transmitter that sends back a signal to the reader, thereby increasing the distance from which the tags can be read. The drawbacks with the battery powered tags are cost, as they are more expensive than passive tags. They also have a limited lifetime due to battery capacity and tend to be more bulky. The operating frequencies of these tags vary, but usually 455 MHz, 2.45 GHz or 5.8 GHz is being used [7].

The passive tags can be powered in two ways, depending on the operating frequency (see table 2.1). A tag operating in the low- or high-frequency bands is powered by inductive coupling. An electromagnetic field is created between the reader and the tag, and the tag uses power from this field to change the electric load on its antenna. The tags operating in the Ultrahigh-frequency (UHF) band are powered by propagation coupling. A tag will use the electromagnetic energy from a reader's radio-waves to send back an altered signal by changing the load on the antenna. This can be done either by changing the amplitude, phase or frequency of the signal [7].

2.1.3 Frequency bands

As previously mentioned the RFID tags can operate at different frequency bands and it is argued that the operating frequency of a system should be dependent on the specific application to maximize system performance [6]. In real implementation of such systems this approach alone is not practical due to regional regulations regarding available frequency bands and allowed signal strengths. A table of common operating frequencies is given in Table 2-1[7].

Frequency band	Most common frequencies
Low frequency	124 kHz / 125 kHz / 127 kHz
High frequency	13,56 MHz
Ultra high frequency	860 – 960 MHz / 2,45 GHz or other parts of radio spectrum

Table 2-1: Common operating frequencies for passive RFID tags.

The earliest RFID systems operated in the low frequency band and a brief examination is given in [6]. One of the positive things with this band is that many countries do not require licensing for using it, making it suitable for global applications. It is also positive that the signals are not as prone to reflection from metal and water as the higher frequency signals are, this certainly makes the band attractive. On the other hand the band also has some major drawbacks compared to the other two, it has slower data rate and a shorter read range. The slow data rate is a problem when one wants to read many tags in a short period of time, which is a key task in supply-chain applications.

The high frequency band is very commonly used as it is globally available and unregulated [6]. There are numerous applications operating in this frequency band, probably the most well known being contactless smartcards. NFC which will be described in further detail later also operates at 13.56 MHz.

The ultra high frequency band offers the highest data rate of these three bands, making the tags very suitable for supply chain applications where simultaneous reads are necessary [6]. The regulations on this band in Europe are stricter than in the USA regarding the power of the signal and the reader frequency range. This important difference has got a lot of attention as there is a concern that it will make it harder to develop global applications. Work is being done however to develop standards to counterfeit this problem.

2.1.4 Read and write capabilities

The tags have different read and write capabilities as described in [6], and this make them suitable for many different applications. The read and write capabilities of the tags can be divided into three categories; read-only, write-once and read-write.

The read-only tags are programmed with a code that uniquely identifies the tag within the lot produced. The small amount of memory needed and the lack of writing capabilities contribute to hold the costs of the tags down. To link the tags to a certain product or item the use of an external database is necessary. The database will store information about the tagged product and the amount of information stored is only limited by the size of the database. It is also argued that it is more secure to store the data in a database as the possibilities of taking precautions against data loss are much greater than if the data is only stored on the tag. It is also argued that read-only tags will be easier to integrate into already existing systems as the bar code system we know today is read only. On the other hand it is also an issue that tags relying on database communication adds a delay that could make them unsuitable for real time sorting applications [6].

The write-once tags have the capability of being programmed once after production. This capability gives a user the option to store whatever data the user wants, limited by the memory of the tag. These tags are also known as WORMs (Write Once - Read Many) or OTPs (One Time Programmable) [6].

The read-write tags can be read and written to as long as they are in use. This result in the most flexible solution and make the tags adoptable to many RFID solutions. The costs of these tags are slightly greater than the other ones [6], but as production volume increases the prices are expected to drop. These tags are crucial in environments where there is no infrastructure for database communication available.

2.1.5 Wal-Mart mandate to use EPC

In 2003 Wal-Mart, the world's largest retailer [8] issued a mandate to all of its 100 largest suppliers. By January 2005 these suppliers had to implement pallet and case level RFID tracking by labeling their supplies with electronic product codes (EPC). EPC is by many thought of as the next generation bar codes, and Wal-Mart's implementation of the technology is considered to be a boost for rapid deployment of the technology.

Electronic Product Code (EPC) is a standard developed for global tracking of goods using RFID technology [9]. The standard was developed by the Auto-Id Center, a research partnership started to develop a system for the tracking of goods using RFID technology. The system had to meet certain criteria [10]:

Low-cost	:	Because tags were to be disposed after use.
UHF band	:	Because the UHF band was the only one to deliver enough read range.
Open standard	:	To make it easier to get different companies to use the same technology.
Networked	:	Open standard over the Internet for companies to share product data.

The center developed a network that works over the Internet for companies to share data about the tagged objects. The center also came up with air-interface protocols and a categorization of tags according to their sophistication. The EPC generation 1 protocol got a lot of criticism because the tag categories used different air-interface protocols. This made the readers more expensive as they had to have multi protocol abilities in order to read all tags.

The Auto-Id center got split into two parts, EPCglobal and the Auto-id labs. The Auto-id labs focus on continuing the research from the Auto-Id center while EPCglobal focus on the development of EPC standards to get EPC technology deployed in the industry. The summer of 2004 the EPCglobal reached consensus on EPC generation 2 protocol [11] to solve many of the shortcomings of generation 1.

2.1.6 US Department of Defense mandate

The US Department of Defense (DoD) is a huge organization, giving it a lot of influence over its suppliers. The DoD mandate states that the United States Military is to work on implementation of passive RFID in its supply chain, and the requirements to be met are described in [12]. The implementation plan is specified in [13], and the implementation started as of January 2005. The DoD has chosen to require the suppliers to conform to the EPC standards, and has chosen not to support any particular RFID vendor. The suppliers are required to use EPC generation 2 when products are available.

The implication of the Wal-Mart and the DoD mandates are considered to be enormous. These companies are such important customers for their suppliers that their decision to use RFID labeling certainly will be result in their suppliers implementing RFID. This will reduce the cost of RFID tags, making it cheaper for new companies to implement RFID. There will

certainly be network externalities rising from this increasing adoption of the technology, and with these large customers taking the first step it is likely the RFID development has got enough momentum to some time in the future replace the bar code.

2.2 Smartcards

Smartcards have been adopted by different business sectors worldwide. It as originally thought of as the technology that would succeed the magnetic stripe cards in the credit card sector. This has not happened probably due to different reasons, but the cost of upgrading the payment infrastructure is considered to be a major cause. Instead the smartcards have seen wide adoption in the telecommunication industry. They have been used for prepaid cards and even more importantly as the subscriber identity module (SIM) in the GSM network. That makes the smartcard important in this project, because it is part of the authentication process of the user when an electronic purchase is done with a mobile phone.

The International Organization for Standardization (ISO) standard 7810 and 7816 series describe the detailed characteristics of a contact smartcard. The cards can be categorized into two different groups, with and without a microprocessor. Both groups have an integrated circuit able to store data, but the ones with no microprocessor can be looked upon as only a memory card. The other group can perform calculations and have extended capabilities due to its microprocessor

A smartcard is the size of a magnetic strip card and has the ability to store more data. The data is more secure on a smartcard than a magnetic stripe card because the data can be protected against unauthorized reads and writes by its cryptographic capabilities [14]. A list of smartcards' security advantages is also presented and some of the arguments are presented here:

Smartcards may increase the security of password based systems:

> The smartcard can store different passwords for various applications while the user only has to remember the PIN for the smartcard. This makes it easier to use good passwords and make it unnecessary for users to remember multiple passwords.

Two Factor Authentication:

> Many systems today only use one factor authentication, something you remember. By adding a second factor, something physical that you hold on to, the security of a system is harder to compromise.

Portability of Keys and Certificates:

> By storing keys and certificates on the smartcard these security mechanisms do not need be imported and exported in order to move them between hardware.

Non Repudiation:

> Transactions that are digitally signed are trusted because a digital signing only can take place by the proper PIN being entered.

2.2.1 ISO 7816

This standard is divided into fifteen parts, making up a detailed specification of a smartcard [15]. It specifies both physical characteristics, security related issues and command interchanges. For this thesis ISO 7816-4: Organization, security and commands for interchange is very important. This will be the standardized format for command interchanges in the system.

The communication happens by interchanging application protocol data units (APDUs), which either contains a command message or a response message. The command APDU is divided into a mandatory header and an optional body. Figure 2-1 shows the layout of the command APDU which will be used for communication, while Figure 2-2 describes the parameters of the command APDU.

Command APDU						
Header (mandatory)				Body (optional)		
CLA	INS	P1	P2	[Lc field]	[Data field]	[Le field]

Figure 2-1: ISO 7816-4 command APDU.

Code	Name	# Bytes	Description
CLA	Class	1	Class of instruction
INS	Instruction	1	Instruction code
P1	Parameter 1	1	To qualify the `INS` field, or for input data.
P2	Parameter 2	1	To qualify the `INS` field, or for input data.
[Lc field]	Length	Variable 1 or 3	The number of bytes present in [Data field]
[Data field]	Data	Variable = Lc	Byte array with command data
[Le field]	Length	Variable 1 or 3	Maximum number of bytes expected in [Data field] of the response APDU

Figure 2-2: ISO 7816-4 command APDU contents.

As a response to a command APDUs the target will respond with a response APDU. The structure of the response APDU is presented in Figure 2-3 while a description of its content is given in Figure 2-4.

Response APDU		
Body (optional)	Trailer (mandatory)	
[Data field]	SW1	SW2

Figure 2-3: ISO 7816-4 response APDU.

Code	Name	# Bytes	Description
[Data field]	Data	Variable	Byte array with response data
SW1	Status word 1	1	Command processing status
SW2	Status word 1	1	Command processing qualifier

Figure 2-4: ISO 7816-4 response APDU contents.

The values of these different fields are depends on the specific usage.

2.2.2 Contactless smartcards

These are smartcards with wireless communication capabilities. They communicate using Radio frequencies and have the same functionality as a contact smartcard. Today it is common to see them used in public transport systems, loyalty programs and access systems. They are also considered very interesting by many credit card companies because they will ease the use and shorten the transaction time of electronic payments.

2.2.2.1 ISO 14443

ISO 14443 is the international standard for "Identification cards - contactless integrated circuit(s) cards – Proximity cards" [16] - operating at 13.56 MHz is ISO 14443. The ISO 14443 was first developed for cards used in financial transactions and for identification, but is now a standard also including RFID contactless smartcards. The wireless communication is considered very useful for the robustness of smartcards as the physical contact point of a smart card has been a weak point due to exposure to dirt and wear.

The ISO 14443 standard consists of four parts:

Part 1: Physical Characteristics [16].

Two important terms are defined:

PICC - Proximity integrated circuit(s) card, the contactless card.
PCD - Proximity coupling device, the reader/writer.

The dimension of the PICC is required to meet the specifications of the ID-1 card specified in ISO 7810, the same as contact smart cards. The card has to meet certain criteria regarding robustness and numerous physical stresses the card has to withstand are listed.

Part 2: Radio frequency and signal interface [17].

The operating frequency is 13, 56 MHz ± 7 kHz. Two communication signal interfaces are defined, Type A and Type B, where only one can be active at a time. Only one interface can be active at a time and the initial communication is defined as follows (quoted):

1 *Activation of the PICC by the RF operating field of the PCD.*
2 *The PICC shall wait silently for a command from the PCD.*
3 *Transmission of a command by the PCD.*
4 *Transmission of a response by the PICC.*

Part 3: Initialization and anti-collision [18].

This part describes in detail the communication happening when a PICC enters a PCD's field. The initialization process is thoroughly explored along with the collision detection mechanisms for each of the two types. In order to read many PICCs in its field a PCD has to incorporate a selection mechanism to read the PICCs sequentially. The schemes for both type A and Type B selection is explored.

Part 4: Transmission protocol [19]

A half duplex block transmission protocol is specified in this part. The protocol-activation and deactivation for both Type A and Type B is explained.

2.2.2.2 MIFARE

MIFARE is a RFID based technology developed and licensed by Philips. The standard is fully compliant with the ISO 14443 standard [20] and the technology has been adopted in different applications around the world. The MIFARE Interface Platform has currently four different product families designed for different applications [20]:

1. MIFARE classic : Hardwired ICCs which fit into an ISO standard smartcard.
2. MIFARE ultralight : Designed to be cheap and fit into paper tickets.
3. MIFARE dual interface : Contains both a contact and a contactless interface.
4. MIFARE reader components : Readers and evaluation kits in compliance with contactless standards like ISO 14443 A /B and ISO 15693[1].

2.2.2.2.1 MIFARE Classic Card

There are currently two different MIFARE Classic ICCs on the market, the MF1 IC S50 [21] and the MF1 IC S70 [22]. Both the ICCs are passive and they differ only in memory. This document will explore the MF1 IC S50 and provide the data for the MF1 IC S70 memory. MIFARE classic card consists of a plastic card with an embedded antenna and a chip. The chip consists of an RF-Interface, Digital Control Unit (DCU) and memory (EEPROM) ref. Figure 2-5.

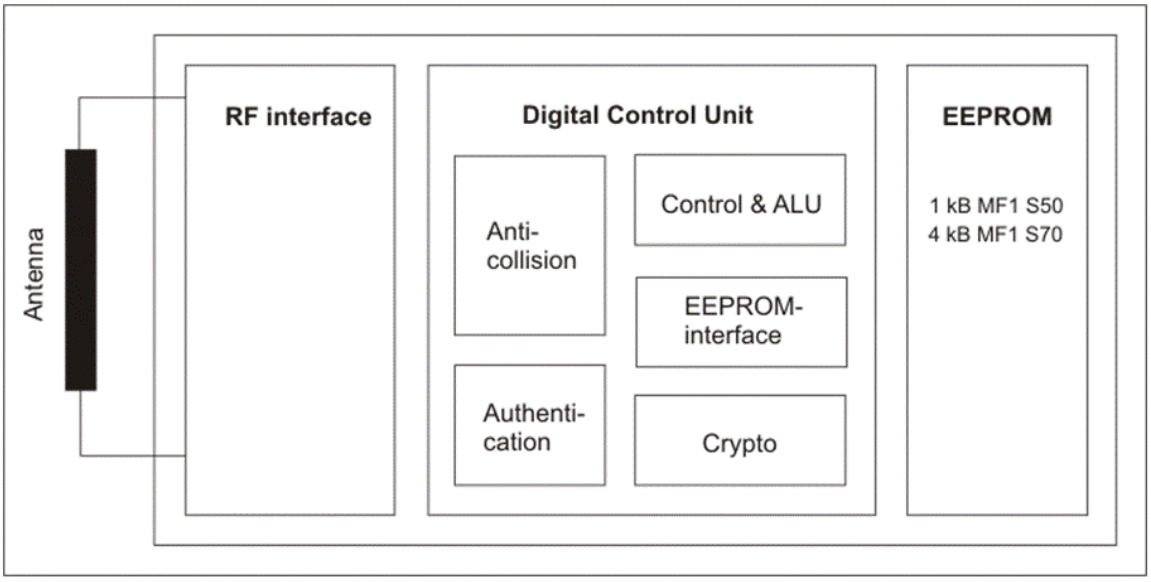

Figure 2-5: Standard MIFARE card.

[1] ISO 15693 is the ISO standard for contactless integrated circuit(s) – Vicinity cards, and can be considered an international standard for item level RFID tracking.

MIFARE Classic Antenna:
The antenna consists of four turns of wire coil around the card directly connected to the MIFARE chip.

MIFARE Classic RF interface:
The RF interface complies with ISO 14443A:

Operating frequency	:	13.56	MHz
Operating distance	<	100	mm.
Data transfer rate	:	106	Kbit/s.
Typical ticketing transaction	<	100	ms.

MIFARE Classic Memory organization:
The MF1 IC S50 has a 1 KB EEPROM memory, organized into 16 sectors containing 4 blocks with 16 bytes each (ref. Figure 2-6). For the MF2 IC S50 the memory is 4 KB organized into 32 sectors with 4 blocks and 8 sectors with 16 blocks. Each block consists of 16 bytes like the MF1 IC S50.

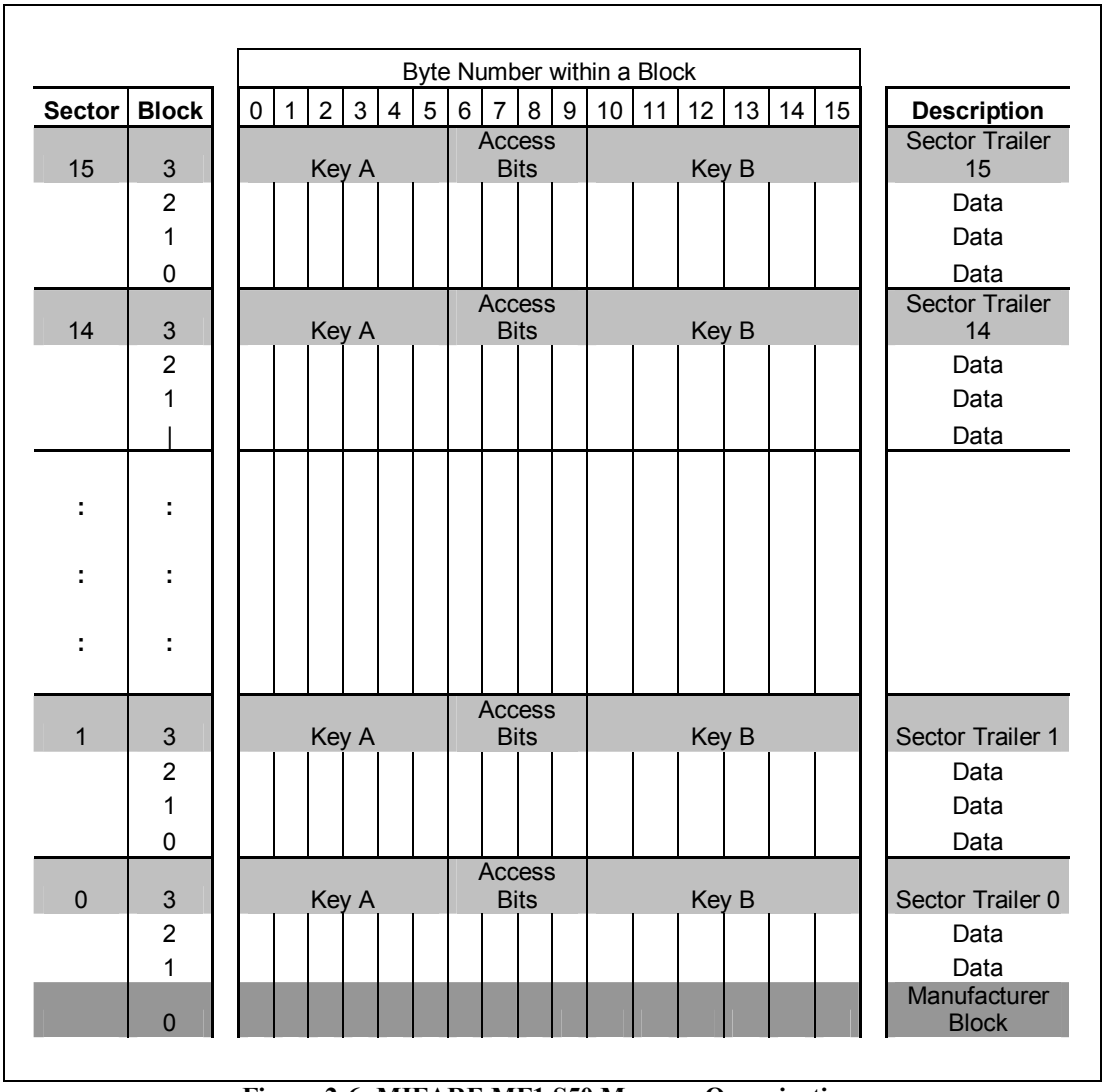

Figure 2-6: MIFARE MF1 S50 Memory Organization.

There are three different block types:

1. Manufacturer block

   This is the first block of the first sector and contains the manufacturer data of the IC. Once programmed by the manufacturer at the time of production this block is write protected. The first four bytes of the block contain the serial number.

2. Sector Trailer

   Figure 2-7 shows the byte numbering of a sector trailer. The access bits specify the type of data blocks and the access conditions of the sector's blocks, and byte nine is available for user data. The secret key(s) needed to be granted access are also stored in the sector trailer. The last six bytes of the sector trailer can be used to store data if Key B is not needed.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Key A						Access Bits				Key B (optional)					

Figure 2-7: Sector trailer byte number description.

3. Data blocks

   Data blocks can be configured in two ways:

   - read/write blocks
   - value blocks

   Read/write blocks are used in applications like access control while value blocks are used in applications where arithmetic on stored values is needed, e.g. electronic wallets.

   Figure 2-8 shows the byte numbering of a value block where the value is stored three times and the address is stored four times. The grey filling of Figure 2-8 indicates that the number is stored inverted, this is done to strengthen the data integrity and for security reasons. Negative values are stored in standard two's compliment format.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value				Value				Value				Adr	Adr	Adr	Adr

Figure 2-8: Data block byte number description.
The grey notation indicates that the number is stored inverted due to data integrity and security reasons.

MIFARE Classic Digital Control Unit:
The Digital Control Unit consists of five parts with different tasks [22]:

Anti-collision:

The anti-collision scheme is in compliance with the ISO 14443A standard described above.

Authentication:

Preceding any memory operation the PICC and the PCD performs a mutual authentication procedure according to the three pass authentication protocol described in ISO 9798-2 [23]. These are the steps in the authentication procedure (ref. Figure 2-9).

4. The PCD specifies which sector it wants to access and selects either key A or key B.

5. The PICC reads the secret key and checks the access conditions for the corresponding sector trailer. The PICC then sends a random number as a challenge to the PCD, message one in Figure 2-9. After this message the communication is encrypted [21].

6. The PCD calculates the response to the challenge and sends back a reply containing the response and a new random number as a challenge (ref TokenPCPI in Figure 2-9).

7. The PICC deciphers the message and compares the response to its original challenge. Then it generates a reply, TokenPIPC of Figure 2-9.

8. The PCD receives the reply and deciphers it. It compares the random number received originally with the random number of TokenPIPC, and it checks that the random number of TokenPCPI agrees with the random number of TokenPIPC.

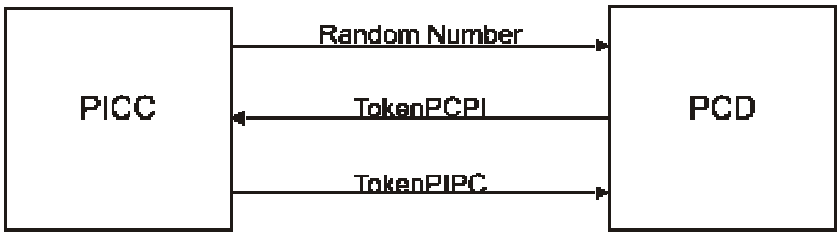

Figure 2-9: Three Pass Authentication.

Control and ALU:
The supported memory operations are presented in Table 2-2 (slightly modified table from [21]).

Memory Operations		
Operation	**Description**	**Valid for**
Read	Reads one memory block	Read/Write blocks, value blocks and sector trailer
Write	Writes one memory block	Read/Write blocks, value blocks and sector trailer
Increment	Increments the contents of a block and stores the result in the data register	Value block
Decrement	Decrements the contents of a block and stores the result in the data register	Value block
Transfer	Writes the contents of the data register to a block	Value block
Restore	Reads the contents of a block into the data register.	Value block

Table 2-2: Supported memory operations.

EEPROM-Interface:
It provides the access to the memory.

Crypto unit:
The control unit uses CRYPTO1 stream cipher.

2.2.2.3 SmartMX

Philips Semiconductors has developed the smartMX (Memory eXtension) platform for enhanced secure smart card ICs. The platform offers three interfaces, ISO 7816, ISO 14443A and USB 2.0 [24]. The standard offers linear addressing of up to 16 MB memory and contains hardware co-processors for enhanced security. The smartcard controller is considered important in providing security to transactions done with NFC, but that issue is outside the scope of this project.

2.2.2.4 FeliCa

FeliCa is a contactless IC standard developed by SONY for RFID and is widely adopted in Asia [25]. It has seen widespread use in transport ticketing- and electronic payment systems in countries like Japan and Korea, and it got implemented in the Hong Kong transport systems as early as in 1997 [26].

The FeliCa standard can be looked upon as an equivalent to MIFARE in Asian countries, and it is also supported by NFC. It uses a proprietary communication protocol and is compatible with 212 Kbps, Passive communication mode of ISO 18092 (ref. Figure 2-10).

In Japan Vodafone is launching a new service called “Vodafone live! FeliCa”. The launch coincides with the introduction of a FeliCa smartcard equipped mobile phone from Sharp. The new service provides customers with the ability to use their mobile handset for transport, electronic money and point services [27]. The launch is the beginning of an attempt to make the mobile phone more integrated and important in people’s everyday lives.

2.2.3 Visa and MasterCard contactless

EMV (Europay, MasterCard, and Visa) is an organization working to develop and maintain specifications for electronic payment using ICCs (Integrated Circuit Cards) with contacts. The organization is owned by JCB International, MasterCard International and Visa International by 1/3 each. Their standard is based on ISO 7816 [28] and the standard consists of four books:

- Book 1 - Application Independent ICC to Terminal Interface Requirements.
- Book 2 - Security and Key Management.
- Book 3 - Application Specification.
- Book 4 - Cardholder, Attendant, and Acquirer Interface Requirements.

The goal of EMV is to ensure wide acceptance and high interoperability for ICC payment systems. The organization is also responsible for developing tests to ensure terminal compliance with the EMV standards [29].

Visa has a vision of “Universal commerce”, meaning to provide for customers “to pay anywhere, anytime, in any way they choose” [30]. Within this vision is a movement towards contactless payment and Visa showed in 2004 a solution to withdraw funds from an EMV compatible contactless ICC in a secure manner [31].

MasterCard has launched a contactless solution named PayPass and the following information is based on information obtained from MasterCard [Annex A]. The cards can be ISO 14443 Type A or Type B while the readers have to support both Type A and Type B. The solution has two parts to it, the ISO 14443 based transport protocol and an application layer. The application layer is divided into two profiles, a magnetic stripe profile and an M/chip profile. The M/chip profile is the MasterCard application for EMV and is used where the payment infrastructure is EMV based. The magnetic stripe profile is used where the payment infrastructure is magnetic stripe based.

PayPass readers can read NFC devices because NFC is ISO 14443 A compliant. According to MasterCard this is they primary involvement with NFC forum, to ensure that NFC stays ISO 1443A compliant. The use of NFC devices as PayPass readers is theoretically possible, but not something that is expected to be seen in the near future according to MasterCard. This would have made it possible to carry out PayPass payments between two NFC devices, offering secure transactions between i.e. mobile phones incorporating NFC. This is not possible now because the current PayPass readers do not support the NFC protocol (ref. 2.3) and that the payment infrastructure is costly to upgrade [*Annex B: MasterCard PayPass questions.*].

2.2.4 Public transport ticketing systems

Public transportation systems have implemented automatic contactless ticketing systems many places including London, Hong Kong and Amsterdam. In China over 60 cities had implemented contactless smartcard technologies in their public transportation systems as of 2004 [32].

2.2.5 Electronic ticket interoperability in Oslo

The public transport system in Oslo will in the coming years see a migration to a contactless ticketing system. The three large cooperating public transport service providers within Oslo and Akershus county (AS Oslo Sporveier, Stor-Oslo Lokaltrafikk AS and Norges Statsbaner AS) setup a Common Specifications for Interoperability (CRSI) [33] for an automatic fare collection system In 2002. As of June 2006 the system is delayed and no new time for deployment is given. The system uses MIFARE technology and will support Philips DESFire and Philips Ultralight cards.

2.3 Near Field Communication

Near field communication (NFC) is originally an effort between Royal Philips Electronics and Sony Corporation to develop an open standard technology to make connectivity between close coupled devices easier. They formed the NFC Forum in 2004 to "promote implementation and standardization of NFC technology to ensure interoperability between devices and services" [34]. The NFC Forum has as of November 2005 over 50 member companies and a sponsor group of 12 world-known companies in key industries.

NFC is a short range RFID based contactless smartcard technology and operates in the 13, 56 MHz frequency band. The technology is backward compatible with current standards for contactless communication and it supports two protocols on its own, NFCIP-1 and NFCIP-2. A NFC chip can operate both as a contactless card and as a contactless reader, making the standard very suitable for device identification and communication initialization.

In order for two NFC devices to start communication the user only have to bring them physically together, thereby the term "touch" initiated communication. The devices will then setup a peer-to-peer network and exchange configuration and authentication data. The devices can then engage in transactions using any of the compatible protocols or setup a connection using faster and longer range protocols like Bluetooth or WiFi.

Philips and Sony jointly developed the NFC Interface and Protocol and submitted it for adoption by the European Computer Manufacturers Association – international (ECMA). The protocols got approved under the names NFCIP-1 and NFCIP-2. The protocols have later been accepted by both ISO and ETSI.

NFCIP-1 specifies the interface and protocol for NFC mode communication. The NFCIP-2 specifies a scheme to select which of the compatible communication standards should be used, avoiding interfering with ongoing compatible communication in the 13,56 MHz band.

2.3.1 Backward compatibility

NFC is backward compatible with ISO 144443A / MIFARE and FeliCa (ref. Figure 2-10), the most widely adopted contactless smartcard standards today. The standard is also backward compatible with ISO 15693, "Identification cards – Contactless integrated circuit(s) cards – Vicinity cards" [6], a widely adopted standard for item-level RFID tracking. This backward compatibility is important because it enables NFC to be used with already existing contactless infrastructure.

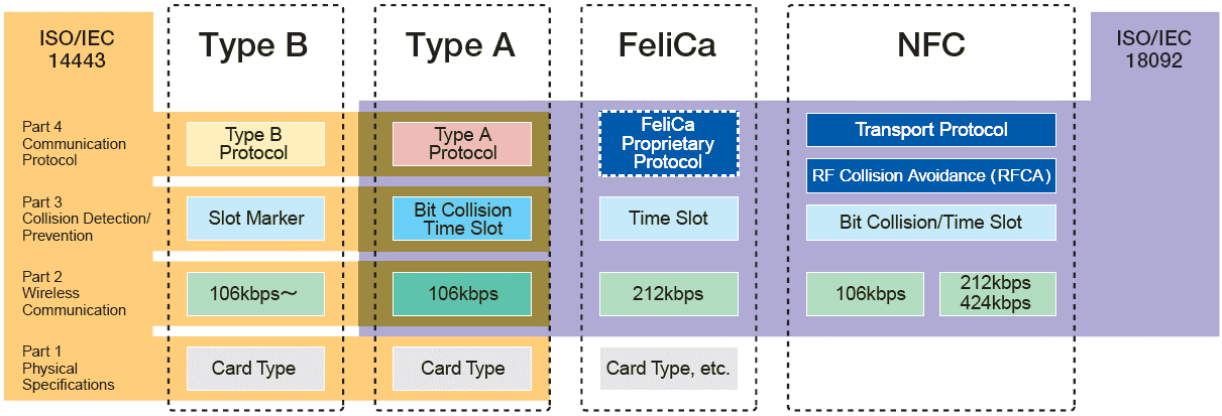

Figure 2-10: NFC backward compatibility with contactless smartcard technology [25].

2.3.2 NFC Interface and Protocol

Due to the backward compatibility NFC has several Interfaces and Protocols for communication, it can communicate through the above mentioned protocols (ref. 2.3.1) or it can communicate trough its own protocol NFC-IP1. The protocol chosen is named communication mode, and the choice of communication mode is made by NFCIP-2.

2.3.2.1 NFC communication mode

NFCIP-1 describes a transport protocol, initialization procedures and RF specifications, i.e. signal modulation, and is specified in ISO 18092 [35]. It specifies the requirements for devices to be in compliance with the NFC standard.

The standard specifies two communication modes that NFC devices shall support, active and passive mode:

Active mode:

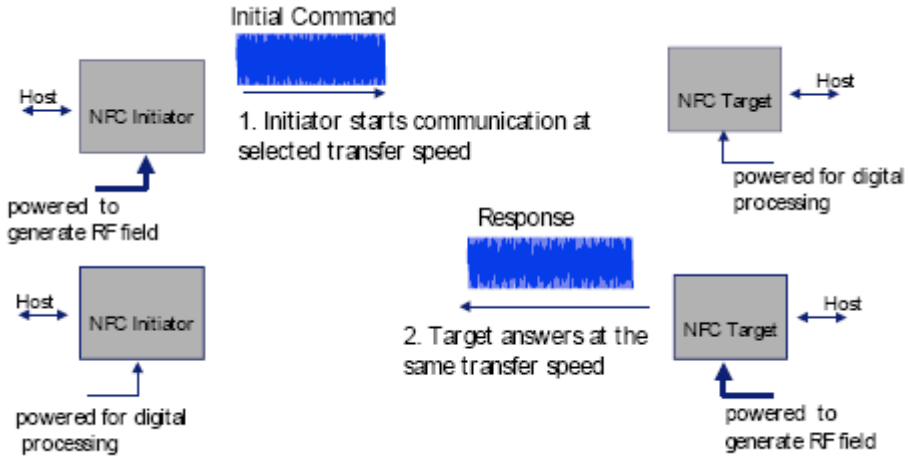

Figure 2-11: NFC active mode [38].

In this mode both the initiator and the target use their own RF field to communicate. The initiator starts the communication and turns off its RF field so that the target can respond by setting up a self generated RF field, ref Figure 2-11

Passive mode:

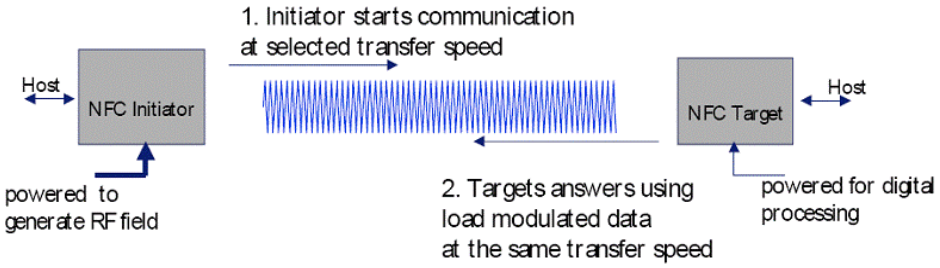

Figure 2-12: NFC passive mode [38].

In this mode the initiator starts the communication the same way as in active mode, but it does not turn off its RF field. The target answers using a load modulation scheme on the initiator's RF field, ref. Figure 2-12.

All NFCIP-1 devices must have the ability to communicate at 106 kbps, 212 kbps and 424 kbps in both active and passive mode.

The general protocol flow is described as follows:

- Default mode for a NFC device is target.
- In target mode it shall wait silently for a command from an initiator and not generate a RF field.
- If an application requires initiating communication, the device should change to initiator mode.
- The application shall decide the communication mode and transfer speed.

Figure 2-13 shows a more detailed flow of a transaction using NFCIP-1:

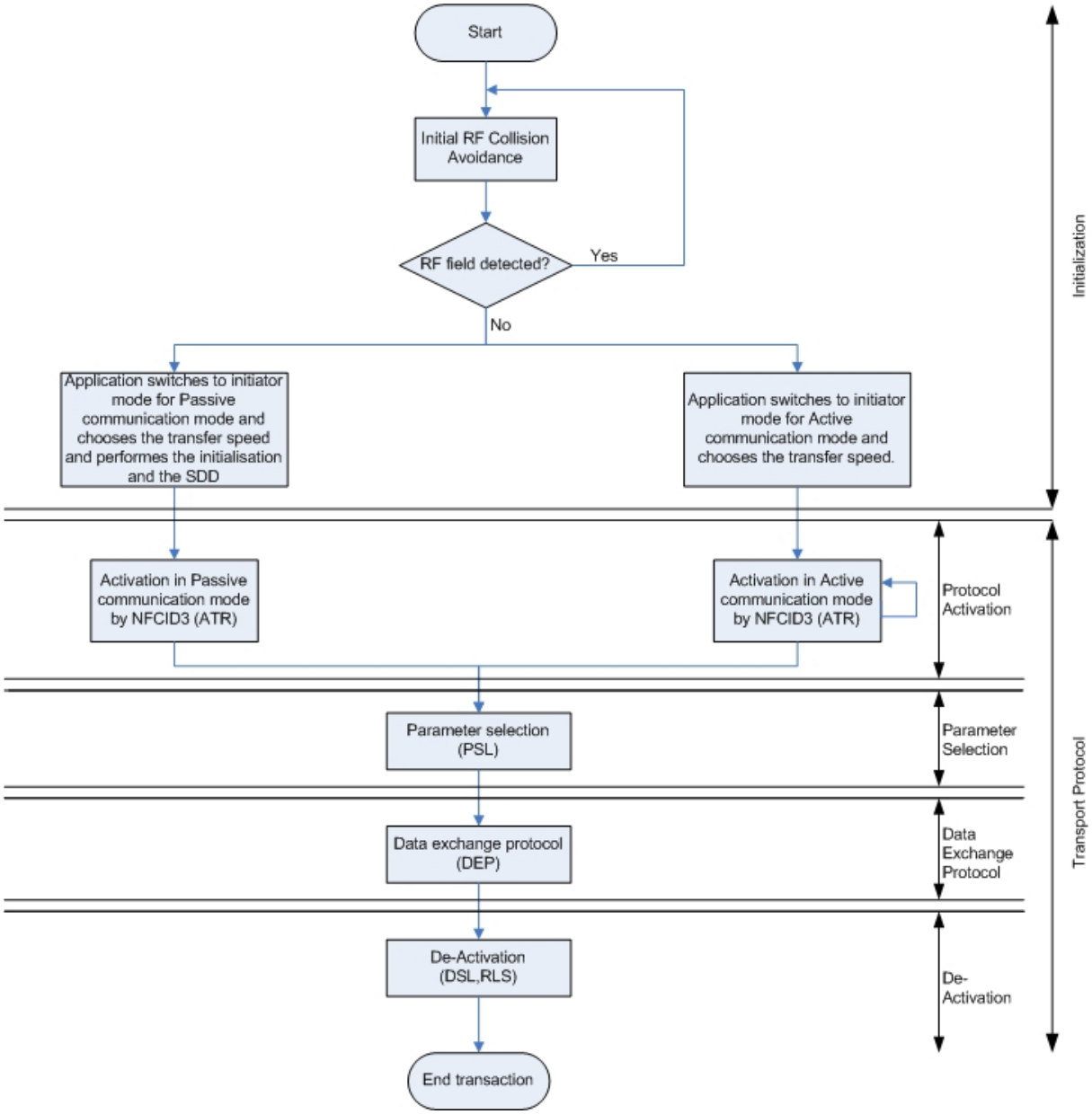

Figure 2-13: General initialization and single device detection flow from ISO 18092.

2.3.2.2 Communication mode selection

ISO 21481 specifies NFC-IP2, "Information technology —Telecommunications and information exchange between systems — Near Field Communication Interface and Protocol -2 (NFCIP-2)" [36]. It describes the mechanism to detect and select which of the available communication standards is to be used, and the standards define different operating modes. All the supported standards operate in the 13,56 MHz band, and NFCIP-2 is designed not to interfere with any ongoing communication with these protocols. A NFCIP-2 device has to implement the following operating modes:

1. NFC mode Initiator and target as specified in ISO/IEC 18092
2. PCD mode Specified in ISO/IEC 14443
3. VCD mode Specified in ISO/IEC 15693

The NFC mode uses the NFC transport protocol, while the PCD mode is used when the contactless card operates as a ISO 14443 card, i.e. MIFARE. The VCD mode is used when the contactless communication is in accordance with ISO 15693, a widely adopted standard for item level RFID tracking.

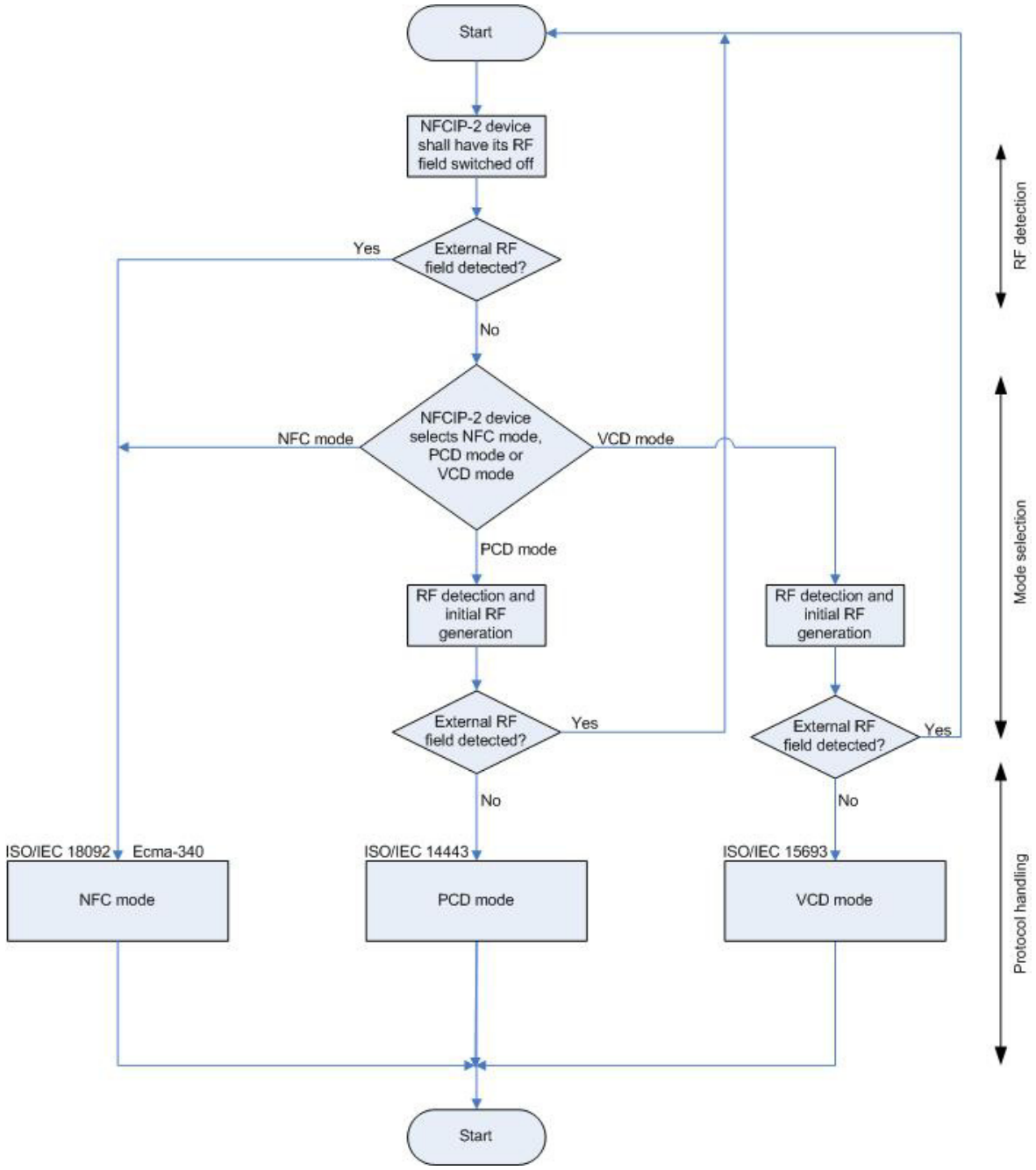

Figure 2-14: NFCIP-2 selection of operating mode.

Figure 2-14 shows the mode selection procedure of NFCIP-2. The device shall have its RF-field switched off and enter the NFC mode if it detects an external RF-field. If no RF-field is detected the device can choose which mode it wants to enter. Upon entering the PCD- or the VCD mode the device performs RF- detection and generation. If no external RF-field is detected the device will enter the corresponding mode and comply with the mode specification [37].

2.3.3 Micro controller based transmission module

PN531 is a μC (micro controller) based transmission module from Philips [38]. It is included in this document to show how an available chip on the market today can act both as a reader and emulate a contactless card. The micro controller communicates using different protocols and supports the following operating modes:

Reader / Writer mode:

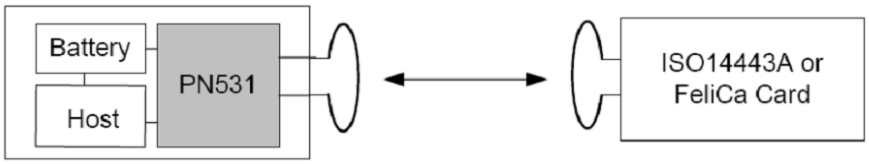

Figure 2-15: Reader / Writer mode [38].

Figure 2-15 shows how the chip in reader / writer mode can communicate with a passive ISO 14443A / MIFARE or FELICA card.

NFC mode:

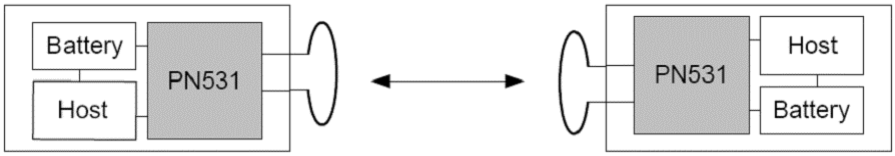

Figure 2-16: NFC Mode [38].

Figure 2-16 shows how two devices can communicate trough the NFCIP-1 protocol. The communication can be either active or passive, ref. 2.3.2.1.

Card Mode:

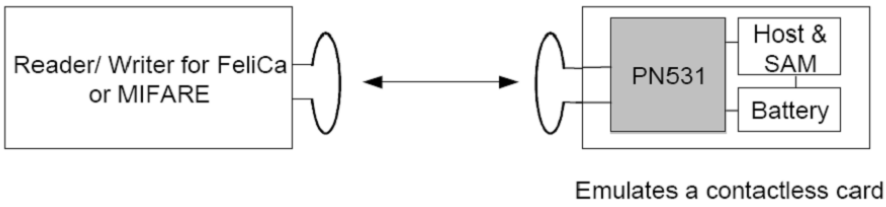

Figure 2-17: Card Mode [38].

Figure 2-17 shows the chip communicating with a FELICA or ISO 14443A / MIFARE reader / writer, the chip acts a contactless card.

2.4 JAVA

The pre-project to this thesis developed a conceptual design. This thesis further develops that design and implements it on a mobile phone. This migration from conceptual design to implementation involves choosing a programming language and an operating environment to deploy the solution. This project will make use of java technology as it is supported by the phone where the solution will be deployed. The Java technology offers a runtime- and programming environment to develop applications, and an increasing number of handsets are supporting the Java technology.

The Java platform seeks to make the application development independent of the underlying hardware and operating system of the device. This makes the applications hardware- and operating system independent, making it easier to develop applications that can be used independent of the handset manufacturer.

The platform differs from many other platforms in that it is a software only platform that runs on top of other hardware platforms. The Java 2 Platform consists of three elements [39]:

- Java programming language
- Java Virtual Machine (JVM)
- Application Programming Interfaces (APIs)

A program written in Java goes trough two steps in order to run on a hardware platform. First the program has to be compiled into byte code, and this is done by a java compiler. Then in order to run, an interpreter in the java virtual machine has to interpret the byte code into the appropriate machine code. By having java virtual machines for different hardware platforms the java programs do not have to consider which hardware platform it will run on, this will be taken care of by the java virtual machine. This is the idea behind the java vision of "write once, run anywhere". The APIs are code that is already written and ready for reuse trough a well defined interface.

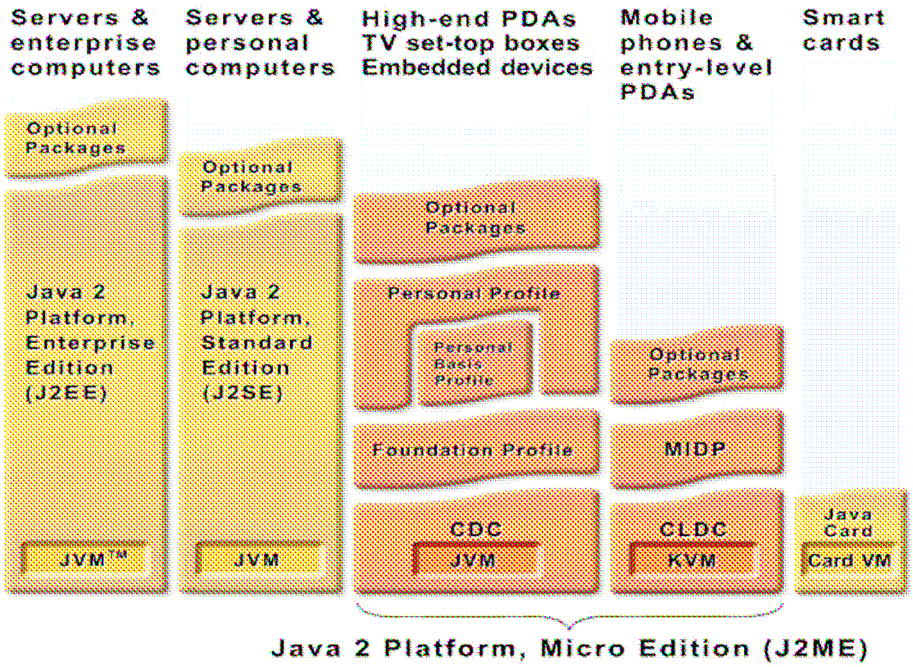

Figure 2-18: Overview of the JAVA environment [40].

Figure 2-18 shows the Java environment and the different editions that are available to make Java fit in different environments [40]. J2EE is a standard for the development of component bases enterprise applications. It supports the development and execution of web services. J2SE is a standard for regular server and desktop applications and J2ME is the standard for applications operating on consumer and embedded devices (ref. 2.4.1).

2.4.1 Java 2 Platform, Micro Edition

The Java 2 Platform, Micro Edition (J2ME) provides the benefits of a java environment to consumer devices with limited resources like battery capacity, memory, processing power etc. An increasing number of mobile phones have added java support over the last two or three years, and the number of java applications available for mobile phones are steadily increasing. This project will be implemented using J2ME with the architecture shown in Figure 2-19. At the bottom is the device specific operating system, which in this project is a Nokia operating system. Above that are several J2ME components which are explained in the subsequent sections.

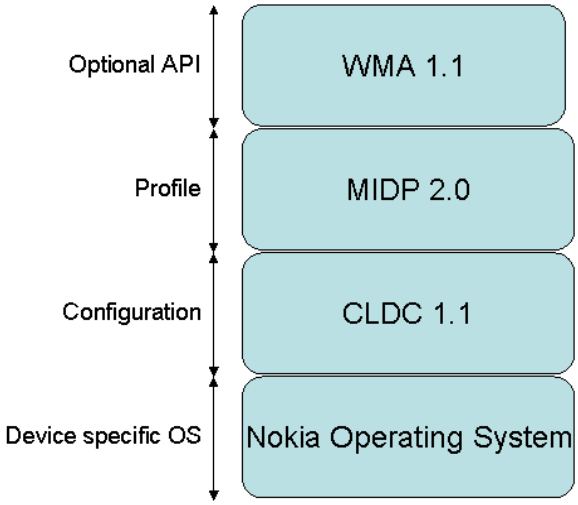

Figure 2-19: The Java architecture used in the MIDlet.

2.4.1.1 Configuration

A configuration consists of a virtual machine and minimal set of class libraries providing the basic functionality for devices sharing similar characteristics, i.e. network connectivity and memory footprint. At the bottom of the J2ME architecture are two configurations (ref. Figure 2-18):

- Connected Device Configuration (CDC)
- Connected Limiter Device Configuration (CLDC)

The CDC provides a full-featured Java virtual machine and a subset of the J2SE platform. It is designed for devices with little power constraints and good network bandwidth.

The CLDC is designed for devices with limited power, memory and processing power, i.e. mobile phones. It is a subset of the CDC and offers less functionality. It uses a basic virtual machine (KVM) which is a subset of the fully-featured JVM. One important aspect of the

KVM is that it only includes a subset of the standard bytecode verifier of the JVM [41]. This means that the verifying of classes has to be split between the KVM device and some external device. This has serious security implications and is addressed by different security architecture in CLDC compared to J2SE.

2.4.1.2 Profile

A profile is a set of higher level APIs shared by a category of devices as part of the runtime environment. A profile defines access to device specific properties and the life cycle model and user interface for an application. For this project the Mobile Information Device Profile (MIDP) is important because it is the profile that fits most mobile phones. Applications conforming to this profile are called MIDlets. MIDP offers functionality like memory management and user interface to applications running on MIDP device. The user interface functionality is important because it allows the MIDlets to easy adapt to the different screens and keypad layouts of mobile phones. The memory management is important because it makes sure that a MIDlet only can write to memory belonging to the KVM. That makes sure the MIDlets do not override each other or each others' data.

Devices conforming to MIDP needs to have an application manager, a device specific piece of software that controls the installation, running, deletion and the state of the MIDlets. The application manager requires the MIDlets to be packaged into MIDlet suites when they are deployed. A MIDlet suite can contain one or several MIDlets, and consists of the following:

- Class files and resources making up one or many MIDlets packaged into a JAR-file.
- Manifest file containing additional information needed at runtime and by the application management software.
- JAD-file (Java Application Descriptor) describing the MIDlet suite JAR.

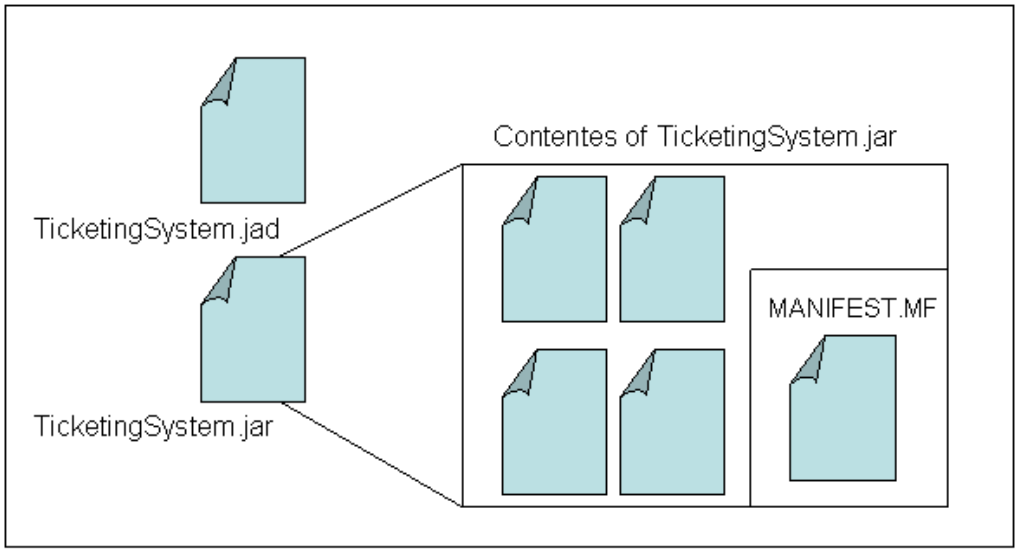

Figure 2-20: Contents of MIDlet suite TicketingSystem, edited from [41].

Figure 2-20 shows the contents of a MIDlet suite named TicketingSystem. The application manager will use data from the manifest to decide whether to load a MIDlet suite. The

application manager will then use the attributes within the JAR at runtime. The MIDlet itself can at runtime use both the manifest and the JAD-file to obtain properties.

2.4.1.3 Optional Packages

These packages offer APIs for using wireless communication technologies, Web services and multimedia. They are modular and device manufacturers can include them to meet very specific market requirements. This project makes use of the Wireless Messaging API (WMA), which is a package for interconnection with wireless communication methods like Short Message Service (SMS). When a MIDlet wants to send or receive SMS it uses the WMA and identifies with a port number which then is considered the MIDlets address. The port number is set in the header of a SMS. The API supports sending of SMS as text or binary message.

This project makes use of SMS in conjunction with the push registry. The push registry lets MIDlets register for incoming connections. The phone listens for incoming connections, and when a connection matches a MIDlet in the push registry the MIDlet is launched. This technique is called push. The MIDlet can either register in the push registry at run time or at installation time by special entries in the JAD-file.

2.4.1.4 Security

Due to the limited resources in the devices implementing a CLDC MIDP architecture, the security model is different than in the case of a standard J2SE architecture. The article in [42] presents a thorough description and evaluation of the security model. Figure 2-21 shows the security model for a CLDC MIDP architecture.

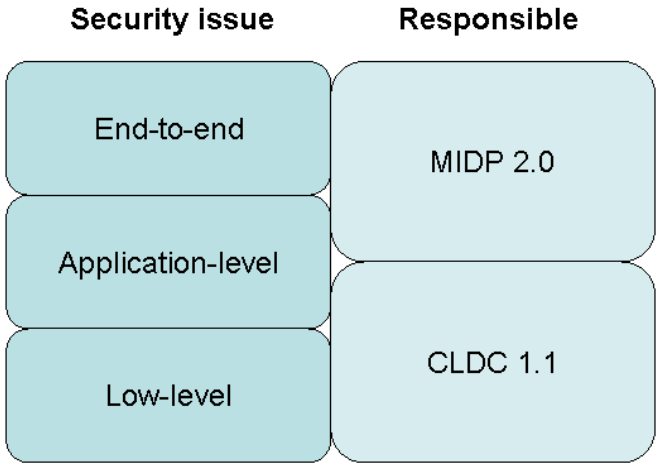

Figure 2-21: The security model of a CLDC MIDP architecture.

The security issues are classifies into three levels:

- Low-level : Addressed by the CLDC and deals with ensuring that the class execution is in accordance to the JVM specification.
- Application level : Addressed by both the CLDC and MIDP. Deals with resource handling and supervision, ensuring the applications do not violate their resource limitations.
- End-to-end : Addressed by MIDP and deals with networking security.

The low level security is handled by the class file verifier and ensures that the class files do not contain illegal instructions, can not execute in illegal order and cannot reference memory outside the object heap. Verification is performed in two steps, where the first happens off the device. This pre-verification removes certain bytecode and adds StackMap attributes to speed up the second verification step. That step happens in the device at runtime when the actual class file verification takes place.

The application level security is spilt between CLDC and MIDP. CLDC uses a sandbox model to ensure that all class files are properly verified and to restrict the available APIs to a pre-defined set. The model also ensures that MIDlet management takes place inside the KVM, protecting it from modification. In addition to this the system packages are protected against modification and an application is only allowed to load application classes from its own JAR.

MIDP 2.0 allows MIDlets access to sensitive APIs based on permissions and protection domains. Sensitive APIs are APIs that offer network connectivity and will result in user charges. The phone has a set of pre-defined protection domains which have different access rights to these APIs. When a new MIDlet suite is installed, the MIDlets get access rights depending on which protection domain they are placed in. The protection domain placement depends on handset rules which are defined in a policy file used by the application management system. Factors in the policy can for instance be source address or authentication of the MIDlet suite using PKI. It is worth noting that any MIDlet is free to obtain access to these APIs by asking for user permission. MIDP 2.0 also offers the MIDlets to store data in persistent memory and share the data between MIDlet suites.

The end-to-end security handled by MIDP 2.0 offers HTTPS with server authentication. A certificate in the device is used to verify a certificate chain provided by the server.

2.4.2 Nokia NFC shell SDK

Nokia has developed a software development kit (SDK) for their NFC enabled mobile phones [43]. It allows for custom development of MIDlets exploring the RF interface and back end system. It includes an API for communicating with external devices.

2.4.2.1 Services in reader mode

When a NFC device operates as a reader it involves service discovery. The NFC device reads a tag and the tag's data will be provided to invoke the offered service. Three services are currently available:

1. Send an SMS
2. Connect to an URL
3. Call a number

The following scenarios will describe possible usage of the different services:

Scenario 1: Sending a SMS.

> Poster advertising for a new album can contain a RFID tag programmed so that the NFC device reading it will automatically send an SMS to request a ringtone from the album. By "touching" the poster with the mobile phone the SMS will automatically be

generated and sent, and the ringtone will be sent in return. This provides an easy and user friendly way of using advanced services.

Scenario 2: Connect to an URL

A poster at a bus stop can be programmed with the position of the bus stop the URL of the bus company's web site containing updated schedule information. When a mobile phone reads the tag in will automatically connect to the website by the use of GPRS and get the most recent schedule or traffic information.

Scenario 3: Call a number

A product can be tagged and programmed with the phone number to customer support. If a customer needs support he/she can just "touch" the tag and the phone will automatically dial the number to customer support. This is an easy way to provide customers with easy access to product support.

2.4.2.2 Services in card mode

When an NFC device operates in card mode it provides data to an application reading the card. The device can have obtained the data numerous ways, i.e. the tag being written to by a RF reader or as a receipt for the use of one of the offered services (ref. 2.4.2.1).

The following scenarios are some examples of the NFC device operating as a card:

Scenario 1 : Ticketing

The device can be used as an electronic ticket. The ticket can be stored on the phone either trough the contactless interface, or when a mobile phone is NFC equipped by sending it to the phone as SMS. When the ticket is to be examined a reader will read the ticket stored in the device.

Scenario 2 : Credit Card

The device can work as a credit card operating in accordance to the standards the credit card issuers follows (i.e. Visa and MasterCard). The card should then follow the guidelines provided by EMV [29].

Scenario 3 : Electronic Key

The device can operate as an electronic key when the lock on the door is equipped with a compliant reader. At hotels this means that a customer's NFC device can be enabled as the key to the hotel room for the length of the stay at check-in time.

2.4.3 Nokia Secure Chip SDK

The Secure Chip SDK is a tool from Nokia for developing MIDlets communicating with the secure chip of the Nokia NFC shell for payment and ticketing. The SDK comes with an API offering methods for managing the communication modes of the shell. This is necessary because the shell offers both internal communication to the secure chip and external communication to other NFC / RFID devices. The API specifies the communication to the

secure chip trough Application Protocol Data Units (APDUs). These APDUs then communicate with a Java Card applet (ref. Section 2.5) on the secure chip.

2.5 Java Card

Java card technology is used to develop and run applications for smartcards. The applications are called java card applets and the platform consists of three specifications [44]:

- Java Card Virtual Machine : Defines a subset of Java and a virtual machine for smartcards.
- Java Card Runtime Environment : Further defines the runtime behavior of java enabled smartcards.
- Java Card API. : Defines the core framework and extensions for the applications.

Like with CLDC the java card virtual machine (JCVM) is implemented in two parts. The external virtual machine is part of the development process and verifies and prepares the applet for on-card execution. It produces a Converted Applet (CAP) file, which includes the applet's classes in a binary representation. The on-card virtual machine incorporates, manages and executes the applet according to the Java Card Runtime Environment Specification (JCRE).

The Java Card API contains a small subset of J2SE APIs in addition to defining its own classes to fit java card applications. On top of these APIs are often vendor- or industry specific APIs customizing the functions of the smartcard. Figure 2-22 shows the architecture of the java card application used in this project. The JetZ extension API is a special API in the Nokia payment and ticketing shell offering access to the secure chip from MIDlets.

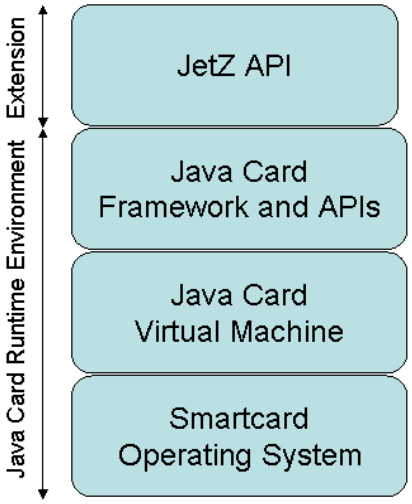

Figure 2-22: The architecture of the java card application on the secure chip.

For the java card application to be useful it is usually a part of a larger system. The smartcard can hold many applets from different vendors. Figure 2-23 shows the architecture of a system using java card applets. The card side shows the applets on the smartcard, and the architecture of these. The host application on the reader-side of the system can reside on a personal computer, a mobile phone, a banking terminal etc. It handles the communication with the

users and back-end systems. The Card Acceptance Device is the interface between the host application and the smartcard.

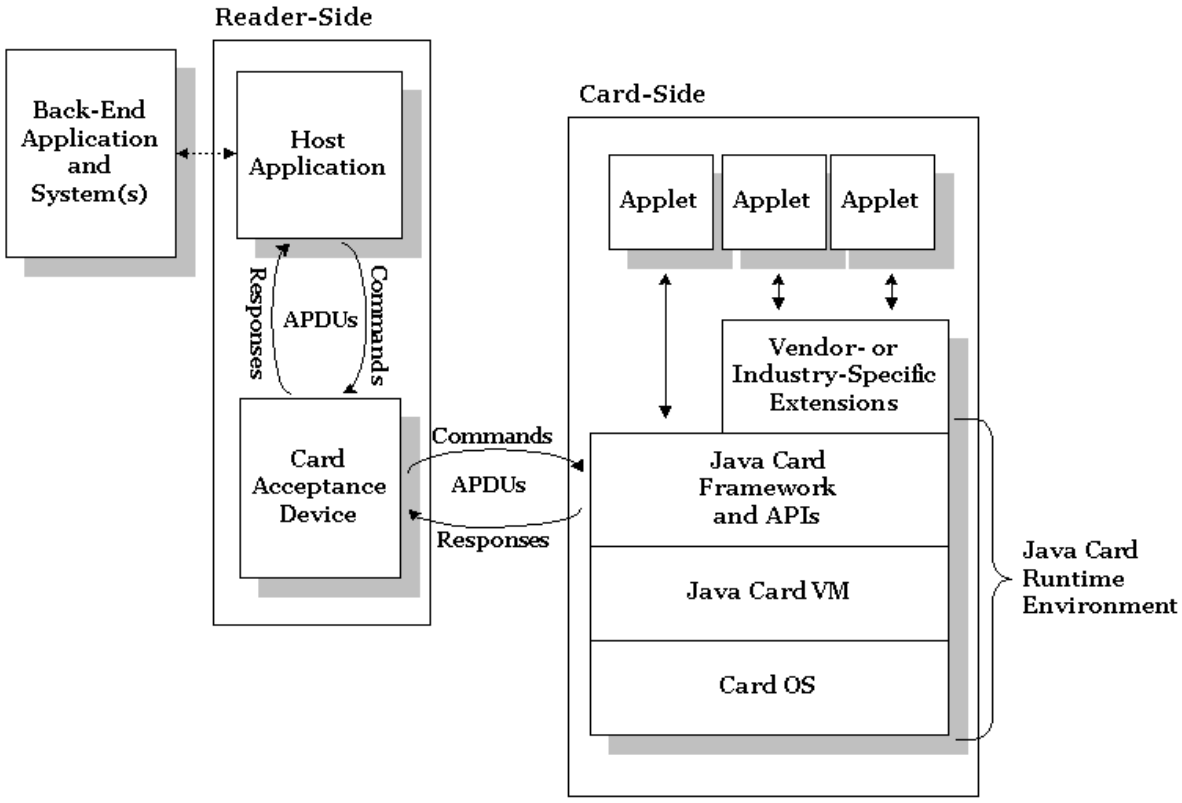

Figure 2-23: Structure of system using java card [44].

Communication happens by exchanging APDUs confirming to the ISO 7816-3/4 standard through the CAD. The communication can be either message passing or java card remote method invocation (JCRMI). JCRMI is a subset of J2SE RMI and is built on top of the message passing model.

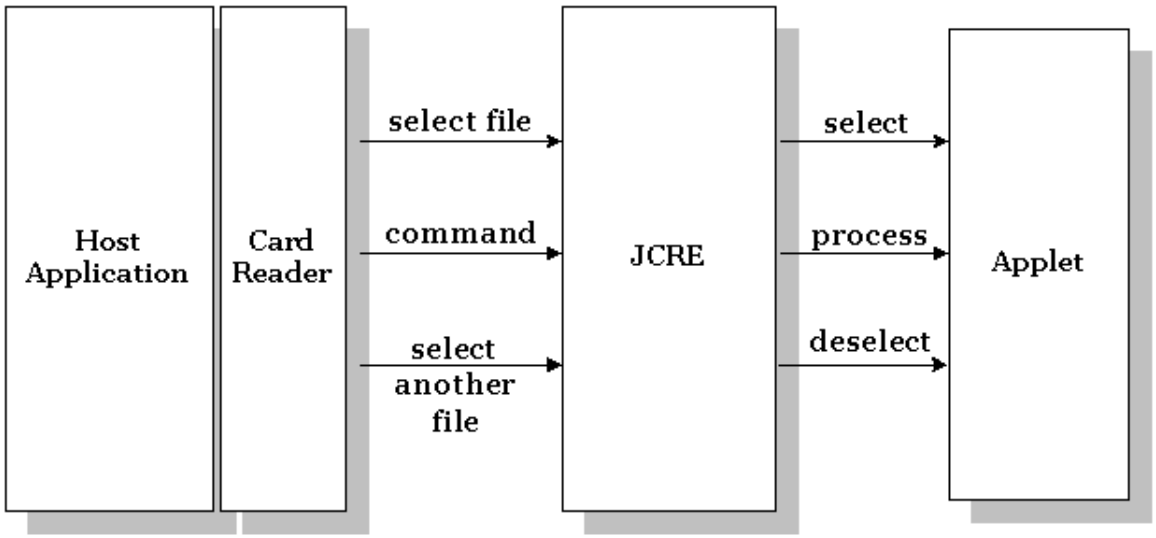

Figure 2-24: The java card selection process [44].

Figure 2-24 shows the selection process when using java card application. The card reader on the host side sends s select APDU with the java card applet ID to the JCRE. The JCRE then selects the corresponding java card application. The host side can then send command APDUs to the JCRE, which passes the commands to the applet and generates a response. Once the host application sends a new select APDU the JCRE will deselect the current java card application and select the new one.

2.6 Summary

This chapter has provided background information on technologies that are important to the project. It has introduced the RFID technology along with existing smartcard technologies. The NFC protocols have been explored, and the relationship between RFID, smartcard and NFC has been highlighted. It has further provided a study of the Java environment which will be the base for implementing the final solution.

3 Problem statement

This section introduces the specific problem at hand and gives an introduction to a pilot study the result of the thesis is meant to take part in. The section also pins out the parts that are crucial for the system to get the desired results.

3.1 Troms bus

Tromsbuss, a bus company in Troms county (Norway), is currently developing a contactless ticketing system for its busses. Tromsbuss is developing a pilot study for testing mobile phones as contactless tickets in collaboration with Telenor R&D, Troms county authority and Fara[2]. Figure 3-1 shows the system architecture where new components will be developed and put together with existing parts. The yellow components already exists, the blue ones will be developed by the project while the part going into the mobile phone is the result of this master thesis. The project is scheduled for trial in September 2006 with 100 test users. For an introduction to similar pilots conducted elsewhere see

A user will be equipped with a Nokia 3220 mobile phone with payment and ticketing shell [Annex B]. A user will download and install a MIDlet which can save electronic bus tickets received as SMS. Tickets are sent to the phone when a user purchases tickets online. When the user enters a bus, she/he will hold the phone up against an NFC reader for the ticket to be read and processed. Depending on the type of ticket, the remaining credit will be written back to the phone.

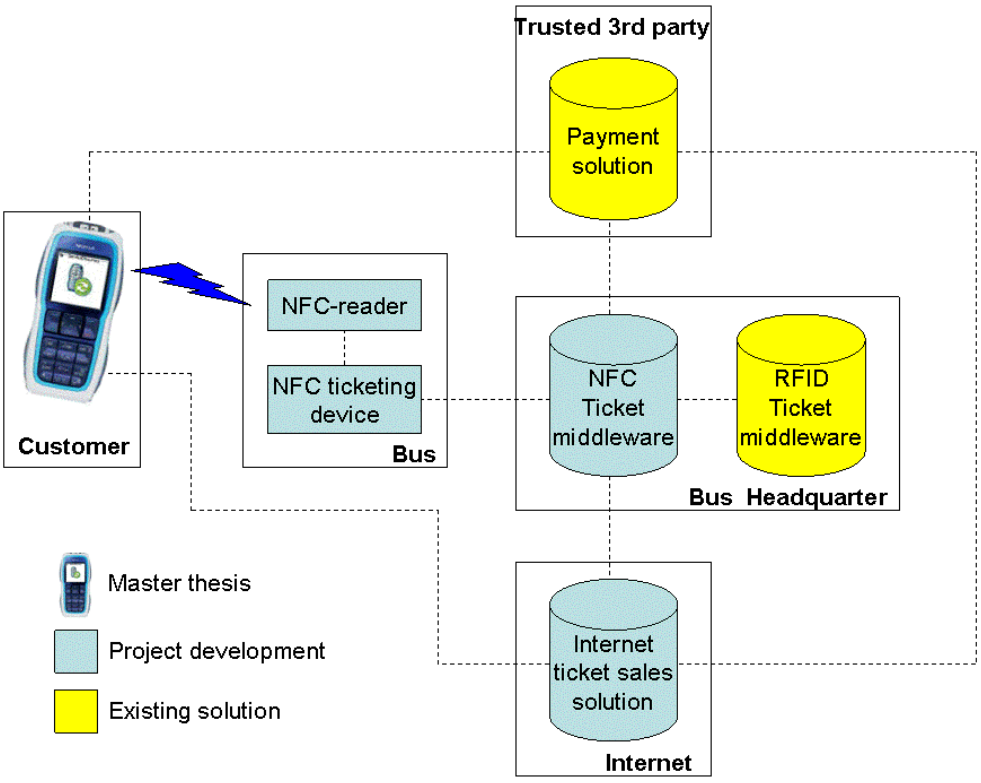

Figure 3-1: Tromsbuss electronic ticketing system overview.

[2] Fara - A Norwegian supplier of products and technology for automatic fare collection.

3.2 Environment

Figure 3-2 shows the operating environment for a mobile phone capable of NFC communication. The system is split into two different worlds, the mobile- and the contactless world. The mobile world represents the cellular phone and the GSM network. The contactless world depicts networked infrastructure and contactless reader capabilities.

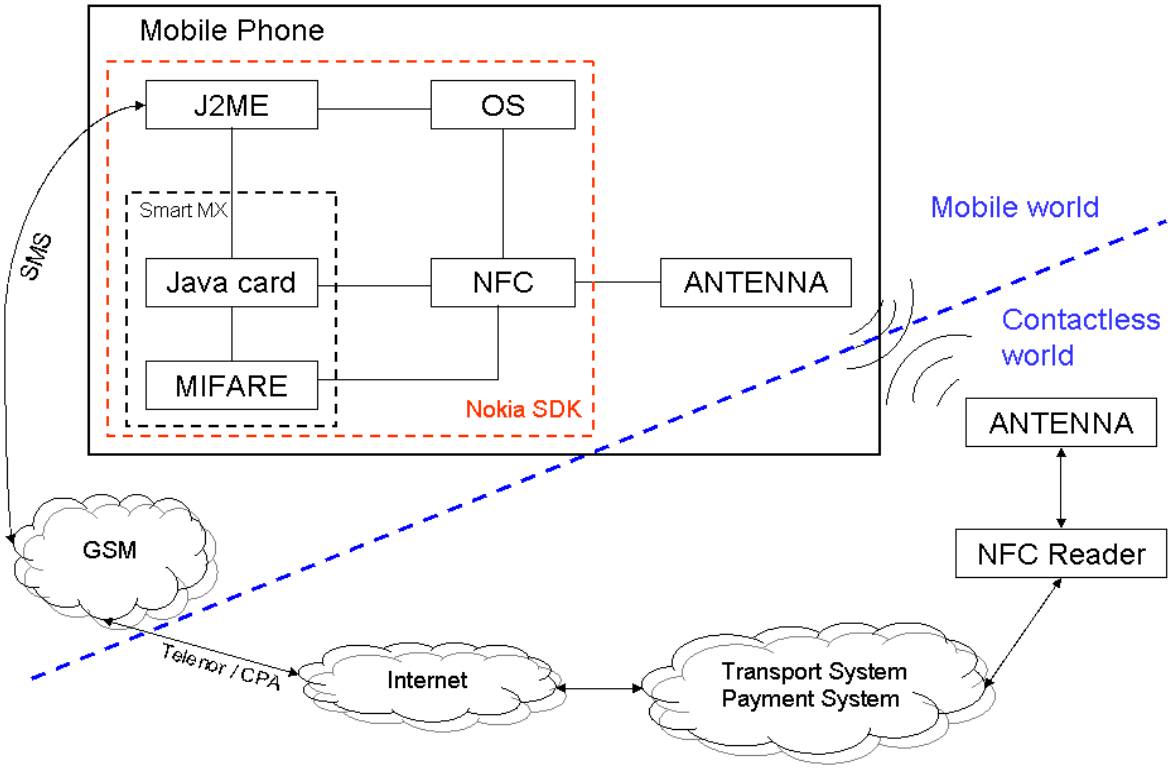

Figure 3-2: The operating environment for an NFC enabled Nokia mobile phone.

The figure is a high-level conceptual view of the involved entities and it is important to understand their functionality in order to identify with problem of this master thesis. The operating environment consists of the following elements:

Contactless world

Antenna	:	The antenna of a contactless reader. It generates a RF-field in order to read data from the NFC enabled phone.
RFID reader	:	The software that requests data from the mobile phone and processes it.
Transport / Payment System	:	These are the back-end systems that offer payments, connectivity between end-system devices, PSTN etc.
Internet	:	Represents the Internet and the service offerings it provides.

Mobile world

GSM : Represents the GSM network and the capabilities provided by GSM. The network can be operated by any network operator. The arrow to the mobile phone represents the capability of interchanging SMS, and the arrow to the Internet represents the connectivity between GSM networks and content providers operating over the Internet.

Mobile Phone : Represents a mobile phone providing regular GSM services and NFC capabilities.

- Nokia SDK : A software development kit for custom development of Java MIDlets. It also offers the ability to interact with the smartMX chip.
  - J2ME : The Java environment on the mobile phone. Its virtual machine offers a runtime environment that executes Java MIDlets.
  - OS : The operating system of the mobile phone. The operating system receives SMS and will start the addressed MIDlet.
  - NFC : The contactless communication scheme of the NFC standard. It can either operate as a reader or as a contactless smartcard.
  - Smart MX
    - Java Card : The Java environment on the smartcard included in the payment and ticketing shell. It holds a java card applet which is responsible for internal reads and writes to the MIFARE memory.
    - MIFARE : A smartcard with hardware co-processors offering secure memory storage.
- Antenna : The antenna on the mobile phone generating and sensing RF-fields in the case of contactless communication.

The link between the Internet and the J2ME entity depicts the ability of the Java environment to handle SMS from the GSM network. This ability is important because a Java MIDlet is started when a SMS is received destined to it. It will then extract and process the data the SMS contains.

The blue line represents a domain boundary between the mobile world and the contactless world. The link labeled CPA between the mobile world and the contactless world represents a gateway enabling content providers to deliver content to mobile network subscribers and at the same time bill the subscribers for using the service. The contactless link between the two antennas represents the new possibility of communication between the two domains using NFC.

The system shows the involved entities when a mobile phone is used as a device for contactless payment and ticketing. The following steps are involved in a typical ticketing transaction:

1. An order is placed and sent to the transport / payment system.
2. A SMS containing the electronic ticket is sent to the mobile phone.
3. The user receives the SMS and saves the ticket on the smartMX chip.
4. The ticket is used by presenting it to a contactless reader.

Step 3 above, where the user receives the ticket by SMS and stores it on the smartMX chip is a key for NFC enabled mobile phones to succeed as a contactless payment and ticketing device. The main advantage of using the mobile phone as a contactless ticket is the possibility to update the mobile phone OTA (Over The Air) without the need of being physically present in front of a ticketing machine. Users will have the possibility to purchase tickets anytime and anywhere.
One major task then stands out as critical for getting the system to work and succeed:

- How to get the data from the SMS onto the SmartMX card?

Addressing that problem will be the focus of the development in this project. If that problem can be solved one of the most critical parts of the described system would be accomplished. It will involve developing a MIDlet which can receive an SMS, extract data from it and write to the SmartMX chip. The problem will be explored in the view of Figure 3-2, and the Nokia Secure Chip SDK version 1.0 will be used in the development.

3.3 Summary

This chapter has provided a detailed explanation of the problem the project addresses. An overview of the operating environment has been provided along with an explanation of the different components of it. For an introduction to existing NFC pilot studies see *Annex C: Pilot studies.*

4 Analysis

This section introduces some scenarios and use cases that are meant to better understand the desired behavior of the system. Based on this the section comes up with requirements and system constraints.

4.1 Scenarios

The scenarios are a description of the system as seen by a user. It is a non-formal way of describing the system and is helpful in gaining insight to which behavior and functionality the system should offer. The following scenarios describe how a user would use a NFC enabled mobile phone for ticketing, payment and service discovery. This will help to gain insight in the problem domain and elicit requirements the system has to meet.

4.1.1 Install electronic ticketing system on the mobile phone

A customer wants to start using the mobile phone for electronic ticketing. The user has an NFC enabled mobile phone, but needs to acquire the electronic ticketing application. The user enters the ticketing company's website where there is a link for downloading the application. The user accepts the download and the application is automatically downloaded and installed on the phone. The user is then ready to buy tickets.

4.1.2 Buy electronic ticket

A user opens the electronic ticketing application and selects "buy ticket" in the main menu. The application automatically sends an SMS to the ticketing company, which sends back an SMS containing the electronic ticket. The ticket is received by the application which stores the ticket in the mobile phone. The user is now ready to use the ticket.

4.1.3 Use electronic ticket

In order to pass a gate a user needs to present an electronic ticket. The user holds the mobile phone up against the contactless reader at the gate. The reader reads the ticket, processes it and potentially writes an updated ticket back to the phone. The user is then granted access trough the gate depending on the presented ticket.

The above scenarios are the ones that are most relevant to the development process of this thesis. More scenarios are presented in"*Annex C: Pilot studies*", and these can help gain insight in more extended use of NFC in conjunction with the mobile phone.

4.2 Use cases

The use cases are more formal methodology means to show how the functionality the system offers meet some need of the user. They are not meant to indicate how the communication between participants of the system is, but rather a tool to identify the functionality the different actors have to offer.

There are three actors in this environment, a user, a seller and a contactless reader. The user represents a user with a NFC enabled mobile phone. The user can interact with the Seller who offers goods and services. The Contactless Reader represents the ticketing system were a user can use an electronic ticket. The system the user interacts with can be viewed as the mobile phone including its communication abilities and the services offered to it.

The use cases are all presented graphically. The use cases that involve interaction with the user are also presented with a written description to clarify the behavior.

4.2.1 General overview of functionality

Figure 4-1 gives an overview of the functionality of the system. A user can explore service offerings from Sellers by reading RFID tags (UC:Service discovery). The user interacts with the seller by making an order (UC:Make purchase) and receiving en electronic token (like an electronic ticket) trough a SMS (UC:Receive ticket). The user will then communicate with a RF reader when the ticket is used (UC:Use ticket). In the written specification of the use cases the Actor is the entity benefiting from system interaction.

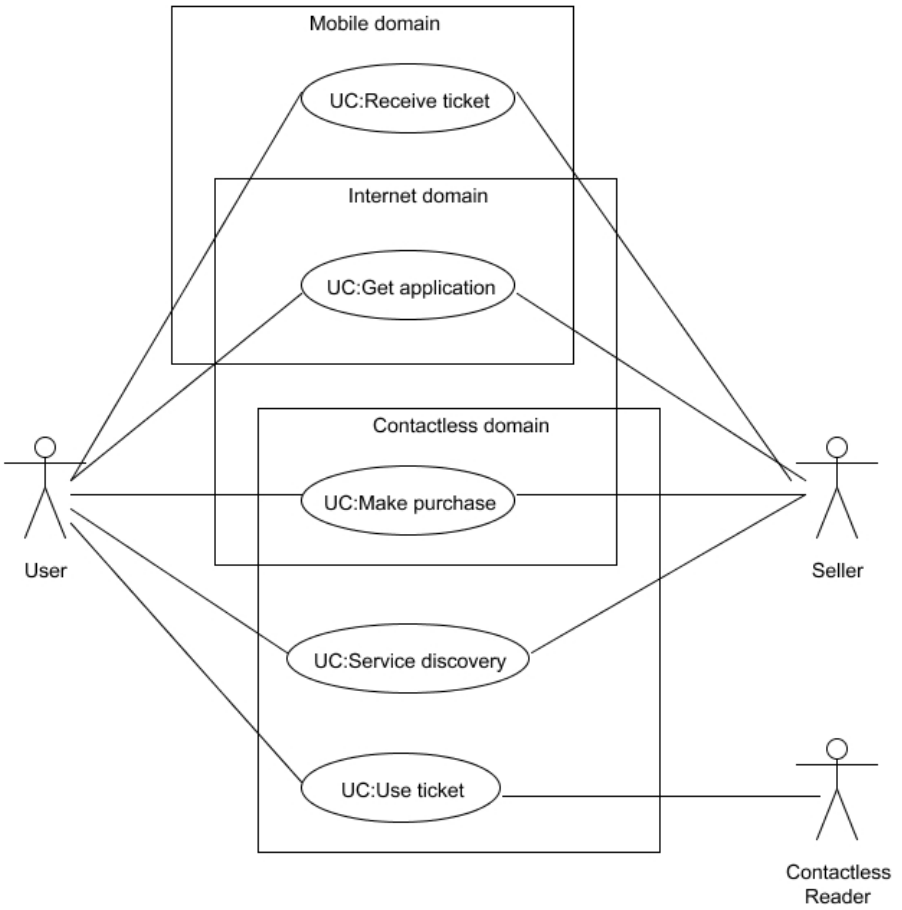

Figure 4-1: High level use cases of general functionality.

As can be seen from Figure 4-1 the system involves a number of distinct domains. It is important to understand the system behavior to come up with requirements for the system, and the interaction between domains definitely has impact on the behavior.

Use Case : **Use Case– "Service discovery".**
Actor : User.
Trigger : The user reads an RFID tag.
Pre condition : The tag must be programmed to offer a service.
Post condition : The user invokes the service offered by the tag.

User input	System response
1 Read tag by "touching" it with the mobile phone.	
	2 SMS initiative.
3 Execute system initiative.	

Table 4-1: System interaction - "Service Discovery".

Variations : 2A : Connect to URL using WAP browser initiative.
2B : Call a phone number initiative.
Related information : The system response is dependent on the tag's programmed service.

Use Case : **Use Case– "Get application".**
Actor : User.
Trigger : The user selects to download the electronic ticketing application.
Pre condition : The user must be connected to a website that offers the application.
Post condition : The application is installed on the phone and registered in the phone registry.

User input	System response
1 Download application.	
	2 Ask for permission to install.
3 Yes.	
	4 Installs the application.

Table 4-2: System interaction - "Get application".

Variations : 3A : No, then 4 will not be executed.
Related information : The amount of user interaction in the installation process depends on the security settings in the phone.

Use Case : **Use Case – "Make purchase".**
Actor : User.
Trigger : The user places an order.
Pre condition : There must be a seller offering a service.
Post condition : The user has completed the purchase and paid for the goods.

User input	System response
1 Place order.	
	2 Ask the user to pay for the goods.
3 Make payment.	
	4 Complete payment transaction.

Table 4-3: System interaction - "Make purchase".

Variations :
Related information : The order can be placed in any way, but the most relevant mechanisms for this system will be further elaborated in more detailed use cases.
The payment can be done using any mechanism, and the most relevant will be specified in more detailed use cases.

Use Case : **Use Case – "Receive ticket".**
Actor : User
Trigger : The user has completed a transaction where the seller will provide an electronic token (i.e. a ticket) by SMS as a receipt for the transaction.
Pre condition : The user has paid for a purchase.
Post condition : The ticket from the transaction is stored on the mobile phone..

User input	System response
	1 SMS received. Start registered application?
2 Yes	
	3 The ticket is stored to memory.

Table 4-4: System interaction - "Receive ticket".

Variations : 2A : The user answers no → This will result in the ticket not being stored in the secure chip.
Related information : The situation that arises in variation 2A will not allow for external readers to read the ticket. The ticket will however be queued and it will be passed to the ticket application the next time the application is started.

Use Case	**:**	**Use Case – "Use ticket".**
Actor	**:**	User
Trigger	**:**	The user wants to use the ticket stored in the mobile phone and "touches" a RF reader at the ticket check point.
Pre condition	**:**	The user has stored an electronic ticket on the mobile phone.
Post condition	**:**	The user has displayed the ticket electronically.

User input	System response
	1 Ticket request or Request for ticket
2 Contact between RF reader and mobile phone.	
	3 Read the ticket.

Table 4-5: System interaction - "Use ticket".

Variations	**:**	
Related information	**:**	A mechanism for limiting the number of times a ticket can be used is implemented at the reader side. After reading the ticket the ticketing system has to grant access according to the ticket. This is outside the scope of this project and is a matter for the respective ticketing and access systems to address.

4.2.2 Place order

The order can be placed using different technologies. Figure 4-2 shows how the user can place the order using either a computer or the mobile phone. There is nothing that prevents the user from placing an order manually in a store, but that would not involve using the system for ordering and is therefore not studied. Yet the user could use the system to pay for such an order (described in "*Annex E: Payment use cases*").

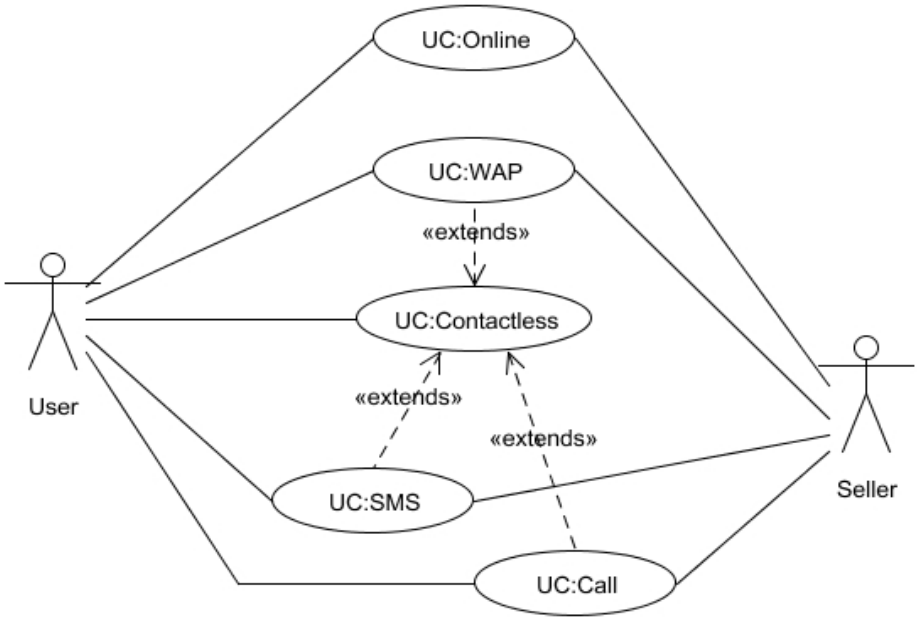

Figure 4-2: Placing the order can be done using different technologies.

If a user orders numerous tickets and wants them sent to different mobile phones it would have to be a specific functionality offered by the seller. In that case the user would have to provide the different phone numbers where the tickets should be sent. This is not considered an important aspect of the system as is therefore not specified in the use cases.

Use Case	**:**	**Use Case – "Online".**
Actor	**:**	User
Trigger	**:**	The user enters a web site and makes a purchase online.
Pre condition	**:**	There is seller providing service online and knows the port number of the application in mobile phones that can receive the ticket.
Post condition	**:**	The seller has information on where to send the ticket.

User input	System response
1 Check out from shop, chooses to receive ticket on SMS.	
	2 Prompt the user personal data, including where to send the ticket.
3 Submits the requested data	
	4 Proceeds to payment.

Table 4-6: System interaction - "Online".

Variations :
Related information :

Use Case : **Use Case – "Contactless".**
Actor : User
Trigger : The user reads a RF tag.
Pre condition : The tag is programmed to offer a service.
Post condition : The invoked service is started.

User input	System response
1 User reads a tag.	2 Provides the service stored in the tag

Table 4-7: System interaction - "Contactless".

Variations :
Related information : The service offered can be any of the services available in NFC service discovery or it can provide content stored directly in the tag.

Use Case : **Use Case – "WAP".**
Actor : User
Trigger : One of the following two events can trigger this use case:
1. The user actively enters a WAP page on the mobile phone and places an order.
2. NFC service discovery connects to an URL.

Pre condition : The user has a WAP enabled phone.
Post condition : The seller has received the order.

User input	System response
1 Place order and choose to receive ticket on SMS.	2 Proceeds to payment.

Table 4-8: System interaction - "WAP".

Variations :
Related information : WAP pages know the phone number of the mobile user and it is therefore not necessary to provide that information.

Use Case : **Use Case – "SMS".**
Actor : User
Trigger : The user sends an order by SMS.
Pre condition : The user is provided with the phone number which to send the order.
Post condition : The seller has received the order.

User input	System response
1 Sends order by SMS.	2 Receives order and proceeds to payment.

Table 4-9: System interaction - "SMS".

Variations :
Related information : The SMS can be automatically generated by NFC service discovery (which would provide the phone number), or the user can manually order tickets without touching a tag (i.e. "Send Buy-Ticket to 2500").

Use Case : **Use Case – "Call".**
Actor : User
Trigger : The calls a number to place an order.
Pre condition : The user is provided with the phone number to call.
Post condition : The seller has received the order.

User input	System response
1 Calls a number.	2 Automatically receives order and proceeds to payment.

Table 4-10: System interaction - "Call".

Variations : 2B： The system receives the orders manually.
Related information : The call can be automatically generated by NFC service discovery (which would provide the phone number), or the user can manually place a call to order.

4.2.3 Receive ticket

When the mobile phone receives an SMS with addresses port number, the operating system will ask the user if she/he wants to launch the application the SMS is destined to. Figure 4-3 shows the behavior, where UC:Receive SMS is the operating system handling the SMS. If the user denies starting the application, the ticket will be put in a queue by the application manager. When the Ticket application is started the next time the ticket is passed to the application. If the user chooses to start the Ticket application it will be launched (UC:Run application) and process the SMS.

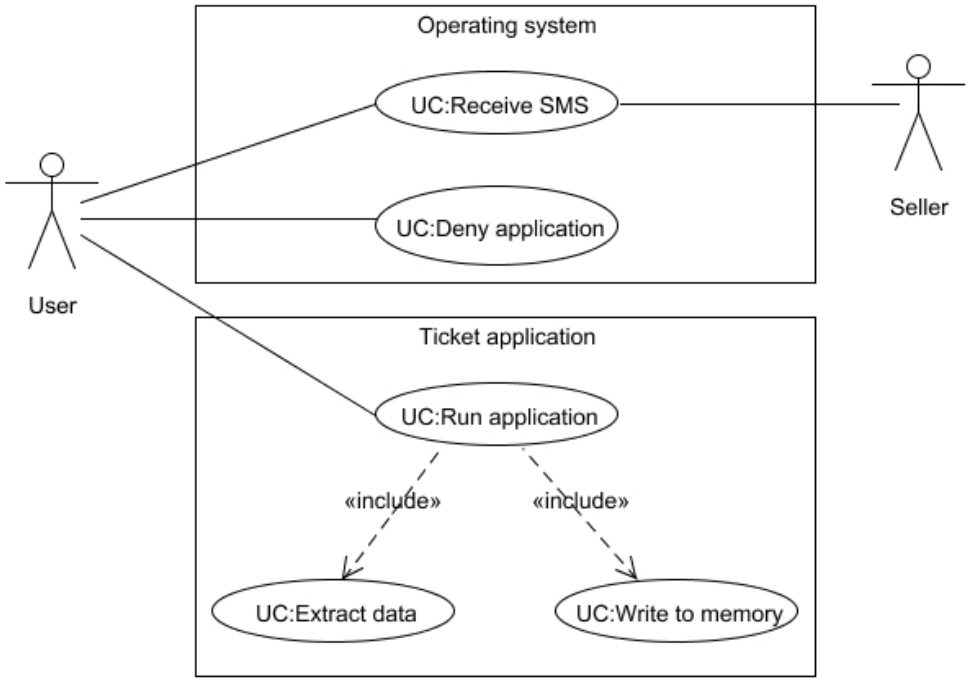

Figure 4-3: Receipt of SMS containing a ticket.

The UC:Extract data and UC:Write data will be launched by the ticket application the SMS is addresses to. They are automated processes and will not be described with a written use case because the user is not meant to provide any input.

Use Case	:	**Use Case – "Receive SMS".**
Actor	:	User
Trigger	:	The operating system receives an SMS addressed to an application.
Pre condition	:	The addressed application has registered its port number address with the operating system.
Post condition	:	The operating system starts UC:Deny application or UC:Run application.

User input	System response
	1 Receives SMS addressed to an application, start application?
2 Yes	
	3 The operating system starts the application (UC:Run application).

Table 4-11: System interaction - "Receive SMS".

Variations	:	2A : No → The operating system does not start the application in 3, thereby starting UC:Deny application
Related information	:	If the user chooses to start the application the process of extracting data (UC:Extract data) and writing it to memory (UC:Write to memory) will be automated.

Use Case	:	**Use Case – "Run application".**
Actor	:	User
Trigger	:	The ticket application addressed by the SMS starts.
Pre condition	:	The user chooses to start the application the SMS is addressed to.
Post condition	:	The ticket is securely stored in memory.

User input	System response
	1 Extract data from the SMS? (UC:Extract data).
2 Yes	
	3 Process and write data to memory? (UC:Write to memory).
4 Yes	
	5 Write to memory.

Table 4-12: System interaction - "Run application".

Variations	:	
Related information	:	The process should be as automated as possible. To achieve that the application has to be signed, and this is a matter of further studies. With a signed ticket application step 2 and 4 would not have been needed.

These use cases catch the functionality that lies behind the development process of this thesis. For the system to be used in a commercial environment there have to be more focus on the payment scheme. The uses cases for payments are not included here, but put in "*Annex E: Payment use cases*" because it is not the focus of this thesis.

4.3 Requirements and constraints

The system scenarios described in 4.1 and the use cases in 4.2 have outlined desired behavior of the system seen from a user's perspective. This section attempts to capture the requirements and constraints of the system, especially the mobile phone in order to realize the described scenarios and user cases.

4.3.1 Functional

Functional requirements are intended to capture the anticipated behavior of the system. There are numerous functional requirements to the proposed system. Table 4-13 summarizes the functional requirements for the system and gives a brief description of the different requirements.

Number	Requirement	Description
FR1	NFC support	The mobile phone must support contactless communication.
FR2	SMS capability	The mobile phone has to be able to receive incoming SMS.
FR3	Smart card functionality	The mobile phone must provide smart card functionality, i.e. security and communication
FR4	Ticket update	The system should be able to identify if a ticket has been used to prevent fraud.

Table 4-13: Functional requirements.

4.3.2 Non-functional

The non-functional requirements try to capture properties of the system that has to do with performance, quality or features that are not fundamental for the system to work. They are however very important because they are often properties that highly desired by the user and can help the system gain competitive advantage over other systems. Table 4-14 lists the non-functional requirements for the system.

Number	Requirement	Description
NFR1	Java implementation	The Ticket application should be implemented in Java, more specifically J2ME to ensure portability.
NFR2	Secure data handling	The data has to be stored in a way that they can not be compromised.
NFR3	Receive ticket fast	The SMS containing the transaction data has to be dispatched to the user quickly.
NFR4	User friendly	The graphical user interface has to be easy to understand.
NFR5	Few menu clicks	When receiving the SMS the user should only confirm that he/she wants to start the registered application.
NFR6	Error detection	If the mobile phone does not accept the SMS the entity originally sending the SMS should be notified quickly.
NFR7	Payment service	The user will have to subscribe to some payment service.
NFR8	Order tickets	The user should be able to easily order electronic tickets.
NFR9	Ticket range	The mobile phone should support a wide range of tickets.
NFR10	Ticket status	The user should be able to check the current ticket status.
NFR11	Memory	The mobile phone needs non-volatile memory to store the ticket so when the power is switched off.

Table 4-14: Non-functional requirements.

Table 4-15 presents some non functional requirements given a java implementation of the system.

Number	Requirement	Description
JNFR1	J2ME support	J2ME is needed to provide a runtime environment for the java MIDlet.
JNFR2	MIDlet	The MIDlet has to be registered with the operating system.
JNFR3	WMA support	The mobile phone needs the WMA to receive incoming SMS.
JNFR4	Known MIDlet port number	In order to send SMS to the MIDlet the address (port number) of the MIDlet has to be known by the sender.

Table 4-15: Constraints given a java implementation.

4.4 Interaction diagrams

The following section describes some interaction diagrams for the system. It is worth noting that the diagrams assume that the ticket application is already downloaded to the phone.

4.4.1 Communication diagrams

How to access the secure chip is described in [45] and its communication modes are presented in Figure 4-4. The MIDlet in the figure is a MIDlet for managing the secure chip.

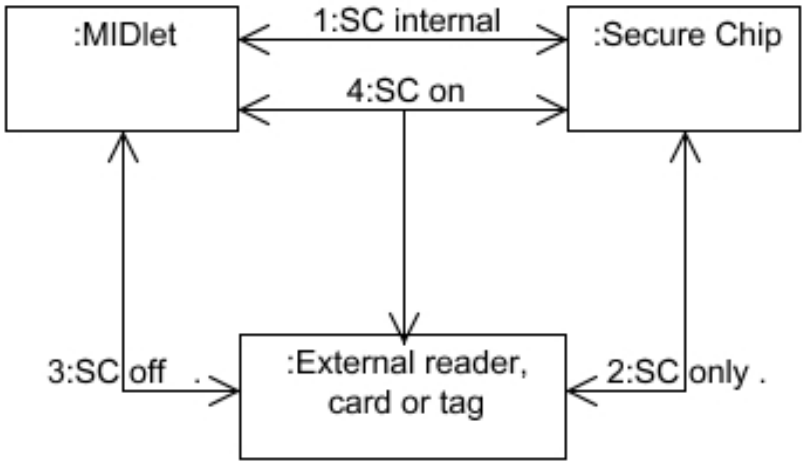

Figure 4-4: Secure chip communication mode [45].

These are the communication modes for the secure chip given in [45]:

1. SC internal : The secure chip is not visible to external devices and all data sent by the MIDlet is routed to the java card application on the secure chip.

2: SC only : The secure chip operates directly to an external device by NFC. The secure chip operates as ISO 14443-4 card and a MIFARE Classic 1k tag to external devices. The java card application on the secure chip can be used to access the MIFARE memory, or one can use ISO 14443 commands straight to the MIFARE memory.

3. SC off : Secure chip functionality is switched off and the device can communicate with an external RFID tag or NFC device.

4. SC on : The combination of 3.SC off and 2.SC only, meaning that the MIDlet can communicate with external targets and external targets can communicate with the secure chip and MIDlet by using NFC peer-to-peer.

The above mentioned communication modes are important because they provide security to transactions that need to be secure and the option of communicating with devices that you have no security association with. This is part of the desired flexibility of NFC.

Figure 4-5 shows a high level communication diagram of the system when a ticket is sent to the MIDlet. The ticket is sent to the phone as a binary coded SMS, addressed to a port where the MIDlet has registered on the phone. The application manager receives the SMS, and looks in its registry for the application listening on the addressed port. The application manager then starts the MIDlet, and the SMS is passed on to it. The MIDlet then processes the data and passes the ticket on to the Java card application. The java card application then writes the ticket to a MIFARE block. The status of the writing is returned to the java card application, which passes a save response to the MIDlet. The ticket is now stored on the secure chip.

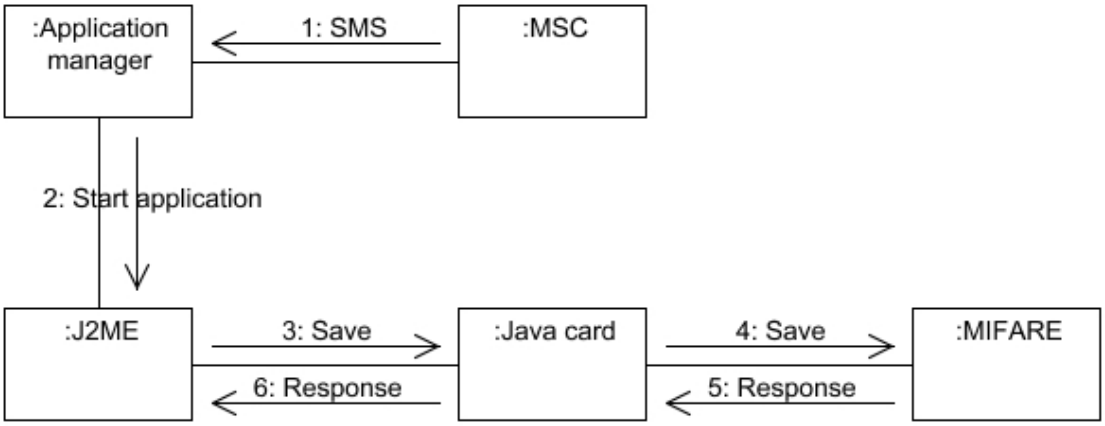

Figure 4-5: Communication diagram, save ticket.

Figure 4-6 shows an external read of the ticket, meaning the ticket is accessed from an external reader and not from the MIDlet. The contactless reader sends a read to the phone, which reads the block from the secure chip. The block is passed back to the reader. The read command to the secure chip can either use the java card application on the secure chip to access the MIFARE memory, or it can access the memory by passing ISO 14443 commands directly to memory.

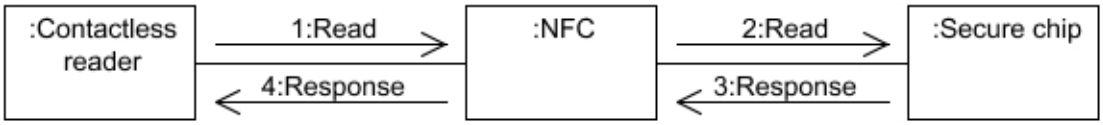

Figure 4-6: Collaboration diagram, read ticket.

A similar procedure will happen when a ticket is written back to the phone, except the "Read"s will be "Write"s instead.

The former diagrams are general and try to give an overview of important tasks in the system. The tasks are further elaborated in the section 4.4.2 as message sequence diagrams.

4.4.2 Message sequence diagrams

The following section describes the basic functionality of the system. The diagrams presented in this section describe successful functions, while diagrams presenting error handling are given in "*Annex H: UML diagrams*".

Figure 4-7 shows a message sequence diagram (msc) of the mobile phone receiving a SMS and starting the MIDlet. The binary coded SMS is sent to the phone and received by the application manager. The application manager looks up the MIDlet registered to the port where the SMS is addressed, and starts the MIDlet. Depending on security settings in the specific device the application manager may ask for user authorization to start the MIDlet.

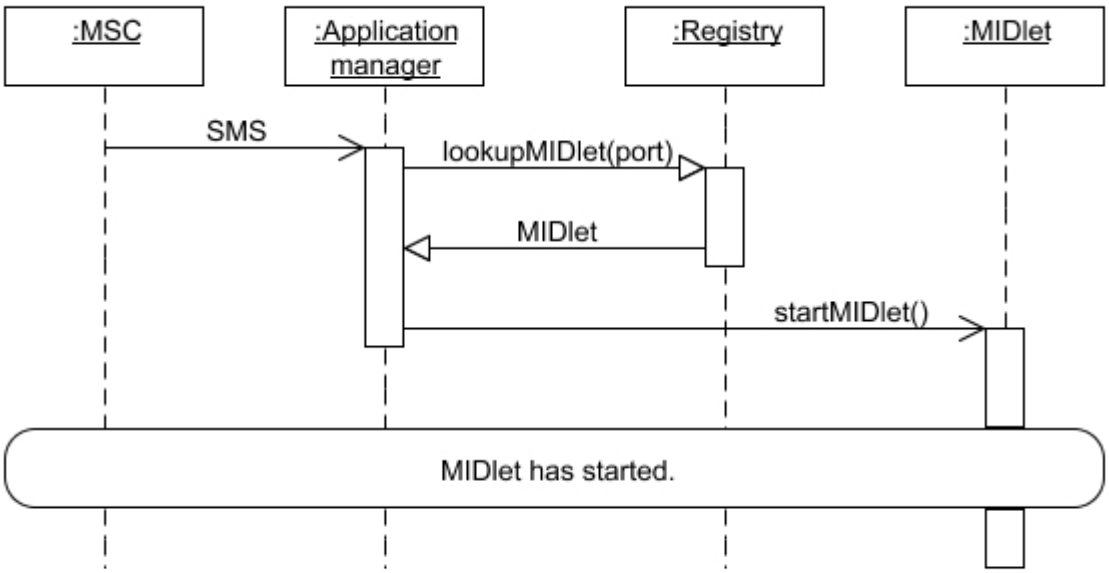

Figure 4-7: Message sequence diagram, start MIDlet when incoming SMS.

Figure 4-8 shows the msc of the MIDlet saving the ticket. The MIDlet gets the SMS from the application manager. The SMS is extracted and the MIDlet sends a selectAPDU to select the java card application. The java card application sends back a responseAPDU and the MIDlet is ready to read the ticket existing on the secure chip. This has to be done because the existing credit has to be added to the new. After reading from the secure chip (ref. Figure 4-9) the MIDlet processes the data. The MIDlet then sends a writeAPDU with the ticket to the java card application. The java card application saves the ticket and returns a responseAPDU.

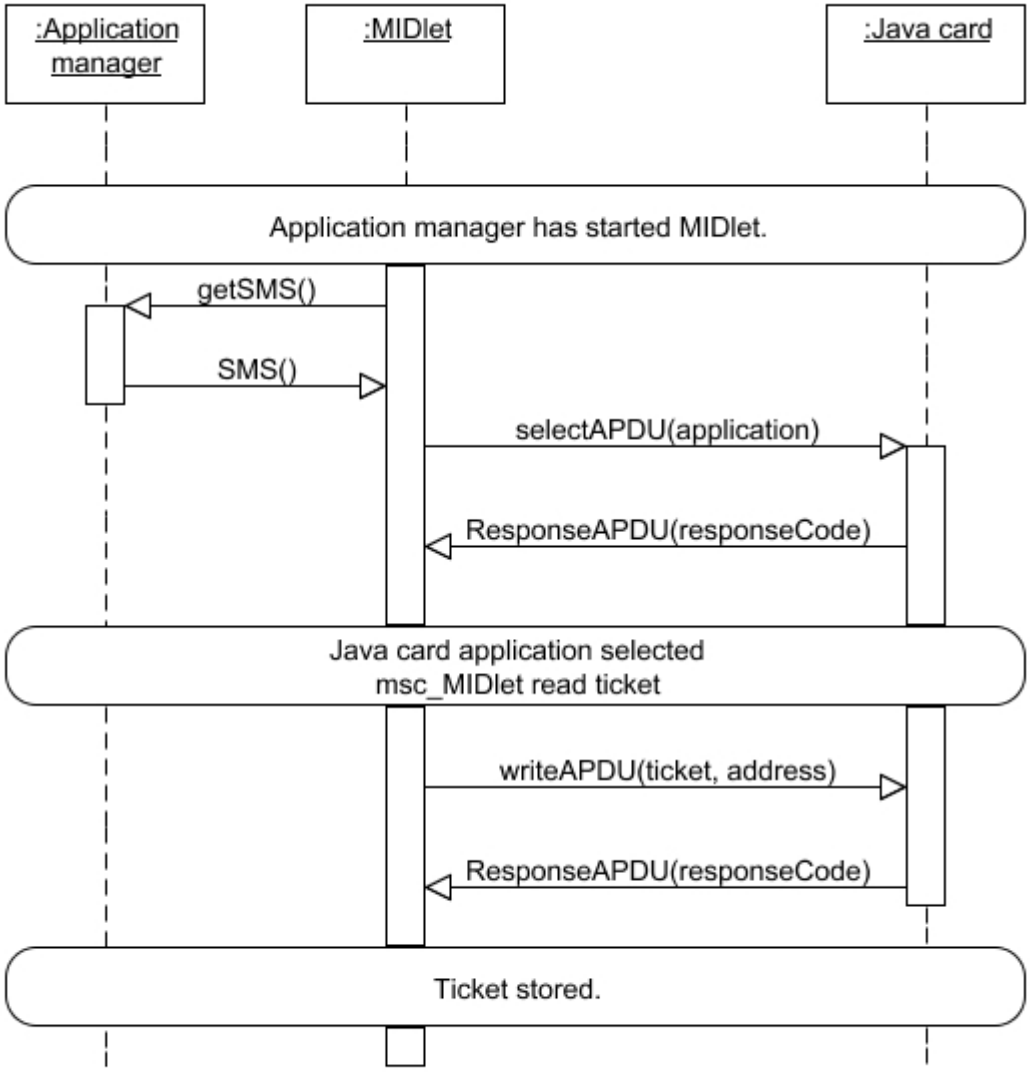

Figure 4-8: Message sequence diagram, save ticket

Figure 4-9 shows the MIDlet reading the ticket from the secure chip. This can either be initialized by the user wanting to check the current credit of the ticket as described in the optional box, or by the MIDlet receiving a new ticket. The selectAPDU(application) selects the java card application, while the response contains the success indicator. This is the same procedure as is done at the beginning of the save ticket msc. When the java card application is selected the MIDlet sends a readAPDU with the address it wants to read, and the java card application responds with the block and success indicators.

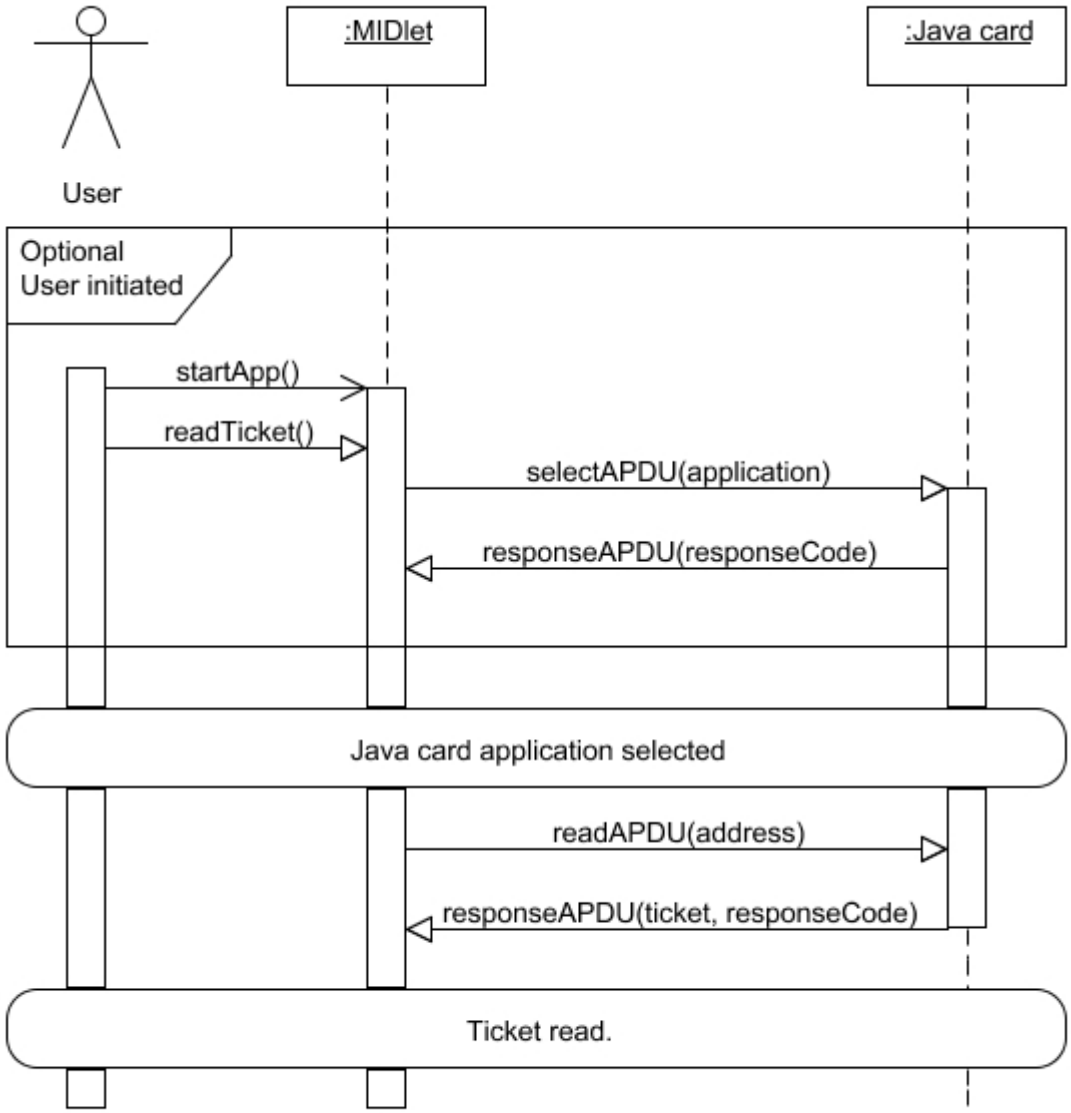

Figure 4-9: Message sequence diagram, MIDlet read ticket.

External reads and writes are necessary when the ticket is used in the ticketing system. The system will not make use of the java card application to write, as it only needs to send MIFARE APDUs to the secure chip. Since that functionality not will influence this design those mscs are put in "*Annex H: UML diagrams*".

5 Design

This chapter seeks to design a system based on the analysis of chapter 4. It addresses the different components involved and the communication between the different components.

5.1 Ticket structure

The design of the ticket is dependent on the desired ticketing model. There are in general two different strategies that can be implemented. One based on the ticketing systems as we know them today, where one can buy different types of tickets. This will result in different electronic tickets being downloaded to the phone depending on the specific purchase. The second strategy is based on an electronic wallet, where electronic money is used as tickets. This means that only electronic money or credits are transferred to the phone, and when used the system will subtract an amount from the electronic wallet. This results in only one kind of "ticket", but with different values depending on the purchased amount.

The Norwegian Public Roads Administration has written a handbook (handbook 206) with recommendations, requirements and guidelines for interoperable electronic ticketing systems [46]. The ticket outline of the handbook is given in *Annex G: Ticket from the Norwegian Public Roads Administration.* It introduces the concept of a product instead of a ticket, existing of rules of usage, pricing regulations and commercial rules. The focus of this thesis is developing a pilot for local use in one system, making the product outline of handbook 206 too excessive. This thesis will therefore use a ticket which will operate as electronic money, and be more lightweight than the outline of handbook 206.

Event though the ticket outline is simplified, incorporation of a product as described in handbook 206 will not have large implications on the MIDlet. The MIDlet does minimal computations with the ticket, and its focus is on writing the data into the secure chip. This job is not complicated by the content of the ticket, making the MIDlet easy adaptable to different ticket structures.

Figure 5-1 shows the structure of a ticket stored on the secure chip. It consists of seven bytes, where the first five bytes represent the last time a ticket was bought. The last two bytes represent the current value of the ticket. The ticket will be stored into a MIFARE block on the secure chip (ref. Figure 2-6). The ticket will be stored from byte 0 of the block, making it easy to add elements to the ticket later as there are 9 bytes left of the block which will not be used.

Byte #	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Charge	Last charge					Current credit	
Parameter	Day	Month	Year	Amount		Amount	

Figure 5-1: Ticket stored on the phone.

The ticket will be sent to the phone as a binary coded SMS. To make the MIDlet more adaptable to future extensions there will be two bytes at the beginning of the SMS identifying the company ant the type of ticket. Depending on the value of these parameters the MIDlet will process the SMS and save the ticket to the correct MIFARE block.

Byte #	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Parameter	Company	Type	Day	Month	Year	Amount	

Figure 5-2: SMS sent to the phone, containing a ticket.

5.2 Class diagram

Figure 5-3 shows a class diagram for the system.

TicketingSystemMIDlet is responsible for the lifecycle of the system, including listening and retrieving for incoming SMS. It is notified by the application manager when there is an incoming SMS to the MIDlet.

SecureChipHandler handles the communication with the Nokia payment and ticketing cover, including managing the communication mode of the MIDlet.

MifareWriter class is responsible for communication with the java card application which stores the ticket in the MIFARE memory.

TicketForm presents ticket data to the user depending on the selection in the main menu. The user can see the current credit of the ticket, or see the date and amount of the last purchase.

BusyAlert shows the progress of the work being carried out. It is used to inform the user when there is an SMS being processed or when the MIDlet is communicating with the NFC payment and ticketing cover

The HTTPConnection class opens a HTTP connection to a ticket server to order tickets. The payment of the order should be handled at the server side.

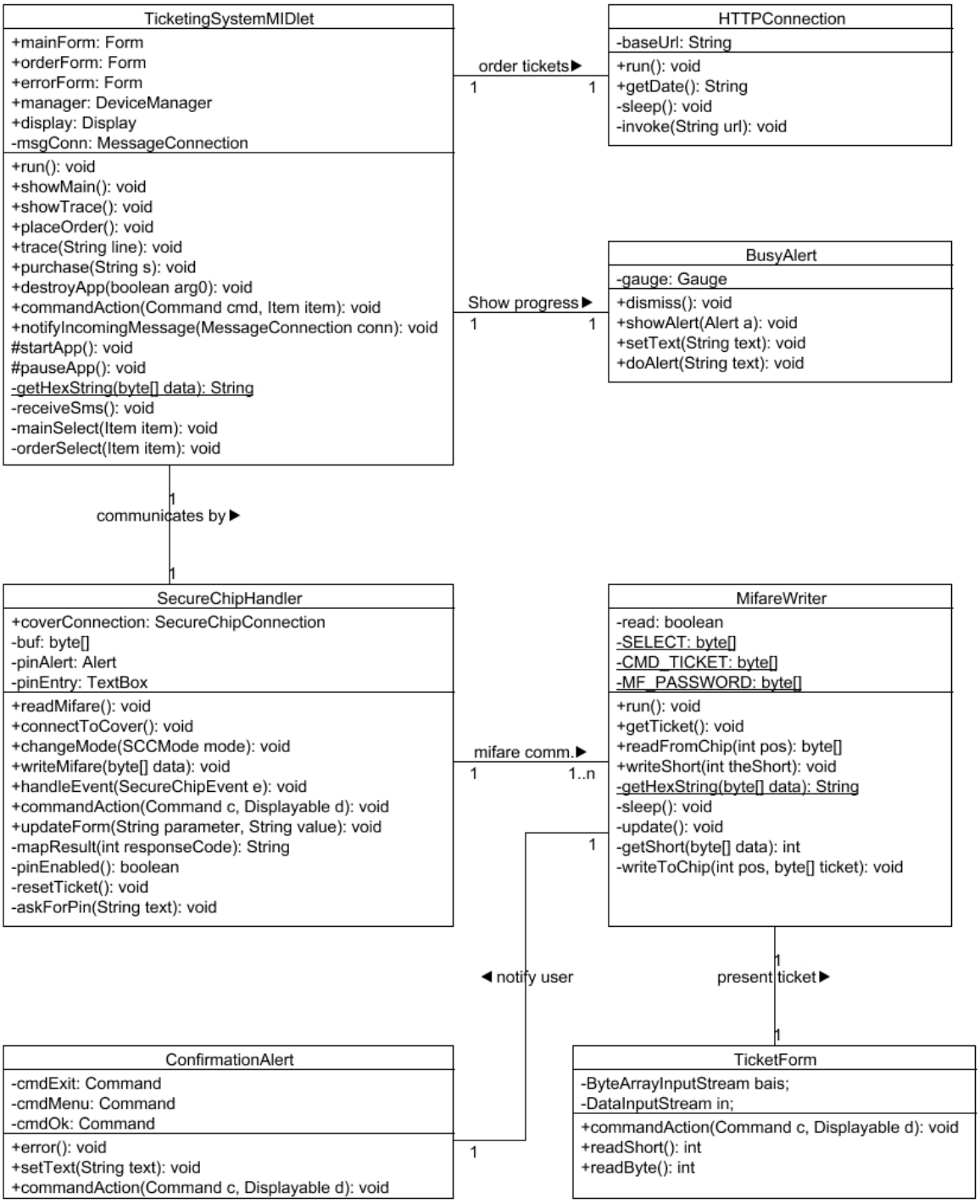

Figure 5-3: Class diagram.

5.3 APDU structure

The interaction between the MIDlet and the java card applet happens by exchanging application protocol data units (APDUs) conforming to ISO 7816-4 (ref. Section 2.2.1). There are three command APDUs that are used in the system. There is a selectAPDU, a writeAPDU and a readAPDU. In addition all of these command APDUs return a response APDUs. The responseAPDUs and the selectAPDU will not be elaborated as they are not a result of the design work of this thesis, but an existing part of java.

The writeAPDU and readAPDU both conform to ISO 7816-4. Figure 5-4 shows the writeAPDU, where the ticket is written to MIFARE block 61 on the secure chip. The MF_PW parameter is the password for writing to the block, while the data field contains the ticket. The java card application will respond with a responseAPDU, reporting the status of the instruction.

WRITE APDU								
# Byte	**1**	**1**	**1**	**1**	**1**	**1**	**8**	**16**
Parameter	CLA	INS	P1	P2	LC	ADR.	MF_PW	DATA
Value	0x80	0x02	0x00	0x00	0x19	0x3D	********	**********

Figure 5-4: Write APDU.

The readAPDU presented in Figure 5-5 requests a read of MIFARE block 61. The java card application will return a 16 byte MIFARE block along with status words in the response APDU.

READ APDU							
# Byte	**1**	**1**	**1**	**1**	**1**	**1**	**8**
Parameter	CLA	INS	P1	P2	LC	ADR	MF_PW
Value	0x80	0x01	0x00	0x00	0x09	0x3D	********

Figure 5-5: Read APDU.

These APDUs are the ones used to save and extract data from the secure chip from the MIDlet. They are passes to the java card application which performs the instruction given from the INS parameter. For external systems to access the secure chip they can either select the java card applet and use the above APDUs, or they can communicate by using MIFARE commands trough the NFC interface on the phone. This is of course only possible when the phone's communication mode allows access trough the NFC interface.

5.4 Summary

The design section provides a ticket structure that can be used for a prototype of the system. It has further developed a design that can be implemented into a prototype. The section also defined APDUs for communication between the involved entities.

6 Realization

The development process made use of various tools and equipment to accomplish the final result. This chapter introduces the tools and hardware that were used in developing the prototype

6.1 Hardware

The thesis makes use of mobile phone to run the MIDlet and a shell which enables NFC to the phone. In order to test the MIDlet an external reader has been used to communicate with the secure chip trough the NFC interface.

6.1.1 Nokia 3220 mobile phone

The Nokia 3220 mobile phone offers an environment to download and run MIDlets, and is also enabled for NFC communication when connected to a payment and ticketing cover (ref. Section 6.1.2). A detailed specification of the phone is presented in"*Annex F: Specification of Nokia 3220* .", while the most relevant data is presented in Table 6-1.

Developer Platform	Series 40 Developer Platform 2.0
Profile	MIDP 2.0
Configuration	CLDC 1.1
Messaging	Wireless Messaging API (JSR-120)
User interface	Nokia UI API

Table 6-1: Nokia 3220 details.

6.1.2 Nokia NFC shell for payment and ticketing

The Nokia payment and ticketing cover is a click-on shell which can be added on the Nokia 3220. It offers NFC communication and a smartcard chip (smartMX) for secure storage. The memory is organized as a MIFARE classic 1k card, which makes it accessible as a regular MIFARE card to external contactless readers. For a MIDlet on the phone to access the secure chip a java card application in the shell has to be used.

6.1.3 External reader

In order to test the application a SCR331-D1 [47] dual interface external reader from SCM Microsystems has been used. The reader connects to a computer trough USB and has been used to read and write to the secure chip. The reader uses standard ISO 7816 APDUs for communication.

6.2 Software

The thesis work has made use of numerous software tools to support the development process. Some of the tools were essential to develop, debug and test the system, while others were used to improve the quality of the code. This section introduces the tools and gives an introduction to their use. For further details on the tools, please refer to their distinct specifications.

6.2.1 Java Platform, Standard Edition

This offers the basic environment for development and deployment of java applications. Generally it offers a java runtime environment and a development kit (SDK) and serves as a platform for the other java technologies. It is necessary that the JAVA_HOME environment variable points to the root directory of the J2SE installation. In Windows XP/2000 it is sometimes necessary to add %JAVA_HOME%\bin to the path. For further information on the Java Platform, Standard edition see the Java homepage [48].

6.2.2 J2ME Wireless toolkit

The Java Platform, Micro edition was presented in section 2.4.1 and is the platform for developing and running MIDlets. The J2ME wireless toolkit (WTK) offers build tools, utilities and a device emulator to support the development process. The WTK has been used extensively in the development process as running the program on the emulator instead of on the real device is time saving. The utilities include the ability to send binary coded SMS to a MIDlet in the emulator, which is of great usage in debugging the MIDlet. It also offers over the air (OTA) provision of the MIDlet, saving time in the task of debugging that process. It is necessary for the J2ME_HOME environment variable points to the WTK installation directory. For further information on the J2ME Wireless toolkit see the J2ME WTK homepage [49].

6.2.3 Apache Ant

Apache Ant is an open source Java based build tool used for automation of software build processes. It uses standard xml to describe the build process and its dependencies. When installing Ant it is necessary to make some adjustments to the environment variables. ANT_HOME needs to point to the Ant installation directory, and the bin directory of the installation needs to be in the path. For further information on Apache Ant see The Apache Ant Project homepage [50].

6.2.4 Antenna

Antenna is distributed under the GNU Lesser General Public License (LGPL) and offers a set of Ant tasks suitable for developing java MIDlets. The tasks are suited to perform steps from the build process to deployment and running of MIDlets. It includes support for compilation, pre-verification, packaging and obfuscating the MIDlets along with other tasks. Antenna is installed by copying the antenna-bin-0.9.13.jar to the lib directory of the Ant installation directory. For more information on Antenna see the SourceForge.net homepage [51].

6.2.5 ProGuard

ProGuard is a free java obfuscator that was used together with Antenna. It removes unused classes, fields, methods and attributes in addition to optimizing the bytecode. This is important because of the limited storage space on MIDP devices. To install ProGuard it is necessary to copy the proguard.jar file into the bin directory of the WTK installation directory. . For further information on Proguard see its homepage at SourceForge.net [52]

6.2.6 Nokia Series 40 Developer Platform 2.0 SDK

This is the reference implementation of the Series 40 Developer Platform, which is the developer platform of Nokia 3220. It provides a development environment for J2ME applications, with an emulator that corresponds well to the Nokia 3220. This is constructive in

developing the graphical user interface. For further information on the SDK see the Forum Nokia homepage [53].

6.2.7 Nokia secure chip SDK 1.0

This SDK offers the ability to develop MIDlets that utilize the secure chip of the payment and ticketing cover (see section 2.4.3). The included emulator allows the MIDlets to connect and manage the communication modes of the secure chip. It also allows for communication with java card applications on the secure chip trough the exchange of command- and responseAPDUs. The SDK also support NFC and RFID communication, but this requires the Nokia NFC & RFID SDK to be installed. For further information on the SDK see [45].

6.2.8 Eclipse SDK

The Eclipse Project is an open source project that provides a platform and framework for application development. It has been used in this project as the integrated development environment (IDE). For further information on the eclipse project see the eclipse project homepage [54].

6.2.9 JCOP Tools for Eclipse

The JCOP tools for Eclipse is an Eclipse plug-in developed by IBM [55] to facilitate java card development. It is used in this project because it offers access to the JetZ API which offers access to MIFARE data structures, which is the case with the secure chip in the payment and ticketing shell. The JCOP tools should be available online from the IBM website [56], but at the time of this writing the tools were yet to be made available. The tools were provided to me after e-mail correspondence with an IBM representative in the USA who answered one of my attempts. The tools are also protected under IBM license, and due to American export restrictions the license key was sent by Federal express.

6.2.10 Kannel SMS gateway

Kannel is an open source WAP and SMS gateway providing the ability to send SMS trough an Internet browser. It is used in this project to send binary coded SMS containing the electronic ticket to a specified phone and port number. The Kannel gateway is implemented at Telenor r&d. To read more about Kannel see the Kannel homepage [57].

To send a SMS trough Kannel one writes a URL into the address window of a browser. The following URL sends an 18 byte (0xAA) binary coded SMS to “my phone number” on port 5555. The full kannel URL, phone number, username and password are hidden in this example for security reasons.

http://my.kannel.com/kannel/sendsms?username=****&password=****&to=%2Bmyphonenumber&udh=%06%05%04%15%B3%15%B3&coding=1&text=%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA%AA

6.3 Motivation for the technology choices

Implementation of a system always requires some technology choices to be made. The choice of programming language, hardware platforms and software tools has important implications on the final product. This section tries to describe the motivation behind the choices that have been made in this project.

6.3.1 Java

The choice of java as the programming language is based on several considerations.

First, it is the preferred programming language at the Norwegian University of Science and Technology (NTNU).

Second, it is widely adopted in the telecommunication industry and offers cross platform interoperability. It makes it possible to make an application that will work independently of the manufacturer of the handset. The application only has to meet the criteria set forth by the java virtual machine. As long as the handset supports java, the java virtual machine will make the MIDlet portable between devices from different manufacturers.

Third, the hardware manufacturers offer tools that ease the task of developing and debugging the applications.

It is also the case that java over the last years has gained considerable support among the mobile phone manufacturers. The number of phones with java support has grown extensively, and java can be considered a de facto standard for mobile phone application development. This makes it reasonable to believe that the solution will be supported by more services in the future and be a base for further exploitation.

6.3.2 Nokia mobile phone

The choice of Nokia mobile phone as hardware was necessary for the project to be realized. The Nokia 3220 with NFC payment and ticketing cover was the only phone on the market offering NFC and smartcard functionality included on the phone at the beginning of the project. As of the June 2006 there are other phones supporting the functionality, but they are not presented to the market yet. The choice of Nokia was therefore a matter of necessity to get access to enough handsets.

6.3.3 Software tools

The choices of software tools are many (ref section 6.2), but they are all based on the philosophy of using open source and freeware where it is possible. Eclipse IDE, Apache Ant, Antenna and ProGuard are all excellent tools for developing good java code. The Series 40 developer platform from Nokia is available free of charge, while the Secure chip SDK was provided to me by Telenor R&D. It is worth noting that these tools only where used to speed up the development process, and not to improve the quality of the code. That was performed by open source tools.

The choice of JCOP tools for Eclipse was given as a result of choosing the Nokia 3220 phone with NFC payment and ticketing cover. To get access to the secure chip from within a MIDlet on the phone, one has to use the JetZ API which is available trough the JCOP tools. It is worth noting that the JetZ API is used in the java card application, not in the MIDlet itself. It

was used in this project to study and debug the java card application installed on the secure chip.

6.4 Evaluation and testing

Testing of the complete system will have to wait until the "Troms buss project" has and is further in its development process. By that time this thesis work will be finished, so the focus of evaluation and testing in this section will be on the results of this thesis.

The MIDlet is developed only for the Nokia 3220 with NFC payment and ticketing. Due to lack of standardized APIs for secure chip access, the MIDlet makes use of specific Nokia APIs for communication with the NFC payment and ticketing cover. It uses block # 61 on the secure chip for storage, which equals sector 15 block 1 on the MIFARE classic 1K memory (see Figure 2-6).

The java card application on the secure chip is developed at Telenor R&D by Kjell Myksvoll. It makes use of the JetZ API which is part of the JCOP tools for eclipse from IBM, which gives access to MIFARE data structures from java card applications. This API had to be used because it is the only API supported in the phone for MIFARE data structure access.

I had a hard time getting hold of the JCOP tools for eclipse, and had to send tens of inquiries to IBM. IBM Norway could not help me with my requests and IBM in the USA did not answer several of my e-mail inquiries. Finally one of my many attempts got answered and the license was sent to me along with an Internet link to download the tools. The tools did not come with any documentation, and attempts to retrieve this from IBM have not been responded. Fortunately Kjell Myksvoll had some documentation available, but it is not clear if this is the full and latest documentation of the tools.

The MIDlet and its corresponding java card application are tested to an extent the time of this thesis has allowed and has shown parts of the system to work fine and parts to be unstable.

The OTA provisioning of the MIDlet has shown to be stable and has never failed. The MIDlet is downloaded and installed automatically, and the push registry registration works fine.

The reception of tickets is working fine both when the user chooses to install the ticket immediately and when the application manager has to queue the ticket. The writing to the secure chip has also proved to be stable and successful. The java card application writing to the MIFARE memory has been a part of a NFC demo at Telenor R&D and has therefore been tested for a longer period of time.

When the MIDlet reads data from the secure chip the java card application has suddenly sometimes generated error responses (6F 00 Status: No precise diagnosis). It is the same java card application that is responsible for writing to the MIFARE block that is responsible for reading from it. The read part has however not been used in the NFC demo at Telenor and is therefore not as well tested. By uploading the java card application again to the secure chip the read functionality has worked until it suddenly generates the same error again. The errors where generated even though the read APDU was not changed after successful reading operations, indicating that the weakness may lie in the java card part on the secure chip and not in the MIDlet. This suspicion was further strengthened by the fact that it was possible to read the MIFARE block by using an external reader and ISO 7816_READ_BINARY

command omitting the java card application. Due to this instable behavior I have been extremely eager to get the JetZ documentation from IBM to debug the java card application. They have however not been able to provide me with this.

The external contactless reader has also been used to test external communication with the phone. External communication which does not make use of the java card application has been stable and not generated errors.

The final demo of the system at Telenor R&D was successful with everything working fine. The system proved correct handling of tickets, including the read operations. However the problem of unstable read operations should be further studied before the system can be deployed to pilot users.

6.5 Summary

This section has provided an overview of the tools and hardware used in building the prototype. It has provided the motivation behind the technology choices that were made, and gives and evaluation of the prototype.

7 Summary

This thesis is the result of a thorough process that started the fall of 2005. It began with a brief study of the possibilities offered by NFC, and continued to look at the technology from the perspective of using the mobile phone as a smartcard for payment and ticketing. The process then continued with a thorough study of existing smartcard standards, communication protocols and technology for realizing a solution. The work continued with an analysis of the problem at hand and exploring the requirements such a system would have to meet. The analysis led to a design which has been implemented into a working prototype.

The development process of the thesis has followed the outline given in section 1.5. The evaluation and conclusion steps are carried out in regards to the result of this thesis. They are not the result of an evaluation and conclusion regarding the complete electronic ticketing system. The discussion and conclusion sections will still include elements from the complete system, but they can not be considered results from the design science process.

The described system and implementation can be analyzed in many ways, but two important ones are part of the thesis discussion. Section 8.1 provides a discussion of the technical solution and the choices that will guide further development. Section 8.2 looks at the system from a more commercial view. The system is based on an analysis of user needs, and the accuracy of these analyses most likely has impact on the user adoption and commercial success of the system.

8 Discussion

This section discusses the solution and findings of the thesis. It is divided into one part discussing the technical solution and a second part discussing the systems adoption and commercial success.

8.1 Technical solution

The MIDlet will be distributed by over the air provisioning, which makes the MIDlet easy to retrieve. There will be a link on a website that will offer the application. The link can be entered manually trough a web browser or by reading an RFID tag that is programmed to open the given URL. Once the link is selected the application manager on the phone will handle the installation procedure. The phone implementing the prototype requires the java card application to be uploaded by a contactless reader

8.1.1 Port number conflict

This above procedure to acquire the MIDlet is straightforward and easy to use. There are however numerous errors that can happen, like loosing the internet connection during the download, running out of battery during installation etc. These are errors that will be handled by the application manager on the phone and not influence the electronic ticketing system. However, during installation the MIDlet needs to register with the phone's registry. An error situation occurs when another MIDlet has already registered with the same port number the electronic ticketing system wants to register at. In that case the installation will not be able to complete.

The port number should definitely be solved before the system can be put into widespread use, and it calls for the system to start using installation notification. The JAD attribute `MIDlet-Install-Notify` can hold a URL where the MIDlet will send an installation status report. The messages are specified in the MIDP 2.o specification, and one of the responses is "911 Push registration failure". This solution requires the system to setup a server listening for these responses, but it will be necessary for the system to be useful in a commercial environment. The "Troms bus project" has been made aware of the possible error situation, but has not provided a specification for a solution to this problem. Such a solution generates a need to track which port number the customer listens to or make the users be aware of their port number when ordering tickets. For the MIDlet to adopt to installation notification it is only necessary to add the `MIDlet-Install-Notify` attribute with a value holding the server URL to the JAD file.

8.1.2 Ticket structure

The proposed ticket structure gives the user information on credit and last purchase. The ticket structure given in handbook 206 (*Annex G: Ticket from the Norwegian Public Roads Administration.*) was considered too extensive for the pilot study, but should be considered in a large interoperable commercial realization of the system. It allows for interoperability and unique identification of the different tickets.

8.1.3 Receive ticket

The problem description introduces the setting of the system with a mobile phone operating in the GSM network and making use of its SMS capabilities. The seller does not have to worry

about which network the user is operating in because the SMS should only confirm to the standard described in [58].

If the user chooses to not start the application when it receives a ticket, the SMS containing the ticket will be stored by the application manager. The ticket will then be passes to the application the next time it is opened.

8.1.4 Security issues

The `MIDlet-Push` attribute in the JAD file allows for restricting who is allowed to send SMS to the MIDlet. The third parameter of the attribute is a filter where one can specify which IP address that will be allowed to start the MIDlet. This parameter is currently set to the wildcard *, allowing any IP-address to send tickets to the phone. In an operating environment this should be changed to a limited number to deny other than the ticket issuer access to sending the phone tickets.

Another security issue is to add a cyclic redundancy check and use public key infrastructure to protect the tickets sent by SMS. This has not been the issue for this thesis, but should be considered before the system can be commercialized. However the security mechanisms of the contactless communication are studied in the background section. The secure chip offers an environment that is considered safe for payment transactions by mobile manufacturers.

8.1.5 User interaction

The automatic saving of a ticket received by SMS is certainly convenient to the user. The issue of securing this transaction by signing the MIDlet is however not fully explored in this document. An unsigned MIDlet will need user approval of the steps in storing the ticket to the secure chip. Although this user interaction is inconvenient to the user, it is not a flaw that jeopardizes the system. The automation process is a matter of security setting in the mobile phone and signing of the MIDlet.

The user is not queried to store the ticket when the application is already running. This is because the user already has provided authorization to run the MIDlet.

8.1.6 Java card applet installation

For the MIDlet to save the ticket it needs to communicate with the java card application on the secure chip. The installation procedure of the java card application is more troublesome for the user because the NFC payment and ticketing cover does not allow for installation of java card applications from a program in the phone. The java card application needs to be installed over the contactless interface directly to NFC payment and ticketing cover. This means that phones either have to be sold with an already installed java card application, or that the users will have to obtain it trough a retailer.

8.2 System adoption and commercial success

For a system to be adopted it must be seen as useful. In order to be considered useful a system generally has to offer some added value to a user. The goal of this project is to add value to the users of mobile phones by providing contactless ticketing and payment.

Contactless ticketing is meant to offer added value to ticketing customers. The transactions will be faster and require less user involvement, because it is only necessary to hold the ticket up against a contactless reader. Contactless ticketing is also less vulnerable to problems with worn and dirty tickets.

The solution in this document will add value to contactless ticketing customers, because it allows the mobile phone to operate as the ticket. There will be no need to carry contactless ticketing cards because it will be included in the customers' mobile phones.

It is also worth to notice that the mobile phone will offer one place to store numerous tickets. Today it is often necessary to use different tickets for different ticketing events. The mobile phone will operate as a ticket library, holding all of the user's different contactless tickets.

The distribution of tickets by SMS is also valuable to the user, because it demands less user involvement to obtain the tickets. The user can buy tickets trough SMS and will receive tickets trough SMS. This makes electronic ticketing more convenient, since they can obtain tickets anywhere and anytime.

The last part of the solution addresses the ability of a gate to read a ticket from the mobile phone. This is the part that actually allows the use of the mobile phone as a ticket. Because the secure chip in the mobile phone can operate as a contactless smartcard, existing systems will not notice the difference between a regular contactless card and a mobile phone. This makes the system adaptable to already installed contactless ticketing systems.

The Nokia 3220 phone with NFC payment and ticketing cover used in this prototype only allows the MIDlet to write to the secure chip trough a java card application. The MIDlet is provided to the user OTA, but the java card application has to be uploaded trough the NFC interface by a contactless reader. This makes it difficult to distribute the java card application to every NFC enabled handset and therefore making the end user adoption more difficult.

This is today the only way to provide the java card application to the mobile phone, hopefully the handset manufacturers will make it possible to provide java card applications OTA in the future.

The results of this thesis can be interesting to companies considering distributing tickets electronically to the mobile phones, because it will add a new sales and distribution channel to their business that could reduce their current administrative costs.

While regular contactless solutions need to be charged trough the contactless interface, this thesis enables the electronic distribution of contactless tickets. This means that ticketing companies can reduce their hardware and maintenance costs by having fewer sales places while still reaching a large customer base.

9 Conclusion

I believe the results of this thesis have the potential of adding value to users and businesses, but it is however necessary that the infrastructure is updated with contactless ticket readers. It is also necessary that handset providers include NFC and smartcard functionality in their handsets. The interest from credit companies, ticketing companies and other business sectors should be a great base for deployment of contactless payment and ticketing. The solution of including this in the mobile phone as depicted in this project is an attempt to add even more value to the users of these services.

9.1 Results

This document presents existing smartcard, RFID and NFC technologies. It gives an analysis of using the mobile phone in electronic ticketing and presents requirements such as system will have to meet. The design process includes designing a ticket structure as well as finding a solution for providing the functionality of the analysis section.

An implementation of the system is provided as a java MIDlet, allowing tickets to be sent to the phone and stored on a MIFARE structured smartcard. The MIDlet is meant as a prototype and should be further refined before possibly resulting in a commercial product.

The application will work as a demo of the mobile phone as a contactless ticket at Telenor R&D. It will also be further refined to serve as the MIDlet in the described pilot study with Troms buss.

9.2 Future work

The MIDlet should be further refined with security mechanisms and error handling. The functionality of buying tickets by sending SMS should also be implemented. Effort should be put into making it possible to install java card applications OTA.

Bibliography

[1] Magnus Egeberg: The mobile phone as a smartcard for payment and ticketing, 2005.

[2] Smart Card Alliance: Contactless Technology for Secure Physical Access: Technology and Standards Choices, October 2002.

[3] VeriFone: The Future is Contactless, Contactless Payment White Paper, September 2005.

[4] March & Smith: Design and Natural Science Research on Information Technology, 1995.

[5] David Arnott: Cognitive biases and decision support systems development: A design science approach (Working Paper No. 2005/01), Design Support Systems Laboratory, 2005.

[6] ABIresearch: Near Field Communications – Consumer Electronics and Digital Content RFID Market Opportunities, 2004.

[7] RFID – journal: The basics of RFID technology, http://www.rfidjournal.com/article/articleview/1337/1/129/, 2005

[8] Wal-Mart: Wal-Mart investor information, http://investor.walmartstores.com/phoenix.zhtml?c=112761&p=irol-irhome, 2005.

[9] Peter Longva: Master's Thesis – Security aspects of RFID based e-payment, 2005.

[10] RFID – journal: A Summary of RFID standards, http://www.rfidjournal.com/article/articleview/1335/1/129/, 2005.

[11] Mark Roberti: RFID – journal - Consensus Reached on EPC Gen 2, http://www.rfidjournal.com/article/articleview/1001/1/1/, June 24, 2004.

[12] United States Department of Defense: Supplier's Passive RFID Information Guide *Version 7.0*, http://www.productivitybyrfid.com/dod.asp, March 31, 2005.

[13] Office of the Deputy Under Secretary of Defense (Logistics and Material Readiness) http://www.acq.osd.mil/log/rfid/implementation_plan.htm, September 23, 2005.

[14] Steve Petri: An Introduction to Smart Cards, http://www.sspsolutions.com/solutions/whitepapers/introduction_to_smartcards, 2005.

[15] International Standard: ISO 7816 electronic identification cards, http://en.wikipedia.org/wiki/ISO_7816, June 2006.

[16] British standard: Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part1: Physical characteristics, BS ISO/IEC 14443:-1:2000, 2000.

[17] British standard: Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part2: Radio frequency power and signal interface, BS ISO/IEC 14443:-2:2001, 2001.

[18] British standard: Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part3: Initialization and anticollision, BS ISO/IEC 14443:-3:2001, 2001.

[19] British standard: Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part4: Transmission Protocol, BS ISO/IEC 14443:-4:2001, 2001.

[20] Philips: MIFARE - contactless Smart Card Ics, http://www.semiconductors.philips.com/markets/identification/products/mifare/#products, October 16, 2005.

[21] Philips: mifare Standard Card IC MF1 IC S50 Functional Specification revision 5.1, May 2001.

[22] Philips: mifare Standard 4 kByte Card IC MF1 IC S70 Functional Specification revision 3.1, October 2002.

[23] British standard: Information technology – Security techniques – Entity authentication – Part2: Mechanisms using symmetric encipherment algorithms, BS ISO/IEC 9798-2:1999, 1999.

[24] Philips: SmartMX platform features, Secure Smart Card Controller Platform, rev. 1.0, 24 March 2004.

[25] SONY: Contactless IC Card Technology <FeliCa>, http://www.sony.net/Products/felica/contents03.html#01, September 2005.

[26] SONY: “Octopus Card” in HONG KONG, http://www.sony.net/Products/felica/csy/hnk.html, November 2005.

[27] Vodafone: News Release, Vodafone K.K. to launch “Vodafone live! FeliCa”, http://www.vodafone.jp/english/release/2005/050920e_2.pdf, 20 September 2005.

[28] EMV: Integrated Circuit Card, Specifications for Payment Systems, revision 4.1, May 2004.

[29] EMVCo: About EMVCo, http://www.emvco.com/cgi_bin/0100_overview.pl, October 12, 2005.

[30] VISA: U-payment: Secure Visa Payment on the Move, 2005.10.12

[31] VISA: Press release: Visa Europe demonstrates world’s first EMV contactless cash replacement solution, 06 December, 2004.

[32] Philips: Identifying with China, http://www.semiconductors.philips.com/markets/identification/articles/success/s95/, September 2004.

[33] A. Vazquez: Common Specifications for Interoperability Smart Media Application Note, 18/02/2005.

[34] NFC Forum: About the NFC Forum, http://www.nfc-forum.org/aboutnfc/about_the_nfc_forum/, 10 November 2005.

[35] International Standard: Information technology —Telecommunications and information exchange between systems — Near Field Communication Interface and Protocol (NFCIP-1), ISO/IEC 18092:2004, 2004.

[36] International Standard: Information technology —Telecommunications and information exchange between systems — Near Field Communication Interface and Protocol -2 (NFCIP-2), ISO/IEC 21481:2005(E), 2005.

[37] ECMA-352: Near Field Communication Interface and protocol -2 (NFCIP-2), 1st edition, December 2003.

[38] Philips: Near Field Communication PN531-µC based Transmission module, Revision 2.0, February 2004.

[39] Sun Microsystems: Overview of the Java 2 Platform, http://developers.sun.com/techtopics/mobility/getstart/, 2005.

[40] Sun Microsystems: Datasheet Java 2, Micro Edition, http://java.sun.com/j2me/j2me-ds.pdf, 2005.

[41] Sing Li and Jonathan Knudsen: Beginning J2ME; From Novice to Professional, Third Edition, Apress, 2005.

[42] M. Debbabi, M. Saleh, C. Talhi and S. Zhioua: Security Evaluation of J2ME CLDC Embedded Java Platform _, in Journal of Object Technology, vol. 5, no. 2, March–April 2006.

[43] NOKIA: Nokia NFC Shell SDK, Version 1.0, Programmer's Guide, 2005.

[44] C. Enrique Ortiz: An Introduction to Java Card Technology – Part 1, http://developers.sun.com/techtopics/mobility/javacard/articles/javacard1/, May 29, 2003.

[45] Nokia: Nokia Secure Chip SDK 1.0, Programmer's Guide, Revision 1.0, 29. November 2005.

[46] Norwegian Public Roads Administration: Handbook 206, Specification for Interoperable Electronic Ticketing Systems, June 10, 2005.

[47] SCM Microsystems: http://www.scmmicro.com/security/secure_card.html, June 2006.

[48] Java Platform, Standard Edition: http://java.sun.com/javase/index.jsp, June 2006.

[49] J2ME WTK: http://java.sun.com/products/sjwtoolkit/download-2_3.html, June 2006

[50] The Apache Ant Project: http://ant.apache.org/, June 2006.

[51] Antenna: http://antenna.sourceforge.net/, June 2006.

[52] ProGuard: http://proguard.sourceforge.net/, June 2006.

[53] Forum Nokia: http://www.forum.nokia.com/main/0,6566,033,00.html#overview, June 2006.

[54] The Eclipse Project: http://www.eclipse.org/, June 2006.

[55] IBM Zürich Research Laboratory: http://www.zurich.ibm.com/jcop/, June 2006.

[56] IBM: http://www-306.ibm.com/software/wireless/wecos/, June 2006.

[57] Kannel: http://www.kannel.org/, June 2006.

[58] 3rd Generation Partnership Project: Technical Specification Group Terminals; Technical realization of the Short Message Service (SMS), 3GPP TS 03.40 V7.5.0, 2001-12.

Annex A: ISO 7816-4 field values.

The following tables are copied from [44].

CLA value	Instruction class
0x0n, 0x1n	ISO 7816-4 card instructions, such as for file access and security operations
20 to 0x7F	Reserved
0x8n or 0x9n	ISO/IEC 7816-4 format you can use for your application-specific instructions, interpreting 'X' according to the standard
0xAn	Application- or vendor-specific instructions
B0 to CF	ISO/IEC 7816-4 format you can use for application-specific instructions
D0 to FE	Application- or vendor-specific instructions
FF	Reserved for protocol type selection

Table 1: ISO 7816-4 CLA values.

INS value	Command description
0E	Erase Binary
20	Verify
70	Manage Channel
82	External Authenticate
84	Get Challenge
88	Internal Authenticate
A4	Select File
B0	Read Binary
B2	Read Record(s)
C0	Get Response
C2	Envelope
CA	Get Data
D0	Write Binary
D2	Write Record
D6	Update Binary
DA	Put Data
DC	Update Record
E2	Append Record

Table 2: ISO 7816-4 INS values.

Annex B: MasterCard PayPass questions.

This is the response on some questions I e-mailed MasterCard regarding their PayPass technology:

Hello Magus,

Please find below some answers to your questions.

All the best,

Jonathan

```
- Which contactless standard is Paypass built on / compatible with?

PayPass is built on the ISO 14443 standards. The PayPass cards may be
either type A or type B, and the terminals must support both type A and
type B for interoperability.

- What is EMVs position in this matter?

There are two parts to the PayPass standard. The first is the transport
protocol, which is an implementation of ISO14443.
The second is the application which runs on top of this standard.
MasterCard has defined two profiles for this application, one which is a
magnetic stripe profile, and the other which is an M/Chip profile. M/Chip
is the MasterCard application for EMV. The magstripe profile is for use in
environments where the acceptance infrastructure is magnetic stripe, while
the M/Chip profile is for environments where the acceptance infrastructure
is EMV based.

- Does Mastercard plan to enable NFC comminication in their contactless
solution if it has not done so already?

The primary concern for MasterCard is that that the NFC protocols are
compatible with PayPass, in particular that an NFC device can emulate a
card. In this case the transaction would occur using the PayPass ISO14443
protocols. Our involvement is to ensure that there is compatibility between
NFC and PayPass, so that NFC devices can also support PayPass.

In the longer term, there is the possibility for NFC devices to act as
readers for PayPass - however this is not something we would expect to see
in the short term.
Running the PayPass application on top of the NFC protocols is something
which is theoretically possible. However the current generation of PayPass
readers do not support the NFC protocols. In payment systems the acceptance
infrastructure is more costly to upgrade and tends to set the speed at
which new technology can be introduced.
```

Hope this helps, let me know if you've any follow up questions.

All the best,

Jonathan

```
--
Jonathan Main
Director, Wireless Standards,
```

Advanced Payment Solutions,
MasterCard UK,
47-53 Cannon Street, London EC4M 5SH, UK
Email: jonathan_main@mastercard.com
Ph: +44 20 7557 6812 Fax: +44 20 7557 6912 Mob: +44 7714 145 717

Annex C: Pilot studies

There are currently numerous pilots on NFC around the world. The pilots are aimed to give the companies valuable feedback before a full scale adoption of the technology can take place. It is interesting to notice that the pilots include handset manufacturers, mobile service providers, public transportation entities and also department stores. This large involvement of different business sectors can give a considerable boost to the speed at which NFC will be adopted.

German mass transit

The German Mass transit association (VDV - Verband Deutscher Verkehrsunternehmen) represents 90 percent of the German mass transit market and has developed an electronic ticketing platform exploring contactless smartcard technology[3]. It allows for cashless- and electronic ticketing and automated pricing for individual journeys. The system offers interoperability nationwide in Germany and seeks to gain it across Europe. The system is piloting in Frankfurt.

In the city of Hanau east of Frankfurt the public transport system is involved in it second pilot for electronic ticketing. The first pilot involved using contactless smartcards and this current pilot lets the users pay their trips with NFC equipped Nokia mobile phones. The users swipe their cell phone over a reader as they enter and exit the bus, the price for the journey is calculated and the customer receives a bill for all trips at the end of the month.

The customers are satisfied because they do not have to carry anything but their cell phone. The entering and exiting of the bus is faster and they only get charged according to how much they travel.

The transport company is satisfies because the driver no longer have to worry about carrying change, and they expect to make large savings from eliminating the paper-based system. They can also use the data collected from the fares to increase the efficiency of the route organization. The system also has the possibility of charging passengers not presenting a ticket when entering a bus on the spot in case of ticket inspections. They are hoping this will help to reduce fraud coming from the lack of ticket readers with physical barriers at the entrance and exist of a bus.

The system in Hanau will later be made compatible with the VDV system.

[3] Contactless News: German mass transit goes contactless, and trials NFC, http://www.contactlessnews.com/library/2005/11/30/german-mass-transit-goes-contactless-and-trials-nfc.php, November 30 2005.

The city of Caen

The city of Caen, France, is currently the stage of a pilot for NFC enabled services[4]. The France Telecom, network operator Orange and Samsung are among the companies cooperating to offer the services. There are currently four services available:

1. Retail application for NFC payments at a supermarket and a department store. The number of stores will continue to increase and eventually a cashless payment scheme will be developed in partnership the consumer credit division of the LeSer Group.

2. Car park access where people can pay for parking by touching the entrance booth with their NFC enables phones.

3. Tourist service where people can get information about the site by touching NFC enabled information points.

4. Active posters will make it possible to download mobile content by touching the embedded tags in the posters. The content can be ringtones, wallpapers or even the local bus schedule.

The pilot is conducted on 200 randomly chosen citizens.

[4] Contactless News: Near Caen France hosts world's premier NFC trial with mobile phones enabling host of contactless applications, http://www.contactlessnews.com/library/2005/11/23/caen-france-hosts-worlds-premier-nfc-trial-with-mobile-phones-enabling-host-of-contactless-applications/, 23 November 2005.

Annex D: Scenarios

Traveling:

Alice is shopping for a vacation online and finds the place she wants to go. She makes all the travel arrangements online, including flight and hotel reservations. She then receives an electronic airline ticket on her cell phone by SMS from the travel agent. She chooses to receive the keys to her hotel room by SMS when the reservation activates. At the airport Alice only has to touch the check-in counter with her phone for her reservation to get activated, and she will be able to check in her luggage. The receipts for the luggage tags will be sent to her by SMS and she just has to leave the baggage at the baggage drop-off. At the gate she passes the entrance to the plane simply by touching the ticket reader with her phone. At the time of arrival at the hotel she has received the room key by SMS along with information about the room number. She can therefore go straight to her room without standing in line to check-in and receive keys. During her stay she can receive information about the sites she is at by touching active posters with her phone. At the end of her stay the room key will automatically get deactivated and she can electronically sign the bill of the stay by touching an NFC enabled payment device at the exit or checkout counter.

Electronic Payment

A customer is about to pay for a purchase in a store. The customer decides not to pay with cash, but with a NFC enabled mobile phone. There is a payment terminal with a RF reader at the check-out register. The clerk asks the customer if she/he wants to pay by credit, debit or prepaid. The customer chooses how to pay and “touches” the reader with the mobile phone to make the transaction. The payment system that the RFID reader is connected to processes the transaction and charges the customer.

Service Discovery

A user is at a bus stop and wants to know when the next buss is scheduled to be there. The bus stop has an active poster where the user can read a tag to invoke the programmed service. In this case the service can be offered in three ways:

1. The tag will make the phone send a SMS requesting the time of the next scheduled bus at the current location. The user will receive a SMS containing the next scheduled stop.
2. The tag will make the phone connect to an URL containing the bus schedule for the appropriate bus stop.
3. The tag will make the phone call an information desk that can answer the question.

The actual service invoked will depend on the service programmed in the tag.

Annex E: Payment use cases

These use cases are included because they are important for commercializing the system. Due to lack of time and resources the payment issue has not been the focus of this thesis, but they are important in understanding the different payment solutions that can be incorporated into the system.

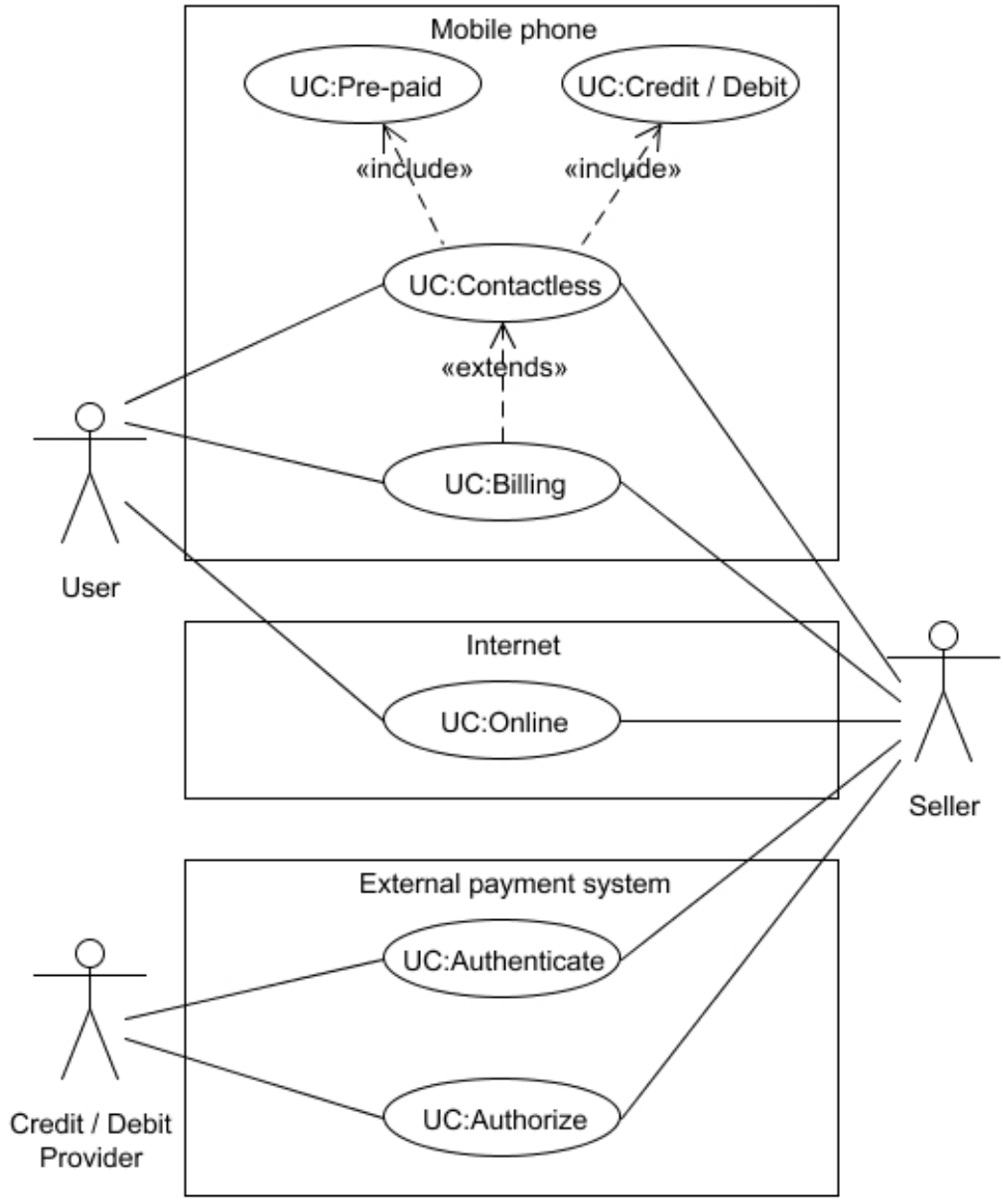

Figure 1: Making payment.

The external payment system is existing infrastructure and systems between sellers and credit / debit providers. This includes the mobile service provider in the instances where a payment is to be charged on the phone bill. The system carries out authentication (UC:Authenticate) and authorization (UC:Authorize) of the customers (users), and there are a number of businesses offering this today. Analysis of those systems is outside the scope of this project, and their functionality and service offerings will be utilized without further studies.

The Internet represents the ability to pay electronically over the Internet. No further investigation of these systems will be provided, but they are represented because they offer a mean for a user to pay for a purchase and receive an electronic ticket to the mobile phone

Use Case : **Use Case – "Contactless payment".**
Actor : User
Trigger : The user "touches" an RFID reader.
Pre condition : The user has subscribed to a payment service accepted by the seller.
Post condition : The appropriate payment scheme will be carried out.

User input	System response
2 Choose payment scheme and "Touch" RFID reader.	1 Ask for desired payment scheme (pre-paid, credit / debit, billing). 3 Invoke UC:Pre-paid, UC:Credit / Debit or UC:Billing.

Table 1: System interaction - "Contactless payment".

Variations :
Related information : If the seller does not offer numerous payment schemes the user will not be asked to choose and only has to "touch" the RFID reader.

Use Case : **Use Case – "Credit / Debit".**
Actor : User
Trigger : Invoked by RFID reader.
Pre condition : The user has the necessary account- and personal data stored on the mobile phone to complete a transaction.
Post condition : The purchase is paid for and the seller will provide the user with a SMS containing tickets where that is applicable for the purchase.

User input	System response
2 Enter PIN	1 Extract data from the mobile phone. Ask the user for PIN code. 3 Authenticate user (UC:Authenticate) and authorize purchase (UC:Authirize). 4 Complete transaction.

Table 2: System interaction - "Credit / Debit".

Variations : 3B : UC:Authenticate or UC:Authirize fails → Step 4 will not be carried out.
Related information :

Use Case : **Use Case – "Pre-paid".**
Actor : User
Trigger : Invoked by RFID reader.
Pre condition : The user has the necessary account- and personal data stored on the mobile phone to complete a transaction.
Post condition : The purchase is paid for and the seller will provide the user with a SMS containing tickets where that is applicable for the purchase.

User input	System response
2 Enter PIN	1 Extract data from the mobile phone. Ask the user for PIN code. 3 Authenticate and authorize user for purchase. 4 Debit amount and write new balance to mobile phone. 5 Complete transaction.

Table 3: System interaction - "Pre-paid".

Variations : 1B : No need for PIN → Continue on from step 4.
Related information : The need for PIN is a matter of choice depending on the specific pre-paid service used.
The authentication and authorization in step 3 is dependent on the specific service. The PIN can for instance be used as a key to allow debiting the balance.

Use Case : **Use Case – "Billing".**
Actor : User
Trigger : Billing can be triggered by the following actions:
1. Invoked by RFID reader.
2. Purchase made by SMS (UC:SMS).
3. Purchase made by WAP (UC:WAP).
4. Purchase made by call (UC:Call).

Pre condition : The user has subscribed to a payment service accepted by the seller.
Post condition : The purchase is paid for and the seller will provide the user with a SMS containing tickets where that is applicable for the purchase.

User input	System response
2 Enter PIN	1 Extract data from the mobile phone. Ask the user for PIN code. 2 Authenticate user (UC:Authenticate) and authorize purchase (UC:Authorize). 3 Complete transaction.

Table 4: System interaction - "Billing".

Variations : 1B : No need for PIN → Continue on from step 2.

Related information : Most likely there will be no need for PIN using trigger 2. The need for PIN is a matter of choice depending on the service offering billing.
The billing scheme can be a system where the user has a pre-paid account, but the account data is not stored on the mobile-phone as in UC:Pre-paid. The amount is then considered to be “billed” to the pre-paid account.
The billing can also involve putting the purchase on a bill, i.e. the phone bill, and receive a joint bill for mobile phone use and purchases. Step 2 will then involve communicating with the user’s service provider.

Annex F: Specification of Nokia 3220 [5].

Nokia 3220 Technical Specification:

Operating System	:	Nokia OS
Developer Platform	:	Series 40 Developer Platform 2.0
Java Technology	:	CLDC 1.1
		Wireless Messaging API (JSR-120)
		Mobile Media API (JSR-135)
		MIDP 2.0
		Nokia UI API
Browser	:	WAP 2.0
		XHTML over TCP/IP
Messaging	:	MMS+SMIL
		SMS
UAProfile	:	Profile 1
DRM	:	OMA DRM v1.0
Delivery Method	:	MMS
		WAP Download
Sound Formats	:	MIDI Tones (poly 16)
Functionality	:	GSM 1800
		GSM 1900
		GSM 850/900
Regional Availability	:	Africa
		Americas
		Asia-Pacific
		China
		Europe
Screen Display	:	Color Depth : 16 bit
		Resolution : 128 x 128
Physical Descriptions	:	Dimensions : 104.5 x 44 x 19 mm
		Weight : 86 g
Memory	:	Heap size : 500 KB
Shared Memory for Storage	:	4 MB
Max JAR Size	:	125 KB
Keypad Descriptions	:	3 Labeled Soft Keys
		5-way Scrolling
		Grid Key Mat
Video Support	:	3GPP formats (H.263)
Network Data Support	:	CSD
		EGPRS
		GPRS
		HSCSD
PC Connectivity	:	USB
Extra Features	:	Handsfree Speaker
		Instant Messaging
		Themes
		VGA Camcorder
		VGA Camera
		Wave Banner

[5] Device details Nokia 3220, http://www.forum.nokia.com/main/0,,018-2045,00.html?model=3220, 31-May-04

Annex G: Ticket from the Norwegian Public Roads Administration[6].

Elements:

Data Element	Description
Product Owner	The actor which owns a product and provides the customer with rights according to the product.
Product Generic Type	
Product Sub Type	
Product Retailer	The actor that sells s product to a customer on behalf of a product owner.
Component Owner	
Component Generic Type	
Component Serial Number	
Product Sequence Number	

Figure 1: Components labeling the ticket.

Unique identification of a product template instance:

Octet #	# Bits	Data element		Range
1	8	Product Owner		0 – 1 048 576
2	8			
3	4			
	4	Product Generic Type		0 - 127
4	3			
	5	Product Sub Type		0 - 255
5	3			
	5	Product Retailer		0 – 1 048 576
6	8			
7	7			
	1	Component Owner		0 – 1 048 576
8	8			
9	8			
10	3			
	5	Component Generic Type		0 - 127
11	2			
	6	Component Serial Number		0 - 4095
12	6			
	2	Date	Product Sequence Number	See Figure 3.
13	8			
14	6			
	2	Time		
15	8			
16	6			

Figure 2: Ticket outline.

[6] Norwegian Public Roads Administration: Handbook 206-3, Specification for Interoperable Electronic Ticketing Systems, June 10, 2005.

Byte #	Parameter	# bits	Range	Example value	Example bit representation							
12											0	0
13	Year	5	0 - 31	2004	0	1	1	1	0			
	Month	4	0 - 15	12						1	1	0
14					0							
	Date	5	0 - 31	24		1	1	0	0	0		
	Hr.	5	0 - 31	09							0	1
15					0	0	1					
	Min.	6	0 - 63	45				1	0	1	1	0
16					1							
	Sec.	6	0 - 63	34		1	0	0	0	1	0	

Figure 3: Date and time representation.

Annex H: UML diagrams

Figure 1 shows an external reader performing a read operation on the ticket.

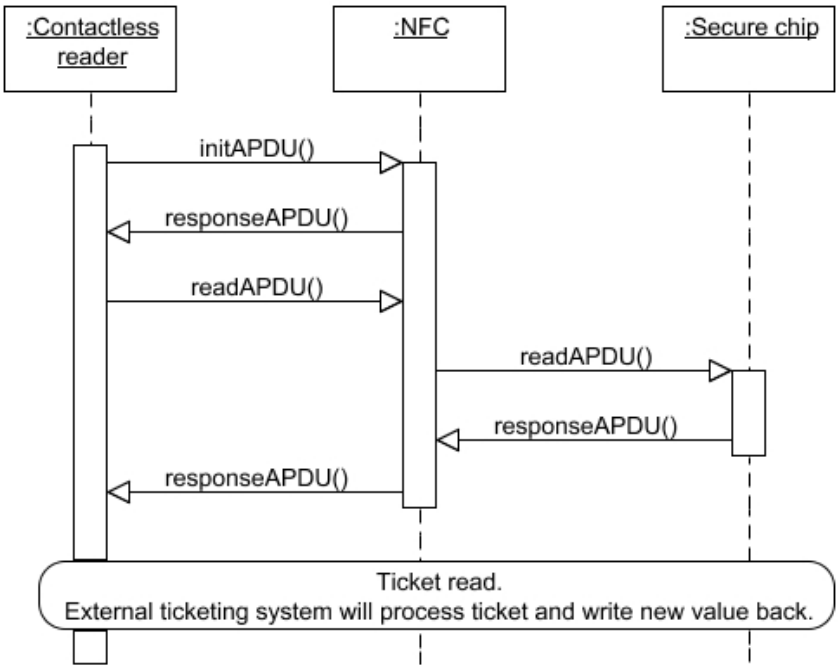

Figure 1: Message sequence diagram, external read.

Figure 2 shows an external reader writing to the secure chip.

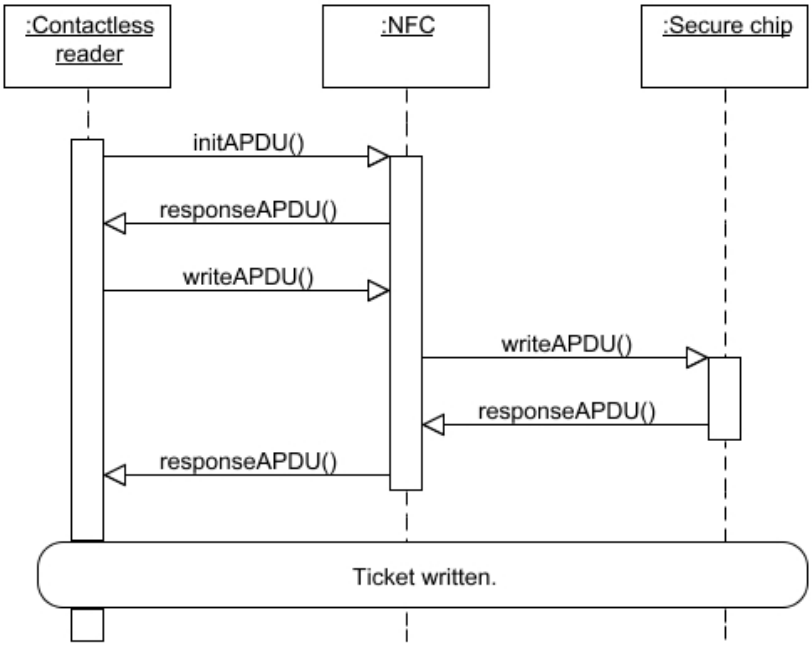

Figure 2: Message sequence diagram, external write ticket.

Figure 3 shows the MIDlet receiving a SMS that is not a valid ticket.

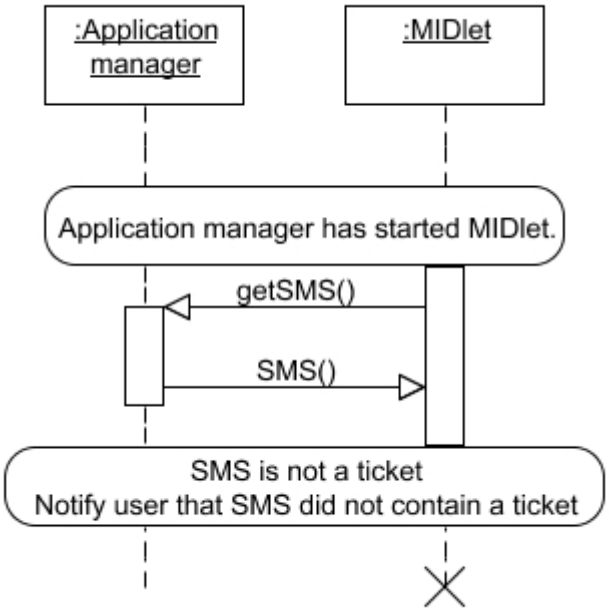

Figure 3: The incoming SMS does not contain a valid ticket.

Figure 4 shows the internal read generating an error, the user should be notified.

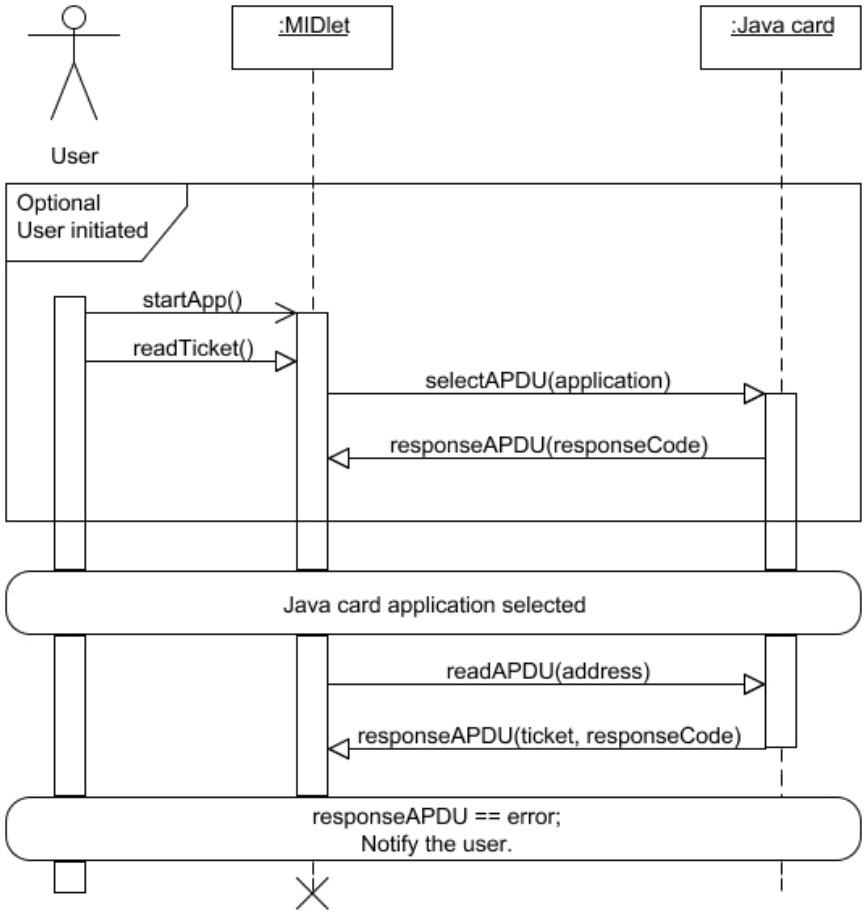

Figure 4: The internal read fails, notify the user.

Figure 5 shows the internal save failing, the user should be notified.

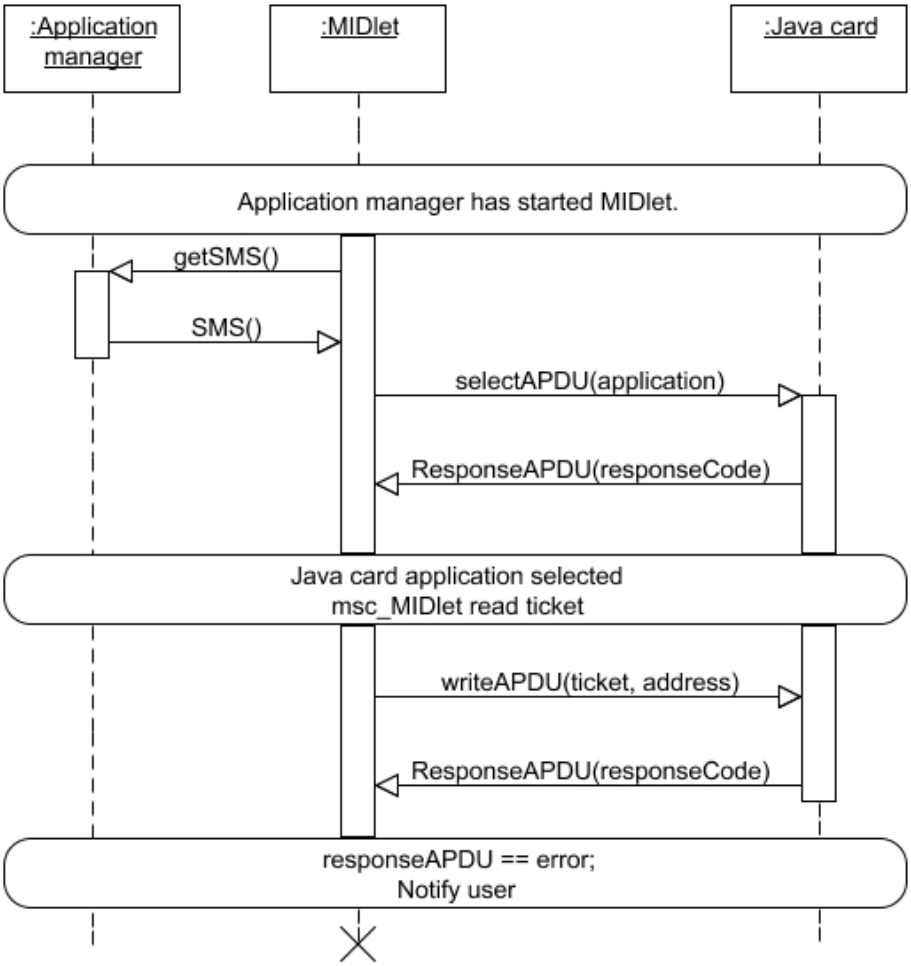

Figure 5: The internal save fails, notify the user.